

5.1 信号処理LSIにおける並列処理技術の発展

信号処理LSIは、主にベクトル演算を対象としているため、ワード処理のパイプライン演算が基本になっている。この点では、スーパーコンピュータやベクトル計算機のアーキテクチャからの発展といえる。これらのアーキテクチャの原形は、VLW (Very Long Instruction Word) 型の計算機である。これは、パイプライン処理を行うのに必要なあらゆる制御(ループ制御、メモリのアドレス制御、データ転送、演算等)を、完全水平型のマイクロ命令を使って、同時に行うものである。

このVLW型計算機を原形とした信号処理LSIの発展を図5.1.1に示す。VLW型アーキテクチャは、高性能な反面、非常に長いマイクロ命令を使用するため、制御機構が複雑になり、規模が大きくなる欠点を有している。一方信号処理LSIは、信号処理システムを小型にするためのキーデバイスであるから、VLW型アーキテクチャの利点を活かして高速化すると同時に、ハードウェア規模を削減する工夫がされた。そこで提案されたのが、マイクロプログラム制御で固定パイプラインの、非常にシンプルな並列処理構造を持った信号処理LSIである。

そしてこの構造は、その後いろんな方向へ発展していった。大きく分けると、性能追及の方向と、構造を柔軟にして信号処理LSIの応用をより広げる方向とである。前者では、パイプラインサイクルを細かくして時間方向への並列度を上げたスーパーパイプラインと、演算器の粒度を細かくして数を増やし、それをアレイ型に配置して空間方向への並列度を向上させたシストリックアレイとが代表的である。

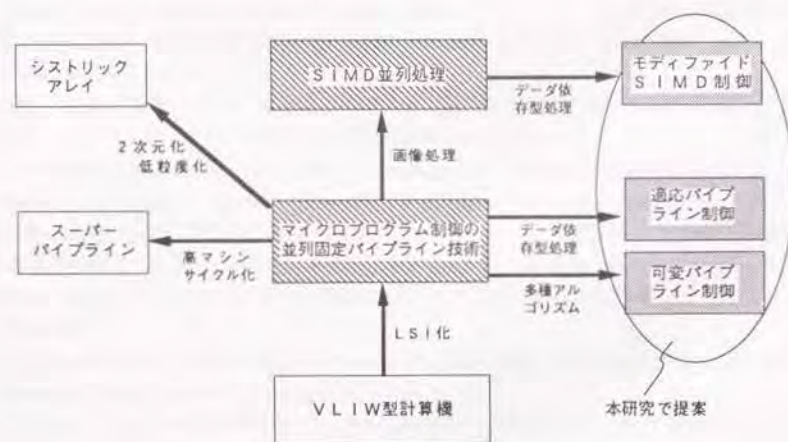


図5.1.1 信号処理LSIにおける並列処理技術の発展

後者の空間方向への並列度向上が本章で取り扱う技術であり、3種類に分けることができる。第1は、可変パイプライン技術である。これは、アルゴリズムに対応して処理のパイプラインを切り替えるものであり、汎用DSPのように、種々のアルゴリズムに対応しなければならないDSPでは必須の技術である。これについては、5.2.2節で述べる。第2は、適応パイプライン制御技術である。この技術は、特にリアルタイム制御の分野で重要である。パイプラインのサイクル時間を、外部から入力されるデータを見て、適切に変化させるものであり、単純な固定パイプライン制御では実現できない性能を実現させる技術である。これについては、5.3節で述べる。最後は、パイプラインを並列に設けて空間的にも並列度を上げた、SIMD型の並列パイプライン制御に関するものである。これは、同一の処理を大量に行うために開発された画像処理LSIから発展したアーキテクチャである。ただしこのアーキテクチャをビデオ符号化に応用するには、データによって処理を適応的に変える機構が必要である。動きの激しい画素ブロックと静止した画素ブロックとでは、符号化のアルゴリズムが異なるからである。この処理の異なる複数の画素ブロックを、並列処理する技術については、5.4節で述べる。

5.2 パイプライン制御技術[1]

前節で述べたように、DSPでは、パイプライン制御は最も基本的な技術である。今日すべてのDSPに採用されており、基本技術は確立されているといって良い。ただし、よりいっそうの高性能化を目指す場合、いくつかの改良が必要である。以下本研究で行った3点の改良について述べる。

(A) 分岐処理によるパイプラインの乱れへの対処法

(A-1) 無条件分岐処理

この場合には、図5.2.1に示すディレイドジャンプの手法が良く用いられてきた。ブランチに伴う1マシンサイクルのオーバーヘッドを無駄にしないために、分岐命令(JUMP)を実行命令(EXEC3)の手前に配置する手法である。しかしこの方法は、高度のプログラミングテクニックを要するので、極く限られたプログラマにしか使いこなせないのが難点といえる。このため、音声DSPやビデオDSPのように、ある程度汎用性を持たせ、一般ユーザもプログラミングさせるものには不向きである。

そこで、プログラマの使い勝手を重視した方法として、インストラクションフェッチの前のサイクルで、プリデコーダによりブランチを検出し、ハードウェアで強制的にノーオペレーション命令を挿入する方法を検討した。この方法は、プログラマが意識しなくても、自動的にタイミングを合わせられる方法である。詳細は、パイプラインシーケンサ構成技術として、5.2.1節で述べる。

(A-2) 条件付き分岐処理

ハードウェア設計の立場からは、条件付き分岐処理は、2つの型に分けて考えると理解しやすい。ひとつは、ループリターンやサブルーチンリターンのように、分岐条件が比較的速く検出できるものである。もうひとつは、条件ジャンプや条件付きサブルーチンコールのように、主として演算結果に基づいて分岐するので、分岐条件の検出に時間がかかり、そのため最長遅延パスとなって、演算サイクル向上のネックとなるものである。この最長遅延パスの問題は、浮動小数点ALUを搭載している浮動小数点DSPの場合に顕著である。これには、多相クロックを用いて、制御側と演算側とで異なった処理時間を配分する方法を検討した。例えば4相クロックを用いて、制御側に3/4マシンサイクル、演算側に5/4マシンサイクルを割り当てるのである。この検討結果についても5.2.1節で述べる。

(B) 種々のパイプライン演算への対処法

信号処理の基本演算は、フィルタの計算に使われる2段パイプラインの積和演算であるが、当然これだけではなく、差分と乗算と累算の3段パイプラインを要する2乗距離計算のもの、累算の様に1段パイプラインのもの、さらには単なるスカラー演算等が混在する。信号処理LSIでは、これらすべてを効率良く処理することが望まれる。

そこで、ビデオ符号化処理の演算タイプを検討し、種々のパイプライン演算に対処できる方法を検討した。これは、種々の処理に対処できる最大公約数のハードウェアを備えさせ、処理するベクタ演算に合わせて、パイプラインデータバスを選択（セットアップ命令）して構築する手法である。この手法を可変パイプライン制御技術と呼び、5.2.2節で述べる。

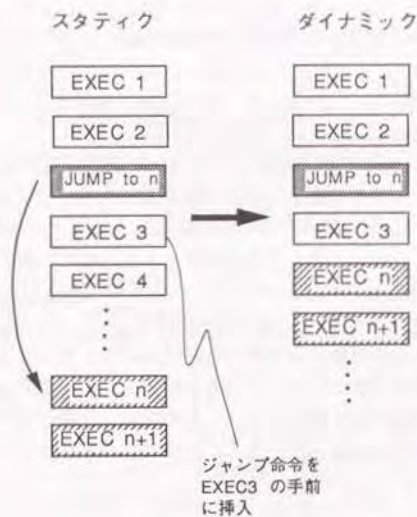


図5.2.1 ディレイジャンプの説明

5.2.1 パイプラインシーケンサ構成技術[2][3]

(1) シーケンサ構成

プログラムとデータのバスを分離したハーバードアーキテクチャは、メモリバンド幅が広く、パイプライン制御に適した構造である。本シーケンサでも、このハーバードアーキテクチャを基本としている。また分岐時に、ノーオペレーション命令（NOP）をハードウェアで自動的に挿入する機構を持たせることで、プログラミングの容易性と、分岐のオーバーヘッド削減を図っている。このプログラムシーケンサの構成を図5.2.2に、また分岐時のタイミングを図5.2.3に示す。

インストラクションレジスタ（IR）の前にプリデコード（PDEC）を置き、インストラクションフェッチをする前のサイクルで分岐命令を検出する構成である。このプリデ

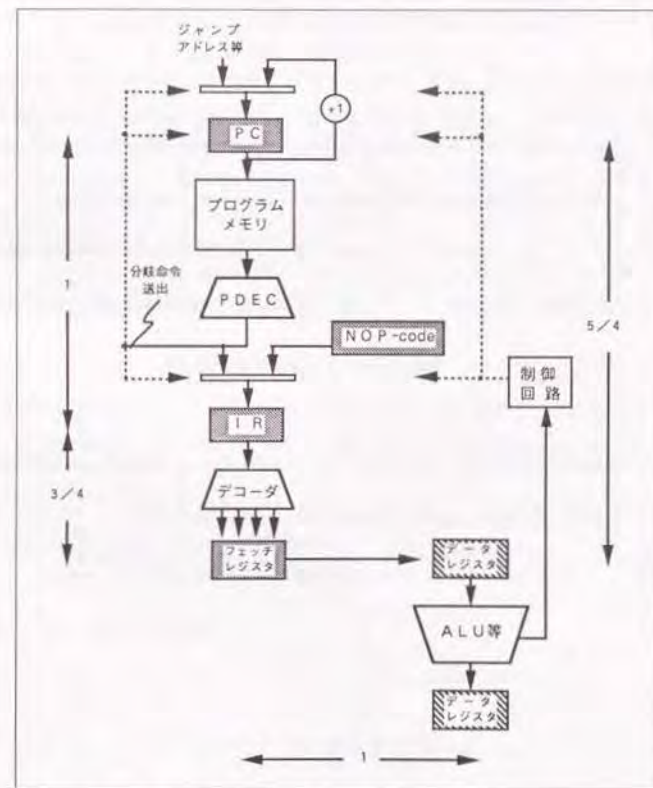
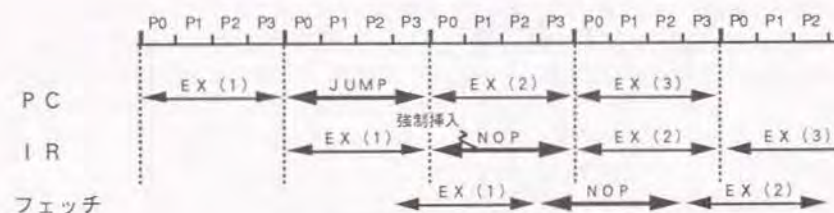


図5.2.2 改良したプログラムシーケンサの構成

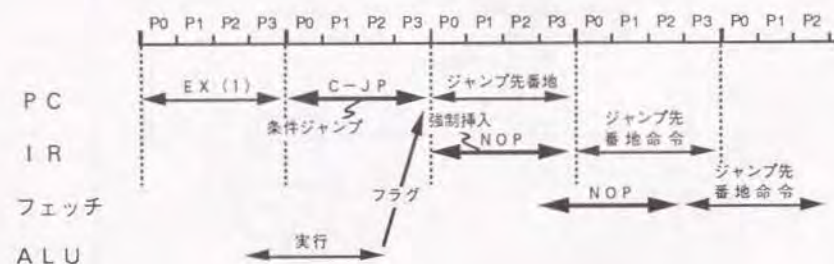
コードによる先行検出を使って、次のサイクルのプログラムカウンタ値 (PC) とインストラクションレジスタへの命令を制御することで、分岐命令でのノーオペレーション命令 (NOP) の自動挿入を実現している。

特に無条件分岐のときには、命令を検出したとき、インストラクションレジスタにノーオペレーション命令 (NOP) を強制挿入し、同時にプログラムカウンタを1サイクル止める処理を行う。

また条件分岐命令のときには、演算に1マシンサイクルを割り当ててクロックを高速化しているため、演算系からの分岐フラグの出力に1サイクルを有することとなり、このままでは、条件分岐に常に2マシンサイクルを費やすことになる。そこで、プログラムカウンタとインストラクションレジスタの位相を1/4マシンサイクル遅らせ、インストラクションレジスタとフェッチレジスタ間を3/4マシンサイクルで処理できるようにした。そして、データレジスタと、プログラムカウンタおよびインストラクションレジスタ間に5/4マシンサイクルを割り当てることとし、フラグの条件判断処理に1/4マシンサイクルを割り当てることができるようにした。これにより、条件付き分岐処理 (特に条件ジャンプやサブルーチンコール) で、余分なNOP命令を1命令省くことができた。



(A) 無条件分岐 (NOPの強制挿入)



(B) 条件付き分岐 (オーバーヘッド削減)

図5.2.3 改良した分岐命令のタイミング

(2) 多相による基本タイミング

前節で述べた条件分岐命令のオーバーヘッドを改善するためと、パイプライン制御を行うときの、データバス系と制御系とのタイミングマージンを、十分とれるようにするために、多相クロック制御を検討した。以下4.3節での音声信号処理DSPを取り上げ、そのタイミング制御について述べる。

(2-1) クロックドライバ系の全体構成

音声処理DSPでは、クロックの相数を4相と定めた。多相クロックの利点は、クリティカルパスの最適配分、制御とデータのマージン確保、内蔵メモリへのタイミング供給等を、遅延回路を使わずに構成できるため、安定な同期設計ができることである。逆にクロックの相数を増やし過ぎると、クロックを高速化したときに、各相間のデューティを正しく取るのが困難となり設計マージンが減少するため、高速化できないことになる。また本来、DSPのようなRISCタイプのプロセッサでは、すべての命令の実行時間を1マシンサイクルに規格化しているため、4相以上の相数を必要としない。

この場合のクロックドライバ系の構成を図5.2.4に示す。40MHzの外部クロックをクロック分周器で分周して、20MHzの内部クロック (CK) と、CKを1/4相遅らせたクロック (CKD) を作り、これを対にして各回路ブロックに供給する。分周器から最終段までのドライバ段数をすべて3段に統一し、かつ、負荷の軽い所にはダミー負荷を設けて負荷を同一としバランスさせた。これにより、クロックスキューを1.5ns以下とした。

各回路ブロックではCKとCKDクロックを受け、ブロック内のNOR回路とNAND回路とにより、レジスタやカウンタのイネーブルを制御するRECK信号と、データバスの3値ドライバを制御するBEN信号を生成する。各ブロックの入り口で制御信号を生成させ

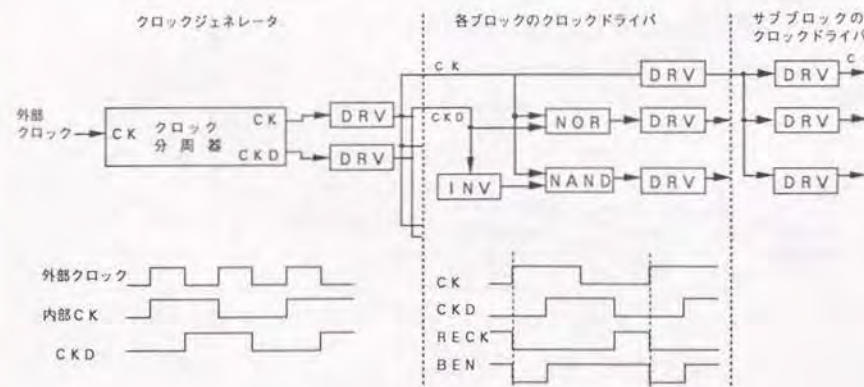


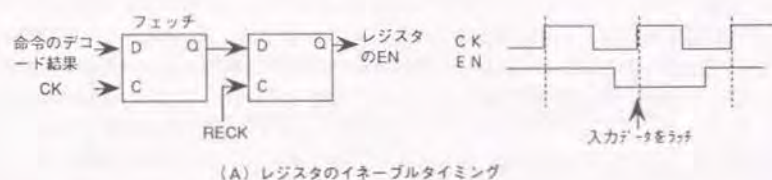
図5.2.4 クロックドライバ系の構成

ることにより、負荷バランスを取りやすくし、制御信号間および制御信号とクロック間のスキューを小さくしている。またブロック内では、同一のCKドライバとRECKドライバでドライブするように統一し、ブロック内のレジスタ間スキューをさらに小さくしている。

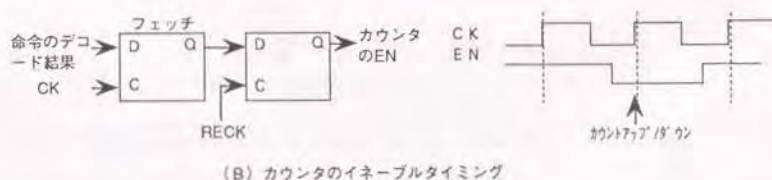
(2-2) 各種タイミング

4相クロックを用いた代表的なタイミングを図5.2.5に示す。レジスタとカウンタのイネーブル信号は、CKとCKDのNORを取って生成したRECK信号を使って、フェッチ出力を3/4マシンサイクル遅らせて出力させる。これにより、クロック(CK)とイネーブル(EN)間に、前後1/4マシンサイクルのマージンを持たせることができる。

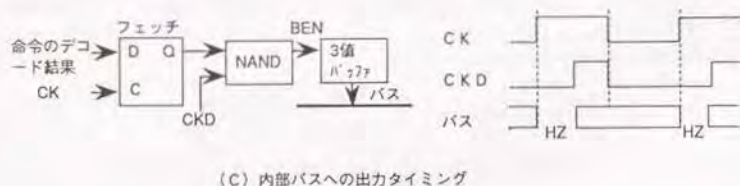
内部バスでは、データを3/4マシンサイクルだけ出力し、1/4マシンサイクルのHZ(ハイインピーダンス)区間を設けることにより、過渡的なバスクラッシュを防止し、LSIの信頼性を高めることとした。そのため、CKとCKDのNANDを取って、後方3/4マシンサイクルがイネーブルとなる信号を生成し、3値バスドライバをドライブする構成とした。



(A) レジスタのイネーブルタイミング



(B) カウンタのイネーブルタイミング



(C) 内部バスへの出力タイミング

図5.2.5 4相による各種タイミング

5.2.2 可変パイプライン制御技術[4]

ビデオ符号化処理で現われる種々の演算を、パイプライン段数に着目して分類すると、表5.2.1のようになる。この表から、演算は、3段パイプラインのベクトル演算から、スカラ演算まで分布していることがわかる。3段パイプラインのベクトル演算の例として2乗距離計算がある。これは、まず、データをデータメモリから読みだし、「ALU+乗算器+加減算器」構成の3段のパイプライン演算器に投入し、ベクトルの2乗距離計算を行って、その結果をデータメモリに格納する演算である。またスカラ演算には、ワード単位でビット処理を行うものから算術演算まで、種々のものが存在する。

上記の種々のパイプライン段数の処理を、すべて行える演算器の例を図5.2.6に示す。データメモリと3個の演算ユニットをパイプラインレジスタで挟み、適宜マルチプレクサを配置して迂回路を設けることにより、種々の段数のパイプラインバスを構成できる構造である。またDSPでは、データメモリのサイクルタイムが最長遅延となることが多いので、このサイクルタイムいっぱいまでマシンサイクルを短くする設計が行われる。このため、データメモリ間でスカラ演算を行うと、メモリのリードサイクルとライトサイクルが加わるため、1ワードの処理に3サイクルを要する。このオーバーヘッドを解決するため、アキュムレータとは別に、スカラ演算用のレジスタファイルを設けてALUまたは乗算器と直結させ、1サイクルで処理可能とした。

ただしこの構造の欠点は、制御要素が多くなるため、マイクロ命令のフィールド長が非常に長くなり、従ってマイクロ制御メモリが大容量になってオンチップ化出来なくなることである。これを解決するため、セットアップ命令を使った制御法を検討した。これは、ベクトル演算を実行する前に、その処理に合わせて、予めデータバスを設定する手法である。一つの長いマイクロ命令を、セットアップ命令と実行命令とに分割する制御法である。この命令分割により、明らかに処理時間のペナルティが生じるが、例えばビデオ処理では、64〜

表5.2.1 各種ビデオ処理演算でのパイプライン段数

演算種別	数式	応用	パイプライン段数
加算、乗算、比較	$X+Y, X \cdot Y, \text{COMP}(X, Y)$	各種	1
ワードの一致検出	$\Sigma \text{EXOR}(x_i, y_i)$ ← 不一致ビットの積算	特定コード検出	2
乗加算	$\Sigma (A_k \cdot Y_k)$	DCT計算 各種フィルタ	2
距離計算、 コードブック検索	$\Sigma (X_k - Y_k)^2, \Sigma X_k - Y_k $	動きベクトル検出 ベクトル量子化	3 (2) (*)

(*) 絶対値累算は、(差分+絶対値)演算をALUで一挙に行えるので、段数は2段。

-122-

5.3.1 開発LSIの位置づけ

装置化にあたっては、特に2つの点を考慮した。第1点は、連続移動描画で各ショット（描画する1つの矩形）をパイプライン描画するときの高速化である。この連続移動描画では、現在のステージ位置を高精度に検出し、それに基づいて、あるショットを実際に描画する未来の時点での描画の位置を予測し、そしてその予測位置でのレンズ歪みとウエハ歪みの補正をリアルタイムで処理しながら描画するのであるが、このショットを描画する時間（ショット時間）や、座標値等を駆動回路に伝えるときのD/Aコンバータのセットリング時間にはデータ依存性があるため、単純なパイプライン制御では処理オーバーヘッドが大きくなってしまふ。そこでこの部分に、適応パイプライン制御の技術を導入した。

以下本節では、上記2つの高速化技術を中心に述べる。そしてこの技術を実体化するために開発した3種のLSIについて述べ、リアルタイム制御用の専用LSI設計法を示す。

(1) 全体の概要

描画チップおよびレンズ等の歪みは、描画に先立って予めマーク検出処理部に求める。ステージに書き込まれた検出マークにビームを当て、その反射信号を解析して補正テーブルを生成し、結果を偏向補正部に送る。偏向補正部は、この補正テーブルを参照して、上記リアルタイム補正を実行する。

本描画装置では、主偏向と副偏向の2階層描画方式を用いている。この制御は、偏向補正部が担当している。先ずホスト計算機が指示するユニットベルト単位の起動命令を、

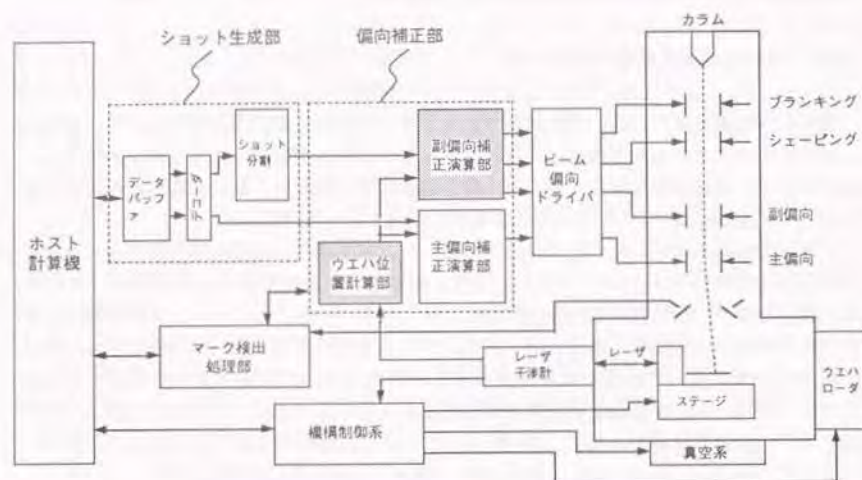


図 5.3.1 電子ビーム直描装置の構成

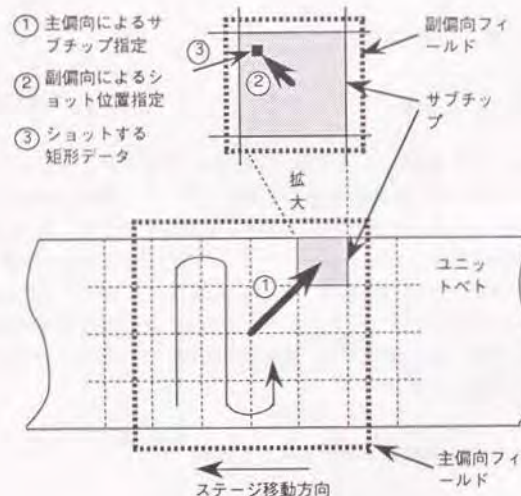


図 5.3.2 連続移動描画の概念図

ショット生成部が受ける。するとショット生成部は、この命令をユニットベルト内の1つのチップ単位毎の起動命令に展開し、チップ単位毎に主偏向補正演算部を起動する。

次に主偏向補正演算部は、ショット生成部より主偏向データを受取り、 $80\mu\text{m} \times 80\mu\text{m}$ サイズのサブチップ単位で制御を行う。具体的には、サブチップ（副偏向描画領域）の中心点を指定し、そのデータをビーム偏向ドライバの主偏向DAC（DAコンバータ）へ送出する。即ち主偏向によって、ビームは描画するサブチップの中心位置を指すことになる。

次に副偏向補正演算部は、ショット単位での制御を行う。即ち、ショット生成部からサブチップの中心点からの相対位置を指定する副偏向データを受取り、またウエハ位置計算部からステージ位置の正確な情報を受取って、リアルタイムでレンズ歪みおよびウエハ歪みの補正演算を行い、描画する矩形データの形状と描画位置情報とをビーム偏向ドライバの副偏向DACへ送出する。この副偏向補正演算部を（A）パイボラのショットタイミング生成LSIと、（B）BiCMOSによる線形行列演算LSIとで構成した。

さらにウエハ位置検出部では、レーザー干渉計から生成される分解能 $0.01\mu\text{m}$ のレーザーパルスを受取り、リアルタイムでカウントして積算し、ステージ位置を算出する。このブロックの主要部は、BiCMOSによるレーザーカウンタLSIで構成した。

5.3.3 リアルタイム制御の概要

（1）主副2階層描画方式

試料を連続移動させながら主副2階層描画を行なう方式の概要を図5.3.2に示す。この方式は、描画速度の向上に効果的である。まず主偏向補正演算部が、（1）描画するサブチップを指定する。次いで副偏向補正演算部が、（2）ショット位置を指定する。位置が決まったところでブランキングが解除され、（3）ショットとなる。（2）及び（3）は、（1）で指定したサブチップ内の全ショットに対して行われ、終了後、次のサブチップが指定され、上記動作の繰り返しとなる。

実現方法としては、主偏向補正演算部と副偏向補正演算部とで、その特長に合わせて異なったアプローチを取った。主偏向補正演算部で行なうサブチップ指定は、制御が複雑でかつ処理量が多いことから、汎用DSPをマルチで動作させ、プログラム制御で行うこととし

表 5.3.1 副偏向描画の補正項目の分類

補正種別	補正内容
位置補正	ウエハ歪み補正 (#1)
	ウエハ位置補正 (#2)
	レンズ歪み補正 (#3)
形状 (矩形) 補正	レンズ歪み補正 (#4)
描画時間補正	隣接効果補正 (#5)

た。また副偏向補正演算部では、サブチップ当り、平均5000程度のショットデータの補正演算を行なうため、非常に高速処理が要求される。そこで、前節で述べた専用のLSIで処理することとした。

(2) 試料連続移動合わせ描画方式

試料連続移動合わせ描画方式は、試料を載せているステージを等速度で移動させながら描画していく方法であり、ステージの静止・移動の加速度動作を必要とするステップ&リビート方式に比べ、格段にスループットの高い描画方式である。ここでは、この試料連続移動合わせ描画方式で副偏向描画を行なうときに必要な、5種の補正処理(専用LSIで必要とする処理)について述べる。表5.3.1に必要な補正処理の項目を示す。

まずウエハ歪み補正(#1)は、種々のプロセスを経ることに伴うウエハの変形を補正するものである。これには、予めブロック単位にウエハ上に設けた補正マークを検出し、一次近似で補正係数表を作成しておき、描画時に、このテーブルをショットデータ毎にリアルタイムで呼び出して、補正演算を行なうことで補正する。

次にステージ移動に対する、ウエハ位置補正(#2)は、試料連続移動描画方式に伴う補正であり、ここでは、主偏向で定めた位置を原点とし、そこからの移動量を逐次測定してその差分を副偏向の座標に変換して描画する。当然のことながら、x方向とy方向の両方について補正演算を行なう。また、ショットデータ毎に行う。

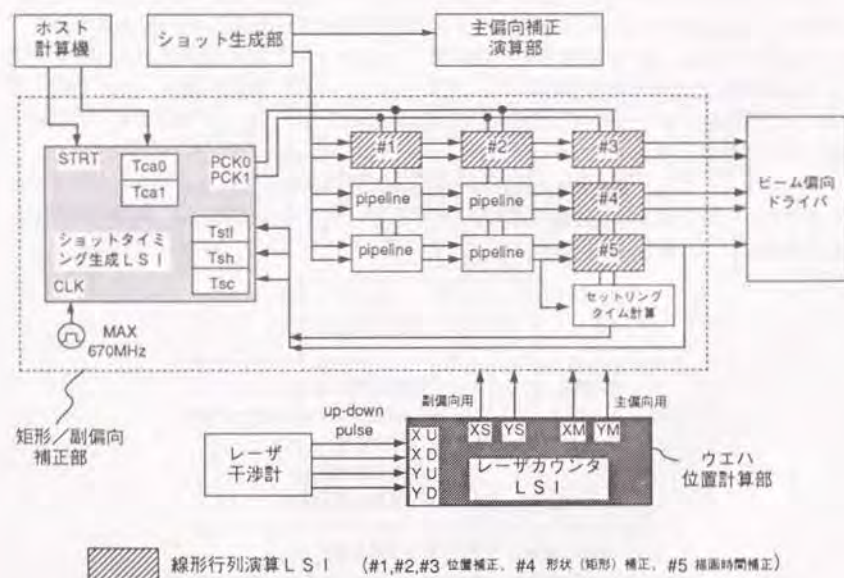


図5.3.3 リアルタイム制御系のブロック図

レンズ歪み補正(#3, #4)では、ウエハ歪みと同様、ステージ上の補正マークを使ってレンズ歪みの補正テーブルを作成しておく。そして、このテーブルを呼び出しながらリアルタイムで補正演算を実行する。補正は、ショットデータの位置と形状の両方について行う。

最後に近接効果補正(#5)は、近接したパターン間の干渉を補償するものである。この補正係数は、個々のショットデータそれぞれが持っているもので、データの座標および形状にそれぞれの補正係数を掛けて正確な値を計算しながらショットする。

(3) リアルタイム制御系の構成[9]

ここでは、制御系の具体的な構成について述べる。図5.3.3に構成図を示す。制御系は、大きく分けて、矩形/副偏向補正部とウエハ位置計算部とから構成されている。矩形/副偏向補正部は、ショットタイミング生成LSIと5個の線形行列演算LSIとからなっている。この内ショットタイミング生成LSIは、2相のパイプラインクロックPCK0、PCK1を生成して、補正とショットからなる基本のショットサイクルを制御する。

5個の線形行列演算LSIは、前節で述べた副偏向描画の各種補正演算を行う。図中の線形演算LSIにつけた番号(#1~#5)は、表5.3.1で示した5つの補正項目に対応している。補正演算で必要な正確な位置情報は、ウエハ位置計算部のウエハのレーザカウンタLSIから受け取る。

5.3.4 高速化技術

前節で述べたリアルタイム制御系で、リアルタイム性と高精度を実現するために、検討し導入した5個の技術を、以下に示す。

(A) 最大時間分解能1.5 ns

照射時間およびDACセットリング時間を1.5 ns単位(タイミング精度±0.3 ns以下)で細かく指定できるようにした。この高速性を実現するため、タイミング生成部をすべて、ショットタイミング生成LSIに集積した。

(B) 2相パイプライン演算

補正演算では、乗算と加減算をパイプライン処理するが、このとき、加減算では、語長の長い加減算とシフトおよび丸め処理を、1サイクルの間に行なうため、乗算に比して演算時間が長くなり、パイプラインサイクルの高速性が制限される。これを解決するため、乗算サイクルと加減算サイクルそれぞれに最適な演算サイクルを割り当てることができる2相パイプライン演算方式とした。具体的には、タイミング生成LSIから、PCK0とPCK1の2相クロックを供給して、乗算と累算の時間を最適に配分した。時間配分例を下に示す。

乗算サイクル: $T_{ca0} = 25 \text{ ns}$

累算サイクル: $T_{ca1} = 40 \text{ ns}$

(C) 補正演算の適応パイプライン制御

描画時には、補正演算とビーム照射（ショット）が繰り返し実行される。この場合、演算時間の方は、LSIの性能で決まるので、常に同じサイクル時間 $T_{cyc0} = T_{ca0} + T_{cal}$ となる。一方ビーム照射の方は、DAC（DAコンバータ）のセットリング時間と照射時間の和で決まるため、照射データによって大幅に変化する。例えば、セットリング時間 T_{stl} は、照射位置が大きく変化した時と小さく移動した時とで一桁以上異なる。また照射時間は、近接効果補正により、数割程度変化する。

このため、ショットデータに対応してパイプラインサイクルを変化させる制御を考案して、処理の効率化を図った。即ち、補正演算の時間とビーム照射の時間を比較して、いずれか短い方に調整時間 T_d を加えることで、パイプラインサイクルが回転するようにした。言い換えると、ショットサイクル T_{cyc} が、演算サイクル（ $T_{cyc0} = T_{ca0} + T_{cal}$ ）と、ビーム照射サイクル（ $T_{cyc1} = T_{stl} + T_{sh}$ ）のいずれか長い方で決定される様にした。すなわち、

- (1) $T_{cyc0} \geq T_{cyc1} : T_{cyc} = T_{cyc0}$
- (2) $T_{cyc0} < T_{cyc1} : T_{cyc} = T_{cyc1}$

さらに、 T_d を演算サイクルの前方で加算する工夫を加え、前のデータのビーム照射が長い場合でも、補正演算終了と同時に次のビーム照射を実行できるようにした。これにより、せっかくの補正演算の後にステージ移動に伴う誤差が混入するという不都合が生じない様にした。以上のタイミングを図5.3.4に示す。

(D) 位置検出の高速化

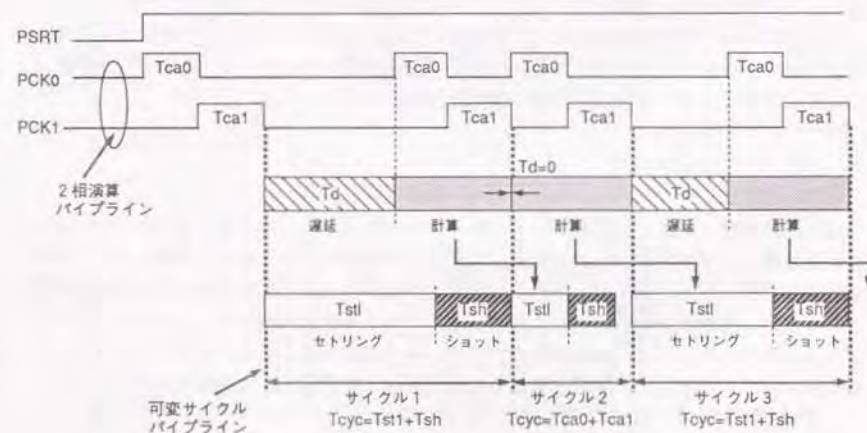
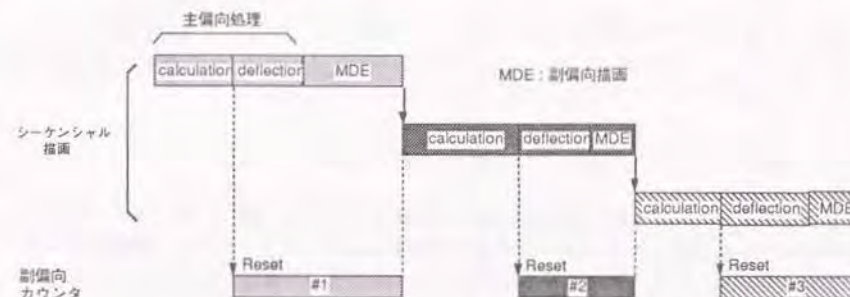
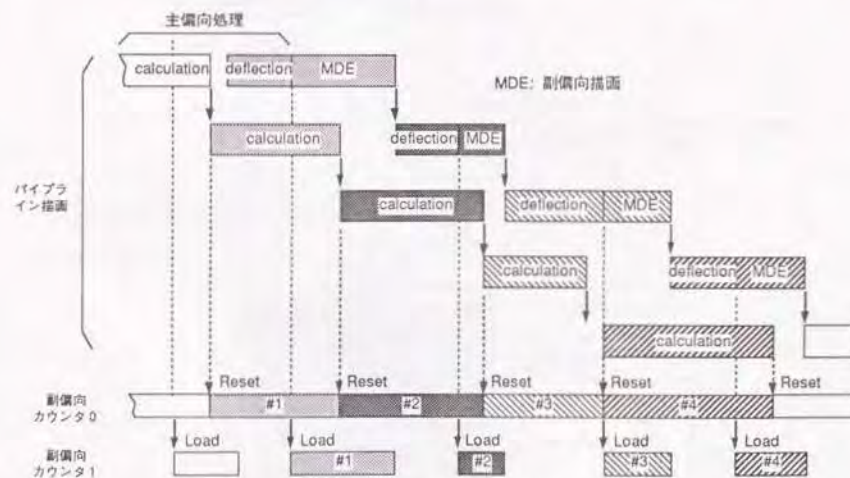


図5.3.4 可変サイクルパイプラインのタイミング

レーザ干渉計では、 $\lambda = 6329.9141 \text{ \AA}$ のHe-Neレーザを基本波長として用いているため、最小分解能は、 $\lambda / 64 = 98.904908 \text{ \AA}$ である。これを、 $0.01 \mu\text{m}$ 単位の座標に変換するためには、レーザカウンタLSIで、積算カウンタからの出力（レーザ干渉計からのパルスをカウントして $\lambda / 64$ 単位の座標値を出力）に、変換係数 “0.98904908...” を乗じる補正乗算が必要である。カウンタおよび乗算器の精度は26b（可動範囲 $\pm 320 \text{ mm}$ ）必要なので、このLSIのクリティカルパスは補正用の並列乗算器である。2次 Booth + 4b-C SA加算器構成とし、さらにクリティカルとなるYデコード系を補正係数入力とすることで高速化し、最高3.2MHzで動作するようにした。これにより、ステージの最高速度を 320 mm/s まで対応可能とし、描画のスループットを十分なものとした。



(a) 従来のタイミング



(b) パイプラインによるタイミング

図5.3.5 主偏向と副偏向の2階層描画の効率化のために開発したデュアルカウンタによるパイプライン処理

(E) 副偏向系をダブルカウンタ構成

主偏向と副偏向からなる2階層描画のバイブライン処理を効率化するため、レーザカウンタLSIの副カウンタを2階層構成(副カウンタ0、副カウンタ1)とした。これは、1つの主偏向が完了し、これに基づいた副偏向処理を行うとき、その副偏向処理の終了を待たずに次の主偏向処理を開始できるようにするためである。この時のタイミングを図5.3.5に示す。副カウンタ0は、主偏向演算開始時点を実点(0,0)とする移動量をカウントし、副偏向処理開始時にその時までの積算量を副カウンタ1へロードする。副偏向処理系は、この副カウンタ1の出力を受けて補正演算と描画を実行する。この時副カウンタ0は、空き状態となるので、このカウンタを使って次の主偏向演算をすぐに開始できる。

5.3.5 3種のLSIの概要

(1) ショットタイミング生成LSI

(1-1) 機能

ショットタイミング生成LSIのブロック構成を図5.3.6に示す。最大670MHzの外部クロックに同期して、各タイミングを生成するためのカウンタ群から構成されている。処理の開始は、コントローラからのバイブラインスタート信号(PSRT)である。

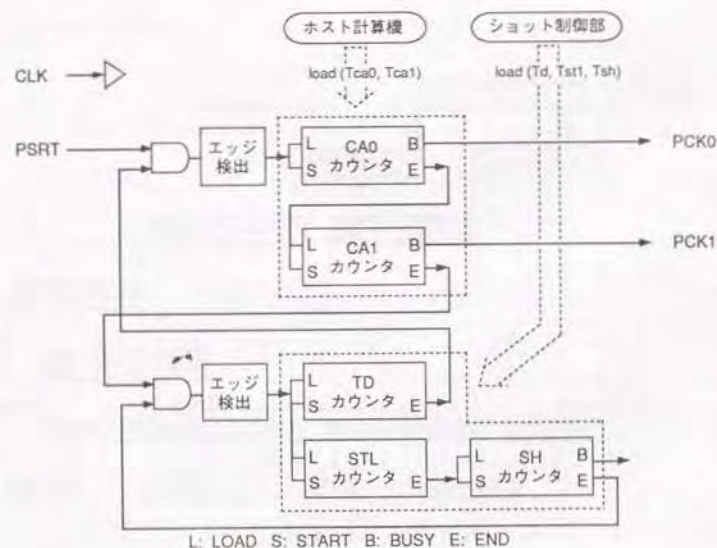


図5.3.6 ショットタイミング生成回路LSIの構成

各カウンタはエッジトリガのロードスタート(L, S)を有するダウンカウンタであり、このうち、CA0およびCA1カウンタは一定の補正演算時間Tca0、Tca1を生成する。この2相演算バイブラインにより、演算遅延を65ns(25ns+40ns)に抑えている。TD、STL及びSHカウンタは、補正演算系より、DACセットリング時間Tsd、ビーム照射時間TSH、およびバイブラインサイクル補正時間(TSC=TSTL+TSH-Tca0-Tca1)を各バイブラインサイクル毎に受取り、DACのストローブ信号や電子ビームプランカの開閉信号(図5.3.6では省略)を出力すると共に、次のバイブラインサイクルの開始エッジを生成する。

(1-2) 性能

LSIの諸元を表5.3.2に示す。基本ゲート遅延80psのSST-1A(Super-Self-aligned process Technology)を使った、高速バイポーラゲートアレイを使用した[10]。最高動

表5.3.2 ショットタイミング生成LSI諸元

機能	可変バイブライン演算の タイミング生成
カウンタ語長	15b (MAX) (0~80μs at 400MHz)
性能	
動作クロック	670MHz (MAX)
タイミング分解能	500ps
I/Oレベル	ECLコンパチ
電源電圧	-3.3V
消費電力	3.3W
パッケージ	120ピンセラミックFP
構成	バイポーラゲートアレイ
実効ゲート数	2KG (セル使用率80%)
基本ゲート遅延	80ps
チップサイズ	5.0mm×5.0mm

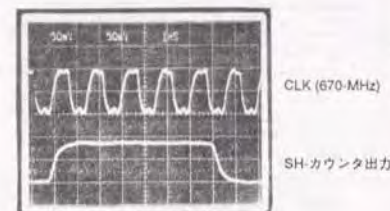


図5.3.7 SHカウンタのB信号波形
(670MHz動作)

作クロック周波数を規定する回路部分は、15bの、TD、STL、およびSCの各カウンタである。高速化のため、4b単位のCLA回路を使用して、最高670MHzの動作を確認した。この時の波形を図5.3.7に示す。これは、SHカウンタのビジー（図5.3.6のB出力）出力であり、670MHzパルス4個分の時間をイネーブルとした例である。回路規模が大きいので、レイアウトをマニュアルで行い、80%のセル使用率を実現した。レアゲート2.5KG、実行ゲート2.0KGである。3.3Wの消費電力に対応するため、チップは、キャビティダウン型のセラミックフラットパッケージに実装し、風速1.5m/sの強制空冷下で使用した。

(2) レーザカウンタLSI

(2-1) 機能

レーザカウンタLSIの構成を図5.3.8に示す。X系、Y系およびヨーイング系から構成されている。それぞれ、X方向の移動量、Y方向の移動量、そして傾きを検出する。X系は、描画開始時からの絶対座標をカウントする主偏向カウンタと、主偏向で指定したサブチップエリア（80μm×80μm）の中心点を原点とする相対座標をカウントする副偏向カウンタとから構成されている。副偏向カウンタは、5.3.4節章で述べたように、主偏向

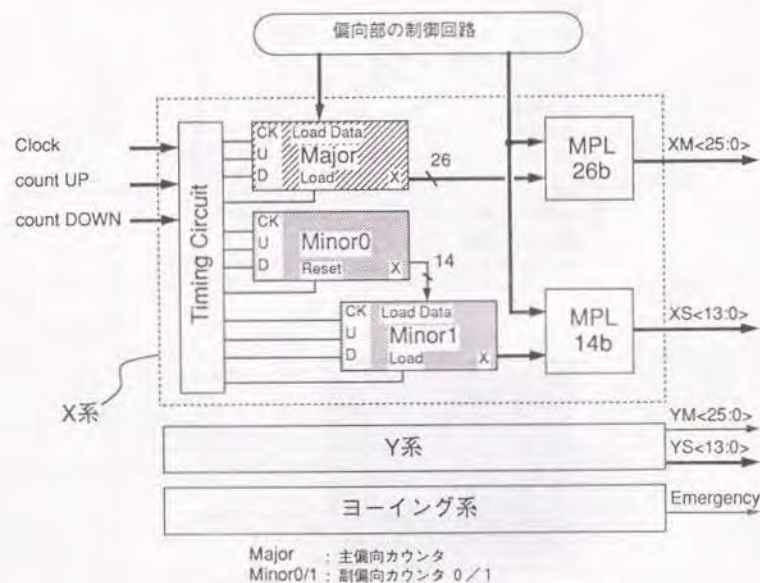


図5.3.8 レーザカウンタLSIのブロック図

と副偏向のバイプライン処理を実現するためにデュアル構成としている。Y系についても全く同様である。

(2-2) 性能

LSIの諸元を表5.3.3に示す。性能を決定する26bの並列乗算器の遅延時間は30ns弱であり、このためLSIは最高32MHzで動作する。低消費電力化を図るため、電源変換回路を内蔵した、外部5.0V、内部3.0VのBiCMOSゲートアレイを使用した。これにより、4個の乗算器がすべて動作している時でも、消費電力を2.0W以下に抑えることができた。レアゲートは62KG、セル使用率は40%である[11]。

(3) 線形行列演算LSI

(3-1) 機能

図5.3.3で述べた5種類の補正演算は、すべて双一次項までを考慮した次式により、十分な精度で補正できることがわかっている。

$$X'' = X + a_0 + a_1 * X + a_2 * Y + a_3 * X * Y \quad (5.3.1)$$

$$Y'' = Y + b_0 + b_1 * X + b_2 * Y + b_3 * X * Y$$

この双一次項までを含めた補正演算が、ショットサイクルの演算時間を決定する最長遅延時

表5.3.3 レーザカウンタLSI諸元

機能	アップ&ダウンパルスを、主偏向と副偏向のx、y座標に変換して出力。
語長（分解能）	26b
主偏向用	(±320mm/0.01μm)
副偏向用	14b
	(±80μm/0.01μm)
動作クロック	32MHz (MAX)
最大遅延	3クロック
I/Oレベル	TTLコンパチ
電源電圧	5.0V
消費電力	2.0W
パッケージ	208ピンセラミックPGA
構成	0.8μm BiCMOS
	ゲートアレイ
実効ゲート数	25KG
チップサイズ	12.2mm×8.4mm

間になっているので、この変換機能すべてを集積化することが望ましい。しかし、(5.3.1)式の機能を集積化すると、50Kゲート相当の回路規模を要するので、コストおよび消費電力の面で望ましくない。そこで計算機解析を行ない、最後の双一次項を除いた近似式の精度を計算した。その結果、レンズ歪み補正以外は精度上無視して差しつかえないことが確認できたので、(5.3.2)式で示す1次の行列演算機能のみを集積することとした。

$$\begin{aligned} X'' &= X + a_0 + a_1 * X + a_2 * Y \\ Y'' &= Y + b_0 + b_1 * X + b_2 * Y \end{aligned} \quad (5.3.2)$$

双一次項まで必要なレンズ歪み補正には、次のように2個のLSIをカスケードに接続することで対処した。

$$\begin{aligned} \text{[LSI 1]} \\ X'' &= X + a_0 + a_2 * Y + f_1 * X \\ Y'' &= Y + b_0 + b_1 * X + f_2 * Y \end{aligned} \quad (5.3.3-A)$$

$$\begin{aligned} \text{[LSI 2]} \\ f_1 &= a_1 + a_3 * Y \\ f_2 &= b_2 + b_3 * X \end{aligned} \quad (5.3.3-B)$$

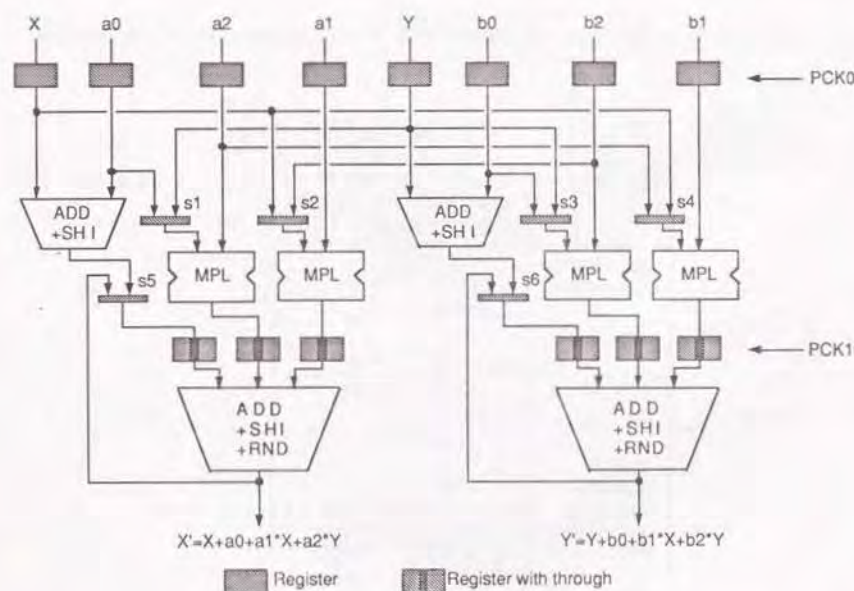


図5.3.9 線形行列演算LSIの構成

並列乗算器4個、2入力加減算器2個、および3入力加減算器2個からなるLSIの構成を図5.3.9に示す。本電子ビーム露光装置で用いる2変数の1次変換以外にも、デジタル信号処理に広く適用できるようにするため、以下の工夫を行った。

- (A) 3入力加減算器に累算機能を付与し、距離計算や相関演算を可能とした。
- (B) 入出力16b、演算40bとし、8入力2出力の多ピン化に対応すると共に、累算および一次変換演算での精度を確保した。
- (C) 第二パイプラインレジスタにスルーモードを付与し、乗累算を25MHz 2段パイプラインで動作させる高速モードと、17MHz 1段パイプラインで動作させる低遅延モードの両方利用可能とした。

(3-2) 性能

LSIの諸元を表5.3.4に示す。レーザカウンタLSIと同一のゲートアレイで試作し、パイプラインモードで、200MOPS (25MHz)、低遅延モードで134MOPS (16.7MHz) を実現した。

表5.3.4 線形行列演算LSI諸元

機能	2個の変数と6個の係数を入力し、2組みの1次変換を行い、結果の2データを出力
パイプラインサイクル	25MHz
第一パイプライン遅延	25ns
第二パイプライン遅延	40ns
スルーモード	17MHz (60ns)
演算語長	16b
入出力	40b
乗累算	
I/Oレベル	TTLコンパチ
電源電圧	5.0V
消費電力	2.5W
パッケージ	240ピンセラミックPGA
構成	0.8μm BiCMOS
ゲートアレイ	
実効ゲート数	22KG
チップサイズ	12.2mm×8.4mm

5.4 モディファイドSIMD制御技術

5.4.1 技術の位置づけ

ここでは動画処理を取り上げ、並列処理技術の一つの改善法について提案する。対象としては、画面を適当な大ききで分割してそれぞれ独立に並列処理させる、負荷分散型局所並列処理とする。負荷分散型局所並列処理に限定した理由は、以下の2点である。

<理由1>

画像処理では、原画像にいくつかの異なる処理（処理（1）～処理（n））を、次々と時系列に加えるのが一般的である。そして特に動画処理では、スループットが最優先されるため、それぞれの処理内容に合わせた最適構造の信号処理LSIを配置して機能分散型のマルチプロセッサ構造とし、全体をパイプラインで処理するのが適している。しかしそれぞれの処理の中身をみると、同じ処理を個々の画素に繰り返し行うのが一般的である。これには、画面を空間分割して負荷分散型の並列処理を行うのが適している。

すなわち、負荷分散型の信号処理LSIを単位として、機能分散型のシステムを構成することで、負荷分散の制御の容易性とシステムの高効率性を両立させることができる。このマルチプロセッサイメージを図5.4.1に示す。

<理由2>

負荷分散型のマルチプロセッサシステムに、大局処理の機能を持たせる方法に2通りある。1つは、トランスペュータのように、個々のプロセッサエレメントに通信ポートを持たせる方法であるが、例えばバタフライ演算を想定すると、プロセッサエレメント間のデータ転送のとき、介在するいくつかのプロセッサエレメントの経路による転送ネックのため、プロセッサエレメントに、1バタフライ/マシンサイクルの能力を持たせても、全体として、1バタフライステージ/マシンサイクルの性能を期待することができない。もう1つは、スイッチマトリクスや高速バスで専用の転送機能を持たせる方法であるが、この場合はハード

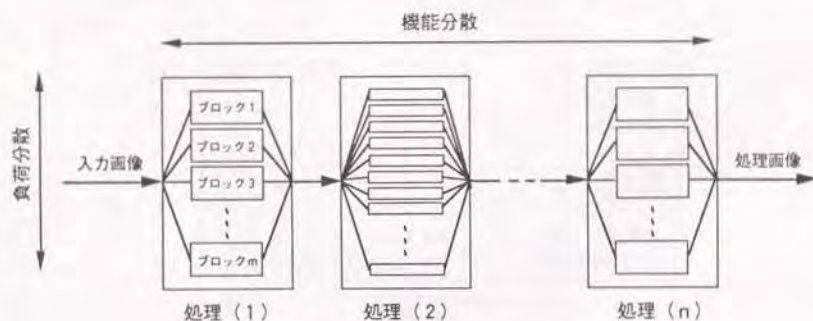


図5.4.1 動画処理のマルチプロセッサイメージ

ウェアの負担が大きいのが問題である。即ち、低コスト化が重要な実際の動画処理システムと、汎用構造での対局処理とは、現状では相いれないと言える。そこで実際上は、大局処理の種類が限られているので、目的毎に専用LSIを開発するのが現実的と言える。このことから、対局処理は専用LSIに任せ、局所処理の範囲内で、マルチプロセッサの最適な並列処理法を考える方が現実的である。

上で述べた画面分割による局所並列処理を実現する代表例である、MIMD（Multiple Instruction stream, Multiple Data stream）型とSIMD（Single Instruction stream, Multiple Data stream）型とを比較する。今日動画の帯域圧縮符号化のアルゴリズムは、適応処理を駆使する方向になりつつある。フレーム間予測符号化を例にとると、画素ブロック毎に動きベクトルを検出して、ベクトルの大きいものはフレーム間差分符号化、小さいものはフレーム内予測符号化を行うことで、画素ブロックの動きの大小に応じて、より符号化効率の高い方の予測方式を選択している。この点から考えると、画素ブロック毎に異なる処理を行えるMIMD型の方が、将来的には望ましいといえる。

しかし一方、MIMD型をLSIに実装することを考えると、制御回路が大きくなるという欠点がある。それは、プログラムシーケンサとマイクロプログラムメモリを共に個々のプロセッサに搭載しなければならないことと、プロセッサ間通信を高速に行なうデータバスおよび制御機能の搭載が必要だからである。このため、動画処理、特に画像符号化の様に、GOPS（Giga Operation per Second）級の処理能力と1チップ・低電力が共に要求されるため、今日の先端LSI技術をもってしても、処理能力の余力がほとんどない分野では、MIMDの導入は時期尚早と考えられる。

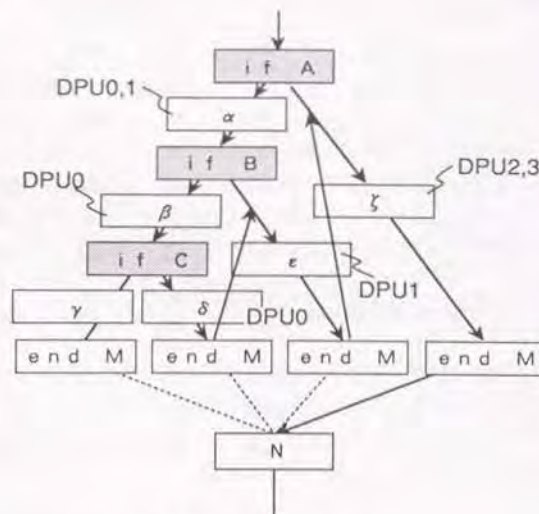


図5.4.2 プログラム例

そのため、LSI設計者の立場からみた現実的な解として、SIMD型を基本として、これに簡単な付加回路を加えて、適応処理に対応できるようにするのが望ましい。即ち、処理の大部分を占める定型処理に対しては、純粋のSIMD演算で高速処理し、時々出現する適応処理に対しては、相応のスピードで処理することにより、わずかな回路の増加で、平均的に高い処理能力を持たせるのである。その一方法として、ブランチ条件が同一のプロセッサをグループ化して、並列に処理する手法を考案した。これを、モディファイドSIMD技術と名付ける。以下この技術を、4個の演算プロセッサ(DPU0~DPU3)を搭載したビデオDSP(IDSP)の場合について説明する。

5.4.2 制御技術

図5.4.2に示す3個の条件分岐(A, B, C)からなるシーケンスを取り上げる。このシーケンスでは、条件(A)の分岐で、DPU2とDPU3は条件が成立してルーチン α へ分岐し、DPU0とDPU1は条件が成立せずルーチン α へ進む。次にDPU0とDPU1は、条件(B)の分岐命令を実行する。そしてDPU1は条件が成立してルーチン ϵ に分岐し、DPU0は条件が成立せずルーチン β へ進む。さらにDPU0は、次の条件(C)の

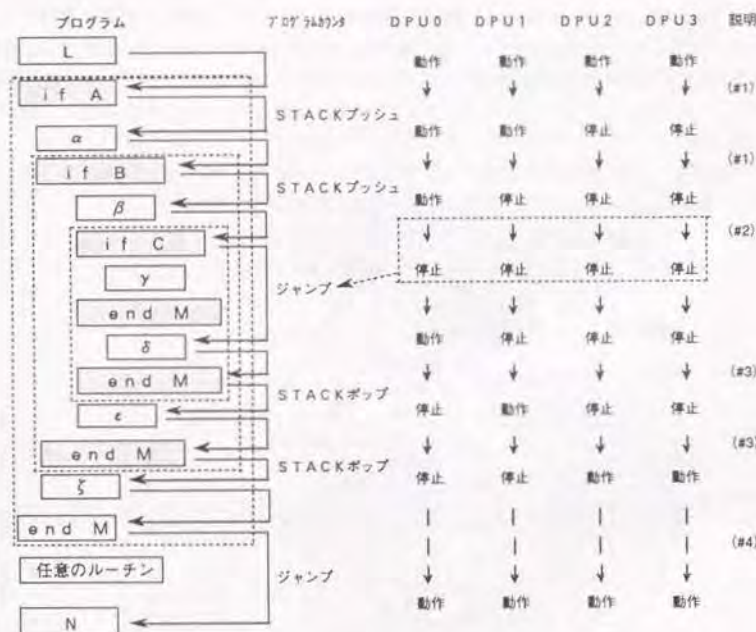


図5.4.3 モディファイドSIMD条件分岐のシーケンス

分岐命令を実行し、条件が成立してルーチン δ へ進むものとする。ここに分岐条件(A)(B)(C)は、各DPU内のフラグにより、それぞれが独立に分岐する。

さて上記分岐のプログラムと、各プログラムステップでのDPU0~3の動作状態を図5.4.3に示す。説明欄(最も右側)の番号は、このモディファイドSIMDシーケンスを実行させるのに必要な専用の命令である。以下この番号に従って、DSP全体のプログラムシーケンス(メインシーケンス)と、4個の各DPUのローカルシーケンスとが、どのように動作するかを説明する。

(#1) 分岐の中間段階：分岐(A)，分岐(B)

それぞれの演算プロセッサ(DPU0~DPU3)について、分岐条件(A)(または条件(B))で指定された演算回路のフラグを判定し、条件の成立したDPU(この場合、DPU2&DPU3(分岐条件(A)のとき)、DPU1(分岐条件(B)のとき))を停止して、ホルトフラグを立てる。このとき、すべてのDPUのホルトフラグを検証し、以下の2つの状態以外の場合、分岐の状態をスタックに記憶する。1つは、全フラグが「0」の場合であり、これは、今までに、すべてのDPUがブランチしなかったため、ブランチ処理を行う必要がないときである。もう1つは、全フラグが「1」の場合であり、今までに、すべてのDPUがどこかの分岐でブランチしたため、この後の処理を行う前に、いままでのブランチの処理を完了させたいときである。

分岐の状態をスタックへ記憶するため、具体的に、以下の2つのプッシュを行なう。1つは、条件分岐命令に記述された分岐先アドレスをプログラムカウンタのスタックへプッシュすること、もう1つは、ホルトか否かを示すホルトフラグをホルトスタックへプッシュすることである。そしてプログラムシーケンスは、その次に記述されたルーチン(図5.4.3の α または β)を続行する。

(#2) 分岐の最終段階：分岐(C)

ホルトフラグの立っていないDPU(本例の場合DPU0のみ)について、分岐条件(C)の演算フラグを判定し、条件の成立した場合、DPUを停止してホルトフラグを立てる。本例の場合、DPU0のホルトフラグが「1」に替わり、そのため、すべてのDPUのホルトフラグが「1」となるので、すべてのDPUについて、今までに、いずれかの分岐条件が成立したことがわかる。そこで最新のブランチであるDPU0が、即座に、条件分岐命令に記述された分岐先アドレスにジャンプして、分岐後の処理を実行する。このときホルトスタックは、プッシュもポップもされない。

もし、すべてのDPUのホルトフラグが「0」の場合には、条件ジャンプは発生せず、通常のインクリメントアドレスによる処理を続行し、その次に記述されたルーチンを実行する。当然プログラムカウンタのスタックおよびホルトスタックは、プッシュもポップもされない。

(#3) 分岐トリートを逆に辿って処理する段階：end(M)(ケース1)

end (M) 命令は、すべてのブランチの中の処理ルーチンの最後の行に記述されている命令であり、そのブランチ処理の最後を知らせる命令である。この命令を検出したとき、ホルトスタックの状態によって処理が分れる。もし、ホルトスタックが空でないときは、少なくとも別のブランチ処理がひとつ以上残っている（現ブランチのより根本のブランチが残っている）ことを示しているの、プログラムカウンタのスタックをポップして、次のブランチ処理ルーチンの番地へジャンプする。同時にホルトスタックをポップして、ブランチのネステイングを最新の状態に更新する。本例は、この場合を示している。

(#4) 最後の分岐を処理する段階：end (M) (ケース2)

end (M) 命令を検出したとき、ホルトスタックが空の状態になっていれば、すべてのブランチ処理が終了したことを示しているの、分岐のネステイングはすべて解消され、新しい処理ルーチン (N) にジャンプする。

以上のシーケンスは、ルーチンε、εの中にさらにべつの分岐がある場合でも、同様に拡張できる。すなわち、スタックを深くするだけで、その深さ分岐を自動的に処理することができる。プログラミングのルールは2つである。1つは、条件分岐のネステイングの階層を壊さないように記述すること。もう1つは、分岐先のルーチンの最後の行を、必ず「end 命令」で閉じることである。

次に、本手法を使って、適応処理（ブランチ処理）を行なったときの効果を見積もってみる。非常に多くの画素ブロックを、プロセッサの並列数N個のマルチプロセッサシステムで処理することとする。各画素ブロックには、複雑な適応処理を施すものとする。まずMIMDのときには、N個の画素ブロックにN個のプロセッサを割当て、それぞれを独立に、完全並列で処理することができるので、単一プロセッサの場合に比較して、理想的には、プロセッサ数であるN倍の高速化が可能である。

一方SIMDでは、すべてのプロセッサが同一のプログラムを処理するので、各画素ブロック毎に分岐が異なり、そのため、画素ブロック毎に異なる処理ルーチンに入る様な処理を並列に実行することはできない。即ち、N個のプロセッサを割り付けても、実際には、1個のプロセッサのみが動作し、残りの(N-1)個のプロセッサは、遊んでいる状態になっている。このため、シングルプロセッサと比較したときの処理時間の改善は、最悪の場合、ほとんどないと言える。

本モディファイドSIMD法の場合には、同一の分岐処理を起こした複数の画素ブロックのプロセッサをまとめて並列処理させることができる。このため高速化の改善度は、分岐の数と並列数Nに依存する。ここでは、簡単のため、2分岐一個当りの改善率を見積もってみる。N個の画素ブロックが2分岐で分れる場合は、(N+1)通りある(0とN, 1と(N-1), 2と(N-2), ..., (N-1)と1, Nと0)。このうち、

(*1) 2分岐で、2つのグループに分れなかったとき：

- ・通り数：2通り
- ・シングルプロセッサと比較した処理時間の改善：Nから1 (N倍)

(*2) 2分岐で、2つのグループに分れたとき：

- ・通り数：(N-1)通り
- ・シングルプロセッサと比較した処理時間の改善：N/2から1 ((N/2)倍)

となるので、出現確率の重みを付けた改善率Eは、以下のようになる。

$$E = \frac{N(N+3)}{2(N+1)} \quad (5.4.1)$$

一例として、4並列 (N=4) のときには、2.8倍の改善となる。

5.5 むすび

本章では、信号処理LSIの並列処理技術を取り上げ、時間方向への並列処理技術であるパイプライン制御技術と、空間方向への並列処理技術であるSIMD技術について述べた。パイプライン制御技術については、5.2節で、最初に音声DSPの具体的な設計に基づき、一般の固定パイプラインの基本技術を紹介した。次にビデオDSPを取り上げ、1つのパイプライン演算器を使って、種々のパイプラインステージを有する各種処理を統一的に処理出来るように設計した、可変型のパイプライン演算器について述べた。

5.3節では、データの値に応じてパイプラインサイクルタイムを可変とする、適応制御型のパイプライン制御技術を提案した。そしてこの技術を有効に活かすことができる、電子ビーム描画装置の副偏向制御系を取り上げ、具体的な設計法とその効果について詳述した。また電子ビーム描画装置のリアルタイム描画を高速に制御するために開発した、3種類の専用信号処理LSIを示し、リアルタイム制御系の具体的設計法について述べた。これにより、リアルタイム制御系への適応パイプライン制御技術の有効性を実証した。3種類のLSI開発で得られた成果を表5.5.1に示す。

5.4節では、定型処理と適応処理を共に高速処理できるビデオDSPの並列処理法を検討した。そして回路規模の制約から、SIMD制御法を基本として、簡単な制御回路を付

表5.5.1 開発した電子ビーム描画用LSIの性能と新技術

LSI名称	機能と性能	新技術
ショットタイミング生成LSI	1.5ns分解能でシステムの基本タイミング生成	可変サイクルパイプライン
線形行列演算LSI	副偏向補正演算を60nsのパイプラインサイクルで実行	双一次を含む行列演算のパイプライン処理
レーザカウンタLSI	ウエハの位置計算を0.01μm分解能で行う	デュアルカウンタによる主副パイプライン処理

加することで、適応処理に対処出来る手法を提案した。そしてこの手法をモディファイドSIMD制御技術と名付け、その有効性を定量的に示した。また、6.6節で詳述するビデオDSPに应用した。

5.6 参考文献

- [1] 金子、山内、"高速浮動小数点VLSIシグナルプロセッサ：DSP1"、電子情報通信学会論文誌、Vol. J72-B-I No.1, pp.67-73, (Jan., 1989).
- [2] 山内、"L S I 設計技術"、平成元年電子情報通信学会生涯教育秋季講座、(Nov., 1989).
- [3] 山内、"DSPの技術動向と応用事例"、1992年春期全国大会併催事業講習会予稿「デジタル信号処理プロセッサ」、pp.2-21, (Mar., 1992).
- [4] 南、山内、田代、鈴木、笠井、高橋、遠藤、浜口、"ビデオシグナルプロセッサI DSPのデータフロー制御"、電子情報通信学会技術研究報告、ICD91-12, pp.25-32, (Apr., 1991).
- [5] H.Yamauchi, T.Morosawa, T.Watanabe, A.Iwata, T.Hosaka, "Real-time Feed-Forward Control LSIs for Direct Wafer Exposure Electron Beam System", IEICE Trans. Electron., Vol. E76-C, No.1, pp.124-135, (Jan., 1993).
- [6] M.Fujinami, N.Shimazu, T.Hosokawa, and A.Shibayama, "EB60D: An Advanced Direct Wafer Exposure Electron-Beam Lithography Systems for High-Throughput, High-Precision, Submicron Pattern Writing", J.Vac.Sci.Technol.B 5(1), pp.61-65, (Jan./Feb., 1987).
- [7] T.Watanabe, T.Morosawa, N.Shimazu, H.Morita, H.Yamauchi, A.Iwata, "Development of Reliable and Comprehensive Direct Wafer Exposure Electron Beam System", The 35th international symposium on electron, ion & photon beams, R6, (May., 1991).
- [8] T.Morosawa, A.Shibayama, T.Murashita, and M.Fujinami, "A High-Speed Patterning Controller for the EB60 Electron-Beam Lithography System", J.Vac.Sci.Technol.B 5(1), pp.66-69, (Jan./Feb., 1987).
- [9] N.Shimazu, T.Morosawa, H.Morita, "A High-Throughput Direct Wafer Exposure Electron-Beam Lithography System: EB60", Review of the Electrical Communication Laboratories, Vol. 36, No.3, pp.343-349, (Mar., 1988).
- [10] M.Suzuki, S.Konaka, H.Ichino, and S.Horiguchi, "Design and Application of a 2500-Gate Bipolar Macrocell Array", IEEE Journal of SC., Vol. SC-20, pp.1025-1031, (1985).
- [11] H.Fukuda, S.Horiguchi, M.Urano, K.Fukami, K.Matsuda, N.Ohwada and H.Akiya, "A BiCMOS Channelless Masterslice with On-chip Voltage Converter", ISSCC digest of technical paper, 176, (Feb., 1989).
- [12] Y.Tashiro, H.Yamauchi, T.Minami, S.Hamaguchi, Y.Suzuki, "An Advanced SIMD Control Technique for Concurrent Adaptive Processing", Proceedings of JTC-CSCC'91, pp.252-255, (Jul., 1991).

第6章 L S I技術制約の解決法

本章では、L S Iの集積規模とゲート速度の制約を解決する技術について検討する。L S Iを実際にインプリメントするときの効率化は、種々の設計段階で行われている。最上位はシステム設計のレベルである。このレベルでは、ターゲットのシステムを実現する方法として、例えば、数種類の専用L S Iのチップ分割か、または標準の汎用DSPの並列動作か、というような選択が行われる。6.1節では専用L S Iの利害特質について検討する。

その次のレベルは、アーキテクチャレベルである。ここでは、要求される処理能力と実現できるL S I技術との兼ね合いで、演算器やバスやメモリの汎用性、並列性、等様々な工夫がこらされる。本論文では、L S Iの面積に大きな影響を及ぼすため問題となっている、オンチップメモリの占有面積削減を実現する有力な方法として、必要なメモリ容量そのものを削減する技術を取り上げ、6.5節で述べる。そして6.6節では、この技術を応用して開発した110KG規模のVLSIであるビデオDSPを紹介し、6.5節で述べる技術の有効性を示す。

さらに下位のレベルでは、演算器の高効率化がある。一般に演算器の高速化を図ろうとすると、ブロックレベルまたはゲートレベルで、何らかの並列処理を行うため、その分、回路規模は必ず増大する。そのため、演算器が占有する単位面積当りの性能の点から見ると、却って悪くなる場合が多々ある。極端な場合、同じ演算器を複数個並列に並べた方が良い場合もある。このため、回路規模(占有面積)の制約が大きい場合はもちろん、占有面積当りの最適設計を行なう場合に、不利となり、実際使用されていない高速化技術もある。そこで本論文では、上記の中から代表的な2つの技術を取り上げ改善手法を提案する。1つは冗長2進加算器であり、もう1つは、通常2の補数での並列乗算器に関するものである。それぞれ、6.2節と6.4節で、その回路規模削減法を述べる。また、6.2節での成果を応用して開発した、冗長2進加算器アレイによる幾何学変換L S Iを6.3節で紹介し、回路規模削減技術の有効性を示す。

表6.1.1 DSPと専用L S Iの得失

項 目	DSP (プログラム論理)	専用L S I (布線論理)
開発の並行性	ターゲットシステム DSPチップ 搭載プログラム ⇒ 均等	各L S I ⇒ 論理に集中
必要なツール	多くの専用開発ツールが必要 DSPエミュレータ プログラム開発、デバッグ環境	強力な論理シミュレーション環境 システムデバッグ環境
L S Iテスト ベクトルの特徴	種類 : 多い ベクトル長 : 短い ⇒ 論理シミュレーションが容易	種類 : 少ない ベクトル長 : 長い ⇒ 論理シミュレーション困難
性能、およびバグ 許容力	多少ある (マルチプロセッサ、プログラム)	ない
性能/回路規模 比	低い	高い

6.1 専用LSI化の利点[1]

専用LSIは大きく、ASIC (Application Specific IC) とASSP (Application Specific Standard Product) に分けることができる。前者は単一の機能を最も効率よくハードウェア化したものであり、純布線論理で構成されているものが多いが、最近では、機能が複雑になってきたため、プログラムROM内蔵のマイクロプログラム制御のものもある。いずれにしても、LSIのプロセス条件が同一ならば、最も高速で最も低コストとなる実現方法である。後者は特定の狭い領域で共通的に使える標準品であり、汎用性の程度により、布線論理のもの、マイクロプログラム制御のもの、マイクロプロセッサ内蔵のプログラム制御のもの等種々の構成がある。

本論文では、ASICとして、電子ビーム直描装置のリアルタイム制御用に開発したLSI 3品種を5.3節で取り上げた。またASSPとしては、画像処理の標準機能である2次元FFT-LSIを4.5節で、すでに取り上げた。さらに別のASSPとして、冗長2進加算器アレイから構成した幾何学変換LSIを6.3節で取り上げる。

これら専用LSIを、汎用のDSPと比較した利害得失を表3.1.1にまとめて示す。専用LSIは、LSI自体の開発コストがDSPより安く、高速で、かつ回路当りの性能にすぐれているので、LSIの技術制約を最初に切り開いていくLSIである。しかしその反面、システムデバックに時間がかかること、設計バグに対する許容力に乏しいこと、性能が要求値を満たさないとき改善のしようがなく、致命的になる等の危険を包含しているので注意が必要である。

6.2 冗長2進演算器の回路規模削減技術

冗長2進では、1桁を3値で表現するため、表現が一意に定まらない。この表現の自由度を積極的に使用して、下位桁からの桁上りを完全に吸収し、桁上りが伝搬を抑さえることで、高速加算を可能にしたのが、冗長2進加算器である。高速な反面、3値を使った冗長表現のため、回路規模が大きい欠点がある。また、冗長2進表現から通常の2進表現に戻すときにキャリー伝搬を生じるため、応用として、通常2進系への変換頻度の低い用途を選択しなければならない。

6.2.1 冗長2進表現[2][3]

一般に冗長2進では、各桁を $\{-1, 0, 1\}$ の3値で表現する。すなわち、冗長2進で表現された次の数字

$$[0, x_1 x_2 \dots x_n]_{SD2} \quad (x_i \in \{-1, 0, 1\}) \quad (6.2.1)$$

を通常の2進で表現すると、

$$\sum_{i=1}^n 2^{-i} \cdot x_i \quad (6.2.2)$$

という値を表す。

冗長2進数体系の最大の長は、2進の加(減)算での、桁上げ(桁借り)の伝搬をなくすることができることである。この原理を図6.2.1に示す。2つの冗長2進数PとQの加算を、次の2つのステップに分解して実行する。ステップ1では、各桁ごとに被加数の値

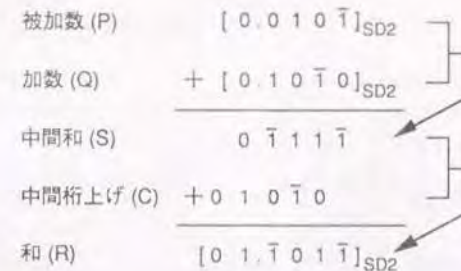


図6.2.1 冗長2進加算で桁上げが伝搬しない原理

表6.2.1 ステップ1での計算規則

被加数 (p_i)	加数 (q_i)	1つ下位の桁 (p_{i-1}, q_{i-1})	中間桁上げ, 中間和 (c_i, s_i)
1	1	-----	1, 0
1	0	両方とも非負	1, 1
0	1	上記以外	0, 1
0	0	-----	0, 0
1	1	-----	0, 0
1	0	両方とも非負	0, 1
0	1	上記以外	1, 1
1	1	-----	1, 0

p_i と、加数の値 q_i とから、

$$p_i + q_i = 2c_i + s_i$$

(6.2.3)

となるように、中間桁上げ c_i と中間和 s_i を表6.2.1に従って求める。1つ下位の桁の加数と非加数の値を見て、下位からの桁上げ（キャリー）を予測し、それを吸収できる様に、中間和 s_i を定め、その後、(6.2.3)式を満足するように、中間桁上げ c_i を定める。ス

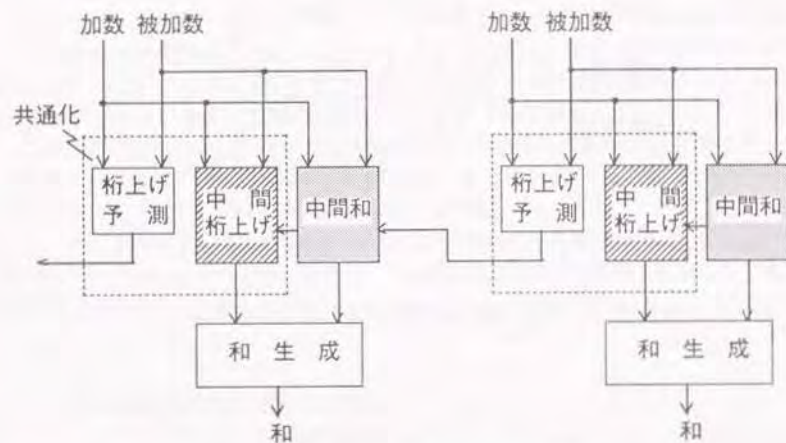


図6.2.2 冗長2進加算回路の基本構成

表6.2.2 冗長2進コードの2値符号化法

冗長2進表現値	2進表現値	
	S bit	V bit
0	ϕ	1
+1	0	0
-1	1	0

ϕ : don't care



C : -1のキャリーが生じない
 \bar{C} : 1のキャリーが生じない
 ϕ : don't care

図6.2.3 新しいコード化による回路ブロック共通化の原理

は、各桁ごとに s_i と 1 つ下位の桁からの中間桁上げ $ci-1$ とから和 ri を、

$$r_i = s_i + ci-1 \quad (6.2.4)$$

によって求める。このときあらたな桁上げは生じない。また、2 つの冗長 2 進数の減算を行うときは、減数の正負を反転させた後加算を行う。以上により、組み合わせ回路による 2 数の加減算が、桁数に関係なく一定時間でできる。即ち、通常 2 進では実現できない高速加算が可能である。

しかしながら先にも述べたように、冗長 2 進加減算器は、回路規模が大きくなる欠点を有している。その理由は、本節での冗長 2 進表現の説明から明らかであるが、以下の 2 点である。第 1 は、1 桁を 3 値で表現するため 1 桁当たり通常 2 進の倍の 2 ビットを必要とするため。第 2 は、加算のプロセスでステップ 1 とステップ 2 の 2 段階を要し、それぞれのステップで通常 2 進の加算に相当する処理を必要とするためである。即ち、最悪の場合、通常 2 進加算器の 4 倍の回路規模になる可能性があり、これでは実用に供することが難しい。

以上より、冗長 2 進加減算器の最大の課題は、回路規模の削減であるといえる。

6.2.2 従来冗長 2 進加算器の構成[4][5]

前節で述べた加算原理に基づく、冗長 2 進加算器の構成モデルを図 6.2.2 に示す。各桁がそれぞれ 4 つの機能ブロックから構成されている。ステップ 1 では、(#1) 桁上げ予測、(#2) 中間和生成、(#3) 中間桁上げの 3 回路ブロックを使用し、ステップ 2 では、最終的な (#4) 和生成を行う。この 4 機能は冗長 2 進演算に必須のものである。従来から、ゲートレベルでの回路規模削減の手法が取られ、上記 4 機能ブロックの回路規模が削減されてきた。しかしそれでも、通常 2 進回路に比べて 2 倍程度の回路規模になっている。

6.2.3 回路規模削減手法

回路規模削減の新しい考え方として、上記 4 回路ブロックの論理を、できるだけ共通化することを試みた。即ち、冗長 2 進表現では、2 ビット使って 3 値を表現しているため、2 値符号の割当に自由度が生じることに着目し、この自由度を活かして回路の共通化を図るのである。

新しい冗長 2 進コードの 2 値符号化法を表 6.2.2 に示す。本構成では、2 ビットを符号ビット (S) と、値ビット (V) とで表現する。そして S ビットに着目すると、冗長 2 進の加算原理より、冗長 2 進の「0」で表わされた値は、2 進の「0」、「1」どちらでも良いので「don't care」とすることができる。このため、桁上げ予測回路、中間桁上げ回路、中間和回路の (S) ビットは、図 6.2.3 に示すように簡単にすることができる。このため以下の 2 つの利点が生じる。第 1 点は、桁上げ予測回路と中間桁上げ回路を共通化できること。第 2 点は、中間和回路が加数と被加数の入力に依存しないため、桁上げ信号による選択回路だけで構成できることである。

以上の考え方の基づく共通化の概念を、図 6.2.4 に示す。またこの原理に基づいて、

CMOS ロジックで構成した冗長 2 進加算器の 1 ビット分の回路を図 6.2.5 に示す。この回路では、セレクト回路を素子数の少ない TG ゲート (トランスファゲート) を使って構成したので、いっそう素子数が削減され、トランジスタ数を 52 としている。これは、通常 2 進の、4 ビット単位の CLA 回路 (Carry Look-ahead Adder) と比較しても、僅かに、17 % 程度の増加に過ぎない。

さらに図 6.2.2 から明らかなように、冗長 2 進加算器は、1 桁を単位とする非常に規則性の高い構成のため、レイアウトを高密度に行なえる利点がある。そのため、実際に L

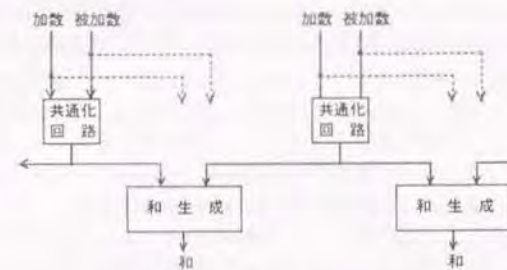


図 6.2.4 回路規模削減に適したコードによる冗長 2 進加算回路の構成

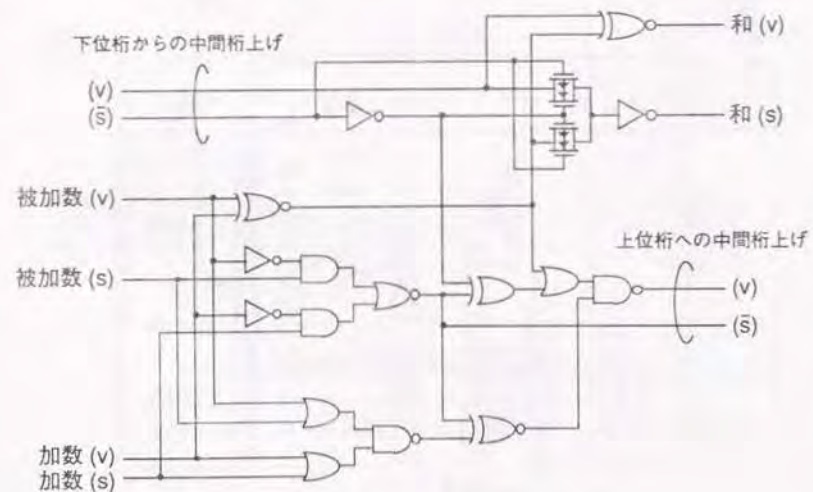


図 6.2.5 回路規模を削減した冗長 2 進加算器の回路図

SI化したときの占有面積と比較すると、通常2進のCLA回路とはほぼ同程度とすることができる。

図6.2.6に、考案した冗長2進加算器とCLA加算器との性能を示す。比較のために、具体的に使用したプロセスは、 $1.2\mu\text{m}$ -CMOS技術である。演算速度の点では、冗長2進加算器は、演算語長に無関係に 5.5ns と高速であり、語長が長くなるに従い有利となる。一例として、本冗長2進加算器をパイプライン加算器アレイとして応用した幾何学変換LSI（6.3節で示す）を取り上げると、要求演算語長が20ビットなので、CLAと比較して、約1.6倍高速である。一方回路の占有面積はほぼ同等である。

以上本節では、冗長2進加算器の回路規模削減法について述べた。冗長2進コードの2値符号割当法を工夫することにより、通常2進のCLA回路とはほぼ同程度まで削減できることを示した。これにより、冗長2進加算器を実用に供することができるようになった。

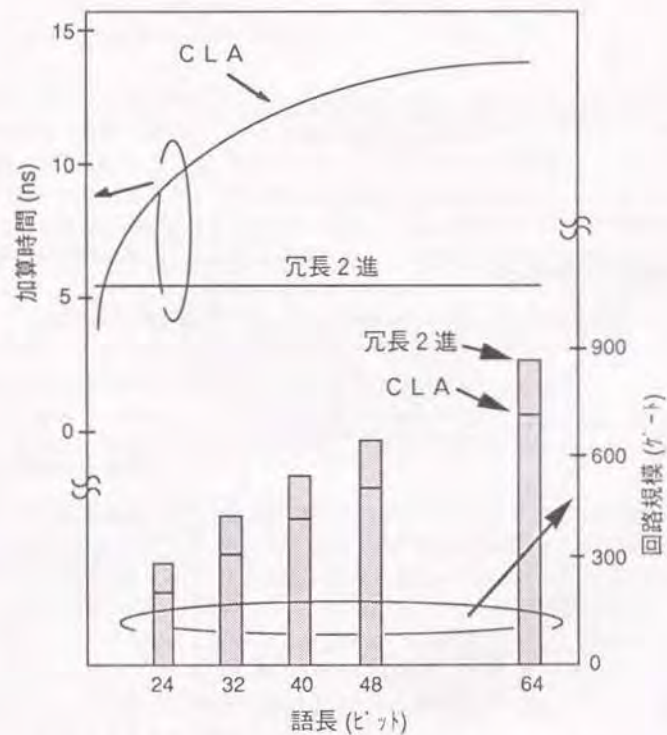


図6.2.6 冗長2進加算器と2進CLA加算器との比較

6.3 冗長2進を活かした幾何学変換LSIの構成法

6.3.1 本LSIの位置づけ

ここでは、前節で提案した冗長2進加算器を使った高速LSIについて述べる。題材として、高精細動画の幾何学変換処理をリアルタイムで行う専用LSIを取り上げる。幾何学変換とは、2次元のデータ配列である画像（フレームデータ）に幾何学変換関数を用いて写像を施し、形状変換させた別の画像を生成することであり、コンピュータマッピング[6]、ビデオエフェクタ、CG(Computer Graphics)、パターンマッチング等、画像処理の広い分野で使用されている。この幾何学変換用に開発したLSIは、幾何学変換の中核処理である変換座標の計算（フレームデータへのアドレス）を高速に行なうものである。

各種幾何学変換の例を図6.3.1に示す。アフィン変換は、座標の変換関数が1次関数と最も次数の低い簡単な変換であり、図形の縮小、拡大、移動、回転等広く使用されている。射影変換は、3次元CGのディスプレイ面への透視に欠かせない変換であり、また球面変換は、人口衛星で取った地球表面写真の球面補正に使われている。また当然、これら関数を組み合わせてより高度な変換を行ない、映像効果を高めることも行なわれている。

以上の状況を考えると、幾何学変換LSIの機能として、用途に応じて自在に種々の変換関数を生成できることが望ましいと言える。また性能として、今後マルチメディア時代を迎えることを考えると、高精細の動画を処理できる能力を有することが望まれる。しかしながら、従来の幾何学変換LSIは、変換関数種をアフィン変換や等角射像等の低次の関数に限っている状況であり、処理速度についても、現状の放送方式であるNTSCレベルの画像に留まっている。

変換種	アフィン	射影	球面
イメージ			
(*) 変換関数	$U = aX + bY + c$ $V = dX + eY + f$	$U = \frac{aX + bY + c}{pX + qY + r}$ $V = \frac{dX + eY + f}{pX + qY + r}$	$U = \frac{rX}{\sqrt{r^2(X^2 + Y^2)}}$ $V = \frac{rY}{\sqrt{r^2(X^2 + Y^2)}}$

(*) 座標変換: $(X, Y) \rightarrow (U, V)$

図6.3.1 幾何学変換関数の例

そこで、本幾何学変換LSIは、画素当りの処理スピードを50MHz以上(20ns/画素)とし、1チップで高精細動画像の低次幾何学変換に対応できるようにした。さらに、マルチチップのチップ間パイプライン構造をとれるようにし、高次の関数を、低次関数のカスコード接続で計算できるようにした。即ち、チップ数を増やすことで、50MHzのスループットで高次の幾何学変換関数を生成することができる。この高速・高機能の幾何学変換LSIを、GMP-1 (Geometrical Mapping Processor-1) と名付ける[7]。

6.3.2 高速化の設計方針

幾何学変換は、変換前のすべての画面の値を、変換後の画面のそれぞれの対応する位置に移動させる処理である。そして特に、動画像(ビデオ)へ応用するには、画素単位のパイプライン処理ができるようにしなければならない。具体的には、変換画素の座標の計算(正確には、座標に対応するメモリ上のアドレスの計算)と、計算された座標にある画素データの転送の2つの処理を、パイプラインで連続的に行わなければならない。先にも述べたが、本幾何学変換LSIは、上記2処理の内、アドレス計算を行うものである。

ビデオデータは、ラスターキャンに従って入出力されるので、幾何学変換後の画素アドレスをラスターキャンし、対応する変換前の画素アドレスを生成するようにした方が便利である。例えば図6.3.1の例では、 (X, Y) がラスターキャンされる変換後のアドレスであり、 (U, V) が変換前のアドレスである。

また今後の方向として、デジタル動画像の高精細化が進み、画面サイズが大きくなる。このためアドレス空間が大きくなり、従って、アドレス語長は長くなる。また、扱われる幾何学変換の種類も多種多様なものとなって行くと考えられる。このため、長語長の高速演算能力を備えることと、多様な関数種に対応できる柔軟なアーキテクチャとすることの2点に主眼を置くこととした。

変換関数の計算を高速に行う方法として、初等関数を生成多項式に展開して、近似計算で求める方法が広く用いられている。しかしこの方法は、高精細画像処理に適用する点から

繰り返し計算の式

$$\begin{cases} X_{i+1} = X_i + mY_i \cdot 2^{-k} \\ Y_{i+1} = Y_i - X_i \cdot 2^{-k} \\ Z_{i+1} = Z_i + S_i \end{cases}$$

K : シフト数
m : ファンクションモード
S₁ : 予め計算しておく定数

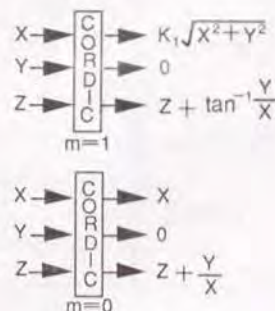


図 6.3.2 CORDIC 演算による初等関数の生成法

見て、以下の2点で不都合である。第1点は、精度を一定としたとき、生成関数種によって必要な多項式次数が異なるため、演算器数を一定としたとき、関数毎に生成速度が変化することである。このため中間結果を格納するメモリを必要とするが、高精細画像では1画面あたり数Mバイトを要するため、実用的な点からは望ましくない。第2点は、演算器を、乗算器とALUとからなるパイプライン積和演算器で構成することになるが、乗算器と加算器とでは演算速度が異なる。このため、パイプラインサイクルをデバイス性能の限界まで上げて高速化するのが容易でないことである。結論として、上記欠点を補う方法、即ち、パイプラインの高サイクル化と、種々の関数を生成する上での制御の容易さとを兼備えた演算方法が望ましい。

その方法として、幾何学変換関数には、乗除算、三角関数、開平、双曲線関数の初等関数を選び、これらの初等関数を、定数の読みだし、シフト、正負判別、加減演算の一定回数の繰り返し処理で生成できる、Waltherの統一型CORDICアルゴリズム[8][9]を採用した。このアルゴリズムを図6.3.2に示す。

また、CORDIC演算の基本演算単位である加減算を高速化するために、 $|+1, 0, -1|$ の3値からなる冗長2進加算器を導入した。これについては前節で詳述した。本LSIでは、この冗長2進加算器をアレイ状に配置して、CORDICアルゴリズムに基づ

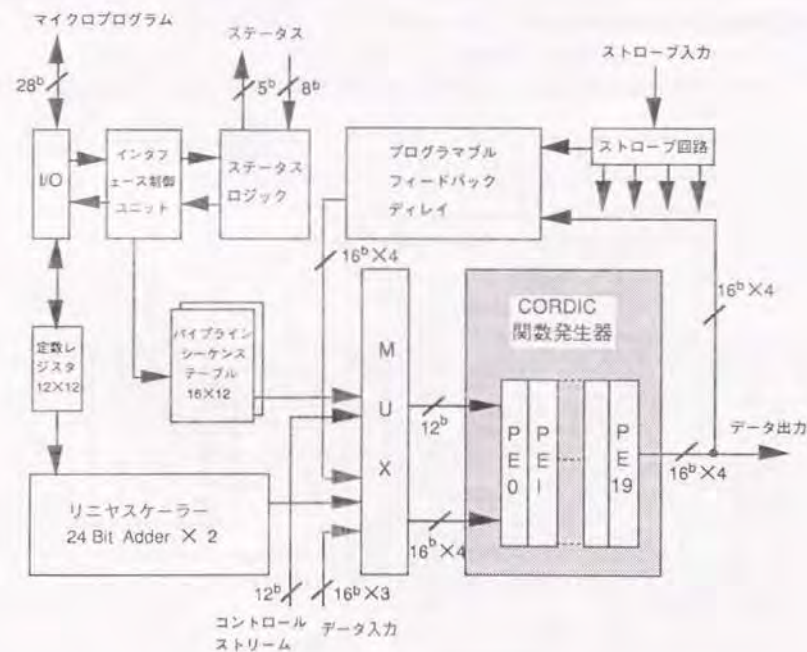


図 6.3.3 幾何学変換 L S I の全体図

さらにマルチチップ化したとき、各チップのCORDIC関数発生器をカスコード接続することにより、各チップで発生した関数の組み合わせ関数を生成できるようにした。これにより、最小構成の1チップの場合には、繰り返し処理することで、高度な複合関数を生成できるし、チップを複数個使ってパイプライン化すれば、上記複合関数をリアルタイムで発生できるようにする。この機能は、インテリジェント端末やワークステーションへ適用したときに、コストと性能見合いで種々の選択ができることを狙いとしている。既存の幾何変換用LSIとしては、アフィン変換用(5MHz動作)[10]と、線形2次関数用(アフィン変換換算で20MHz動作相当)[6]が発表されている。本LSIはこれらに比較して、生成速度だけを見ても、2.5~10倍高速(50MHz動作)となることを目標とした。

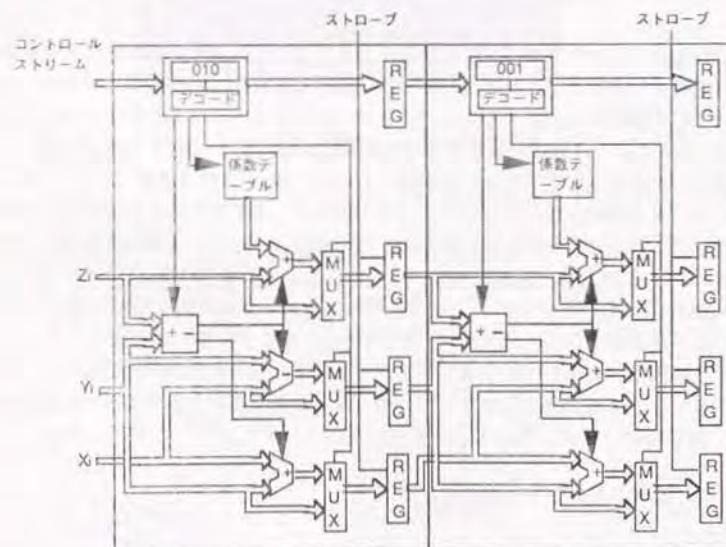
以上の方針に基づき設計したS Iの構成を図6.3.3に示す。CORDICによる初等関数発生器(CORDIC関数発生器)、アレー入力マルチプレクサ(MUX)、可変長ディレイ(プログラマブルフィードバックディレイ)、線形演算による係数生成用累算器(リニヤスケール)、パイプライン演算設定部(パイプラインシーケンステーブル)、およびチップ制御部(インタフェース制御ユニット)から構成されている。チップ制御部は、発生関数種とその組合せ方法をユーザが指定できるようにするため、マイクロ命令をホストからダウンロードして起動する構成にしている。

Figure 1 is a block diagram of the control stream of the first embodiment. The diagram is divided into two main sections: "コントロール ストリーム" (Control Stream) on the left and "ストロブ" (Strobe) on the right. The control stream contains several functional blocks: a "COEF TABLE", a "COMP" (Comparator), two "SHIFT" blocks, and two "REG" (Register) blocks. The strobe section contains four "REG" blocks. The diagram illustrates the flow of data and control signals between these components. Key equations shown are $Z_{i+1} = Z_i + S$, $Y_{i+1} = Y_i \cdot X_i 2^{-K}$, and $X_{i+1} = X_i + m Y_i 2^{-K}$. A detailed view of the storage stream shows four registers (P, P, P, P) and their corresponding data (E, E, E, E) and addresses (0, 1, i, 19).

-154-

関数発生器は、CORDICの反復演算を1ステージとする1次元パイプラインアレーとした。X、Y、Zの3個のデータストリームそれぞれが、CORDICの6関数モードに応じた加減演算制御を独立に行う。このMISD型の配列演算器の構成を図6.3.4に示す。演算語長とパイプライン段数は、HDTVで要求される演算精度をもとに、Waltherのアルゴリズムを解析して決定した。アレー内加算器の演算語長は20ビット、CORDICの演算ステージの段数は19段である(PE0~PE18)。図中の最終段(PE19)は、冗長2進系から通常2進系への変換のための加算器である。

また、複数チップをパイプライン状に接続して組み合わせ関数を生成できるようにする



-155-

ため、パイプライン演算部に外部入出力を設けた。この複数チップでの生成のときには、生成の絶対遅延は組み合わせ関数の次数に比例して増大するが、生成レートは関数に拠らずに一定で、パイプラインの各サイクル毎に高速生成される。

6.3.4 高速化技術

(1) 冗長2進技術の採用[5]

6.2節で述べたS(符号)とV(値)の2ビットによる2値符号化に基づき、加算器を冗長2進で構成した。回路規模は従来のCLAに比較して17%程度増大するが、ディジット単位の同一回路で構成できるので、規則性の高いレイアウトができる。このため占有面積は、CLAを使った場合と比べてほぼ同程度にできた。

生成されたアドレスは、外部へ出力され、画像データが格納されているフレームメモリのアクセスに使用される。このとき、メモリのアドレスは、通常2進系でなければならないので、チップ内部で、冗長2進を通常2進系に変換して出力しなければならない。通常2進系の加算器は、冗長2進加算器の2倍近い遅延を生じるので(語長20ビットで高速加算器を使用したとき)、2段のパイプラインで分割して、CORDIC関数発生器の最終段に配置した。この加算器としては、キャリーセレクト加算器を用いた。この加算器は、図6.3.3の(PE19)に配置されている。

(2) コントロールストリームによる演算制御方式[11]

CORDICの各ステージでの演算モードを高サイクルで制御する方式を考案した。これは、関数モードをタグとして演算データに同期させて転送し、各パイプラインステージでデコードして制御する方法であり、コントロールストリーム方式と名付けた。この構成を図6.3.5に示す。この例は、乗算と開平の組み合わせ関数を生成する場合を示している。隣接する2つのステージで、それぞれ乗算(コード:001)と開平演算(コード:010)を実行している。それぞれ、コードをデコードして演算モードを知り、その演算モードに従って、係数テーブルから必要な係数値を読みだして、加算器へ投入する構成である。この例からわかるように、本方式は、高速制御上、以下の2点の特長を有する。

- (A) 演算制御をパイプラインクロックのみで行えるため、各ステージ間のクロックスキューを低減できる。このため、パイプラインサイクルを高速化できる。
- (B) 生成関数に対応して行う各パイプラインステージでの制御を、関数モードタグのみで行える。このため、1チップでの繰り返し演算で組み合わせ関数を生成する複雑な制御も容易に行え、従って、制御用の追加ハードウェアも必要としない。

6.3.5 幾何変換LSIの試作結果[12]

CORDIC関数発生器の部分は、LSIの中で最も大きな領域を占めているが、冗長2進加算器を使っていることと、パイプライン構成であることから、極めて規則性が良い。

このためマニュアル設計を行った。この部分の集積密度は、 $1.2\mu\text{mCMOS}$ ルールで、 $3.5\text{K素子}/\text{mm}^2$ であり、十分な結果を得た。シーケンス制御部は、RTL(Register Transfer Logic)記述から論理合成を行い、スタンダードセル設計をおこなった。全回路規模は、純ロジックで約100Kゲートであり、チップ面積は 14.5mm^2 である。LSIのチップ写真を図6.3.6に示す。また代表的な諸元を表6.3.1に示す。 $1.2\mu\text{mCMOS}$ にも拘わらず、最高70MHzの高速動作を達成し、冗長2進技術とコントロールストリーム技術の有効性を確認した。

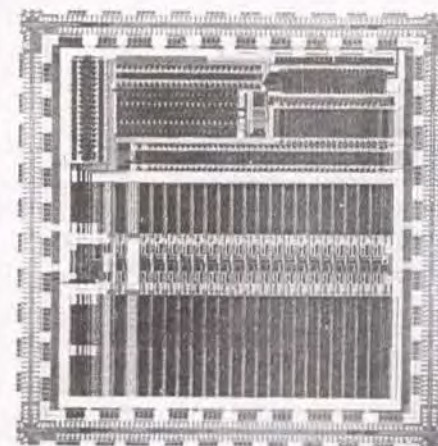


図6.3.6 幾何変換LSIのチップ写真

表6.3.1 幾何変換LSIの諸元

動作周波数	> 50MHz
アフィン変換	20 ns / pixel
射影変換	40 ns / pixel
	20 ns / pixel using two chips
ゲート規模	105 K gates
プロセス	$1.2\mu\text{mCMOS}$
チップサイズ	$14.5 \times 14.5\text{mm}$
消費電力	4 W at 50 MHz
動作電圧	5 V
I/Oインターフェース	TTL
パッケージ	208 / 280 PGA

6.4 並列乗算器の回路規模削減技術

本節では、並列乗算器の部分積加算部のハードウェア規模削減技術について述べる。並列乗算器の部分積加算では、指定桁数の異なる部分積の加算を行わなければならない。従来この加算は、図6.4.1の上図に示す様に、すべての項の符号ビットを最上位の符号ビットの桁まで拡張して行っていたため、符号ビット拡張部に余分な加算回路を要していた。そこで、すべての項の符号ビットを反転し、最下位の項の符号ビットの桁と、各項の符号ビットと符号ビットの間のすべての桁とに1を加算することにより、等価な結果が得られることに着目し、これを「1加算法」と名付けて採用し、回路規模の大幅な減少を達成した。以下1加算法の等価性の証明を示す。また拡張データ入力を受けて拡張乗算器を構成できるように、機能を拡張した場合については、第3.1節で詳述した。

(A) 1加算法が符号ビット拡張と等価であることの証明[13][14]

8n×8nビットの乗算を2次Boothのアルゴリズムで行った場合について行う。このとき、2ビットづつシフトした4n個の部分積PPi(i=0,1,2,...,4n-1)の加算を行うが、各部分積の符号ビットを(16n-1)桁まで拡張して行う場合と、1加算法で行う場合とが等しくなることを以下証明する。

i番目の部分積PPiを

$$PP_i = \left(a_s^{(i)} \cdot 2^{8n} + \sum_{k=0}^{8n-1} a_k^{(i)} \cdot 2^k \right) \cdot 2^{2i} \quad (6.4.1)$$

で表す。as(i)は符号ビットの値、ak(i)はその他の各ビットの値である。符号ビット拡張法に基づく第1項の和をS0とすると、

$$S_0 = \sum_{i=0}^{4n-1} \left(\sum_{k=0}^{8n-2-2i} a_s^{(i)} \cdot 2^k \right) \cdot 2^{8n+2i} \quad (6.4.2)$$

となる。符号ビットの反転信号as(i)=1-as(i)を用いて式(6.4.2)を変形すると、

$$\begin{aligned} S_0 &= \sum_{i=0}^{4n-1} (1 - a_s^{(i)}) \cdot 2^{8n-2i-1} \cdot 2^{8n+2i} \\ &= \left[\sum_{i=0}^{4n-1} (1 - a_s^{(i)}) \cdot 2^{16n-1} \right] \\ &\quad + \left[\left(2^{8n-1} \cdot \sum_{i=0}^{4n-1} 2^{2i} \right) + \sum_{i=0}^{4n-1} a_s^{(i)} \cdot 2^{2i} \right] \cdot 2^{8n} \end{aligned} \quad (6.4.3)$$

となる。式(6.4.3)の第1項は16n桁以上なので無視し、第2項を

$$2^{8n-1} = 1 + \sum_{i=0}^{8n-2} 2^{2i}$$

を用いて変形すると、

$$\begin{aligned} S_0 &= \left[\left(1 + \sum_{i=0}^{8n-2} 2^{2i} \right) \cdot \sum_{i=0}^{4n-1} 2^{2i} + \sum_{i=0}^{4n-1} a_s^{(i)} \cdot 2^{2i} \right] \cdot 2^{8n} \\ &= \left[\left(1 + \sum_{i=0}^{4n-2} 2^{2i+1} \right) + \sum_{i=0}^{4n-1} a_s^{(i)} \cdot 2^{2i} \right] \cdot 2^{8n} \end{aligned} \quad (6.4.4)$$

となり、1加算法に基づく演算結果と一致する。式(6.4.4)の第1項は、1を加算する項であり、第2項は、符号ビットの反転に基づく項である。

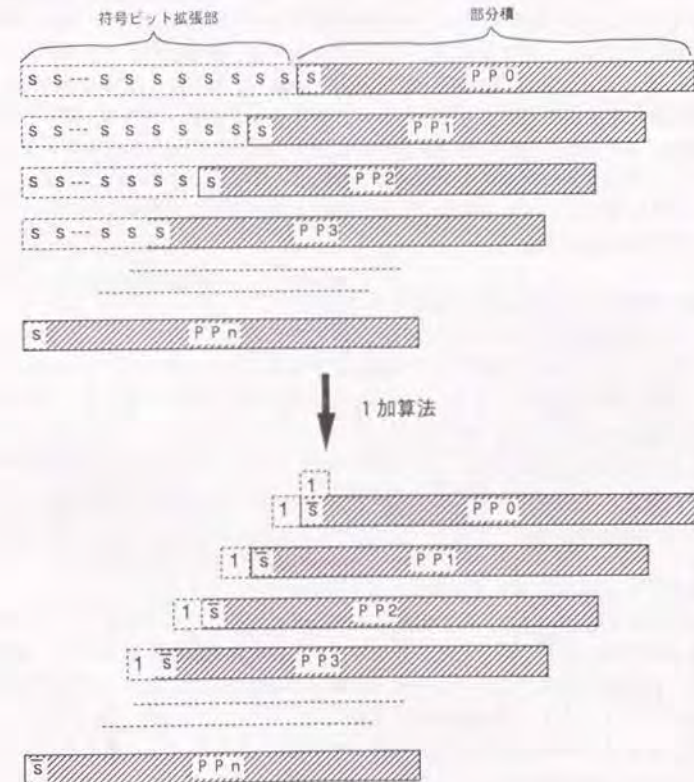


図6.4.1 2の補数型並列乗算器の部分積の加算部

6.5 オンチップメモリ容量削減技術

DSPでは、オンチップのプログラムメモリとデータメモリが大きな領域を占めている。このため、メモリ容量の削減は、LSI化の重要な課題である。ここでは、プログラムおよびデータメモリの容量削減技術について述べる。まずプログラムメモリでは、並列プロセッサにおける2階層マイクロプログラム制御技術を提案する。データメモリでは、データのコンパクションによる削減法と、デュアルポートのメモリセルを高密度に構成し、結果的に容量削減と同等の効果のある技術について述べる。

6.5.1 並列プロセッサにおける2階層マイクロプログラム制御[15][16]

DSPでは、パイプライン演算器をマイクロプログラムシーケンサで高速に制御している。このマイクロプログラム制御方式は、ハードワイヤで構成したシーケンサと比較して、制御回路が簡単な反面、制御されるデータパスが複雑になるに従って、マイクロ命令のフィールド長が長くなり、それに比例してプログラムメモリ容量が大きくなる問題を有している。特にビデオDSPのように、高い処理能力を必要とするLSIでは、必然的に、パイプライン演算器を複数個並列動作させることになるため、並列度に比例して制御しなければならない対象が増大する。このため従来技術をそのまま延長して使うと、非常に長いマイクロ命令となり、オンチップ化が困難となる。

従来マイクロ命令フィールドを短くするために、種々の方法が提案されている。その中で以下の2手法が代表的である。

(#1) 制御の頻度により制御対象を区分する方法：

あるベクタ演算を想定したとき、制御を固定して良い機能ブロックと、各命令サイクル毎に制御しなければならない機能ブロックに分類できることに着目して、2種の命令に分割する手法である。制御を固定して良いものには、SET-UP命令を使って、ベクタ演算開始時に状態を指定する。その後、各命令サイクル毎に制御が必要なものを、EXEC命令で制御するのである。通常、SET-UP命令の方がEXEC命令より出現頻度が低いので、その分、プログラムメモリ容量を削減できる。この方法のペナルティは、SET-UP命令の分だけ、処理スループットが低下することである。

(#2) 2階層マイクロプログラムによる方法：

1つのプログラムの中では、同じ処理ルーチンが再起的に使用される。この性質を使った効率的なプログラミング手法として、サブルーチン技術が良く知られている。通常、サブルーチンコール命令と対応するサブルーチンの中身の命令群とは、同一のプログラムシーケンサでの命令セットであり、命令長も等しい。

2階層プログラミング手法は、上記の考え方を発展させ、サブルーチンコールの命令系とサブルーチンの中身の命令群とを2つのシーケンサとして分離させたものである。前者はマスタシーケンサであり、プログラム全体の流れを制御する。後者はスレーブシーケンサから処理の起動、先頭番地、パラメータを受けて、ルーチン処理だけを実行するスレーブシーケ

ンサである。当然両シーケンサ間に、適当な同期機能を持たせている。この利点は、両シーケンサの命令長をそれぞれ最適に設定できることである。通常マスタシーケンサの命令長を飛躍的に短くできるので、プログラムメモリの節減効果は大きい。

この2階層プログラミング手法を発展させ、メインおよびスレーブの両シーケンサを、マイクロ命令で制御するのが、2階層マイクロプログラムによる方法であり、マイクロプロセッサでは、1979年に68000で始めて導入された。68000では、マスタシーケンサをマイクロ制御、スレーブシーケンサをナノ制御と呼んでいる。基本動作は、マイクロ制御のマイクロ命令が、ナノ制御命令メモリ内のナノ制御命令の番地を指定する構成である。当然、ナノ制御命令のセットで、プロセッサのすべての動作を実現できるようになっており、68000の場合、約280語で構成されている。そして当然、ナノ制御メモリは、ROMで構成する。

ここでは、上記2つの代表的な削減技術を、並列処理プロセッサへ拡張した場合について検討した。以下具体的に、6.6節で詳述するビデオDSPを取り上げて説明する。このDSPは、3段パイプライン演算器を4個搭載した並列プロセッサである。このマイクロプログラム制御方法を図6.5.1に示す。

まず削減技術(#1)により、SET-UP命令とEXEC命令に分離した。本ビデオDSPでは、5個のデータメモリと4個の演算プロセッサを10組のデータバス(VBUS0~9)を使ってほぼ自由に接続して、種々の並列演算を行う構成になっているので、演算モード毎に最適な接続に切り替える。この切り替え制御に多くのマイクロフィールドを必要

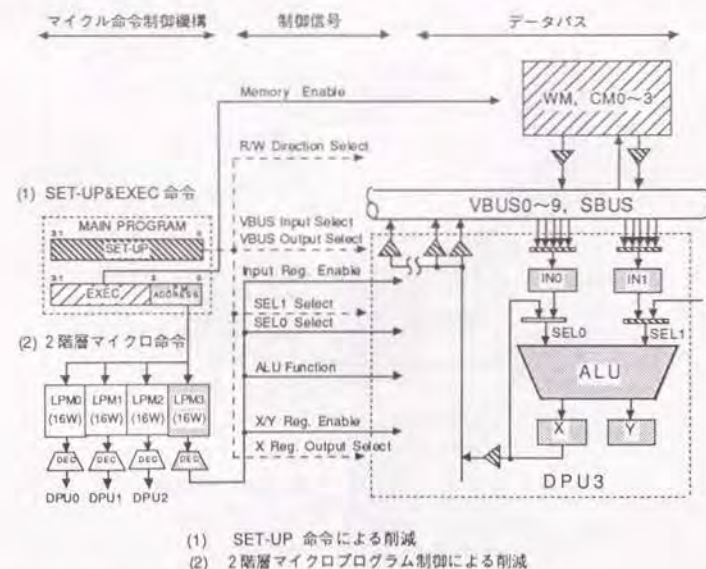


図6.5.1 2つのマイクロフィールド削減技術

とする。このため通常のシングルプロセッサのLSIに比べて、(#1)の削減技術は、いっそう効果的である。具体的には、SET-UP命令で、DPUへのデータ入力バス、データバスへのデータ出力バス、ALUへの入力バス等を指定する。これはベクタ演算の性質によって決まるものであり、そのベクタ演算が終了するまで変更されない。これにより、EXEC命令で制御しなければならない対象は激減する。

次に、(#2)の2階層のマイクロプログラム制御技術を、並列処理用DSPへ発展させて適用した。改良点は以下の3点である。

- ・DPU毎に、独立したスレーブのマイクロシーケンサを持たせ、そのマイクロ制御メモリ(LPM: Local Program Memory)にそれぞれ異なるプログラムを搭載できるようにした。これにより、DPU毎に異なる処理を、並列に実行させることができる。
- ・EXEC命令をベクタ演算のみに限定して適用した。これは、ビデオのベクタ演算のステータスステップが非常に短いため、各LPMに置かれるプログラムステップ数も非常に短いものとなり、プログラムメモリ容量の削減効果が大きいからである。
- ・LPMをRAMとし、プログラムのダウンロードができるようにした。これは、DPU内の演算の組み合わせが非常に多い(複数の入力セクタ、複数の入力レジスタ、ALU、乗算器、複数のアキュムレータ、複数の出力ドライバ等)ため、全モードをROM化して搭載するのが、ほとんど不可能だからである。汎用レジスタ間処理に限定されているマイクロプロセッサとの大きな違いである。

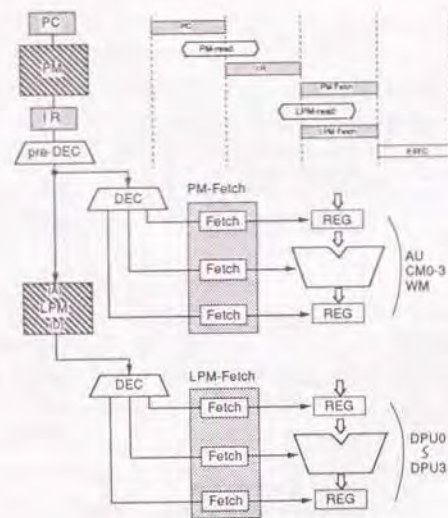


図6.5.2 PMとLPMのリードタイミング

LPMの容量に関しては、ビデオ符号化に現われる種々のベクタ演算を分析し、ほぼすべてのベクタ演算のステータスステップが16W以内に収まることを確認して、16Wと定めた。LPMのプログラムローディングのオーバーヘッドをなくすため、LPMをデュアルバンク構成とし、現行ベクタ演算の実行と並行して、次のベクトル演算のプログラムローディングをできる構成とした。すなわち本ビデオDSPは、32b×512Wのプログラムメモリ(PM)と、デュアルバンクの32b×16WのLPMメモリからなる2階層マイクロプログラム構成とした。

2階層マイクロプログラムのデコード方式を図6.5.2に示す。EXEC命令の中の4bのLPMアドレスフィールドが、LPMのアドレスを指定し、32bの命令を読み出す。この命令は、各DPU内の機能モジュールである、ALU、ADD/SUB、データセクタ、レジスタ等を制御する。EXEC命令の残りのフィールドは、内部および外部のアドレス生成器(AU、AU EX)の更新、データメモリ(CM0~3、WM)のリード・ライト制御、3個の外部パラレルポート(PIO0-2)の転送制御等を行い、チップへのデータ入力から出力に渡ってのパイプラインベクタ演算を、円滑に実行させている。またインストラクションレジスタ(IR)から直接出力される命令と、LPMから読み出した命令のデコードタイミングを、同一のフェッチ信号で合わせることで、2階層を

表6.5.1 プログラムメモリ容量の削減効果

削減技術	構成種別		
	Type-A	Type-B	IDSP
VDB & DPU セットアップ 命令を使用	NO	YES	YES
2階層マイクロ 命令を使用	NO	NO	YES



マイクロ フィールド長	300b (#1)	156b (#2)	32b (PM) 32b (LPM)
プログラム メモリ容量	150Kb (300b×512W)	80Kb (156b×512W)	20.5Kb (32b×512W) + (32b×16W×4x2)

- (#1) * VDB set-up : 10x(3+9)=120b
 * DPU set-up : 24b
 * Main field : 32b-4b = 28b (4b : LPM address)
 * Local field : 32b×4 = 128b (x4 : DPU0-3)
- (#2) * Main field : 32b-4b = 28b (4b : LPM address)
 * Local field : 32b×4 = 128b (x4 : DPU0-3)

全く意識しないでプログラミングできる。

以上述べた2つの技術による、プログラムメモリ容量の削減効果を表6.5.1に示す。本ビデオDSPを、単純なマイクロプログラム制御方式(Type-A)で制御すると、マイクロフィールド長は300bになり、メモリワード数を512Wとすると、プログラムメモリ容量は150Kbとなる。これは、0.8 μ mのLSI技術でもオンチップ化は困難である。

削減技術の内、セットアップ命令と実行命令とを分離させる手法のみを採用(Type-B)した場合には、フィールド長は156bとなるので、メモリ容量は80kbになる。しかしそれでも、オンチップは経済的に望ましくない。そこでさらに2階層マイクロ制御技術を採用すると(IDSP)、マイクロフィールド長は、PMとLPMが共に32bとなり、PM1個のメインフィールドとLPM4個のローカルフィールドを加えて、合計160bのマイクロフィールドとなる。しかしローカルメモリのワード数を大幅に削減できるため、全メモリ容量を20.5Kbに削減できた。この場合、オンチップ化は容易である。

6.5.2 データメモリ容量削減技術

(1) エクスパンダ/コンパクト

ここでは、原画像および処理画像のデータの語長と、処理途中のワーキングデータの語長差を利用して、データメモリ容量を、見かけ上増やして使用する手法について述べる。例えば、本DSPのデータメモリは5バンクから構成され、それぞれのバンクメモリ容量は、

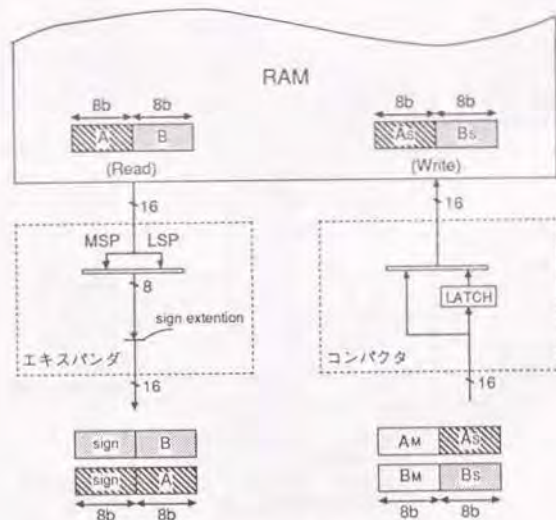


図6.5.3 エクスパンダとコンパクトの機能

512Wである。このため、16 \times 16画素からなるブロック処理では、メモリ容量が不足する場合がある。それは、ポート0を使って現ブロックの処理を行い、ポート1を使って前ブロックと次ブロックのデータを、外部メモリと転送するときである。このインターリーブ処理では、現ブロックの256W以外に、転送バッファを256W以上必要とするからである。この対策として、前ブロックまたは次ブロックの領域を128W程度に削減したい。

具体的には、画像の有効データ語長は、8ビット精度で十分なことに着目し、書き込み時に、8ビットの画像データ2Wをメモリ1Wに蓄積し、読みだし時に、元の2Wに戻せる機能を搭載し、実質上のメモリ容量を1KWとした。この時のメモリ構成を図6.5.3に示す。書き込み時には、2個の16ビットデータA、BのそれぞれのLSP (Least Significant Part) がコンパクションされ、2Wからなる16ビットデータとして書き込まれる。このとき、8ビット幅のラッチを使用して、2マシンスサイクルに1回ずつ書き込む。逆に読みだし時には、同一データを2回ずつ読みだし、最初のサイクルでAデータ、次のサイクルでBデータを出力する。出力される16ビットデータの有効領域は、下位8bであり、MSP (Most Significant Part) 8ビットに符号ビットが付与される。上位へ符号ビットを拡張するのは、演算時のオーバーフローを抑止するためである。またこのときのアドレス制御として、同じアドレスを自動的に2回ずつ繰り返して生成するリピータ機能を、アドレス生成器に持たせ、プログラマが全く意識しないで済むようにした。

(2) デュアルポートRAM構成技術[15][16]

デュアルポートRAMは、各ポート毎に専用の読出しバスとデータバスを有しているの、異なる2つのデータの読出しと書き込み、または2つのデータの同時読出しが可能である。このため信号処理のように2項ベクトル演算の頻度が高い処理には、極めて適したデータメモリである。しかしこのメモリは、CMOSでメモリセルを構成したとき、占有面積が大きくなる欠点を持っている。その原因は、メモリセルを構成するのに、PMOS (P型MOSトランジスタ) を2個、NMOS (N型MOSトランジスタ) を6個必要とするからである。具体的には、フリップフロップ用に、PMOSとNMOSがそれぞれ2個ずつ、そしてゲート用にNMOSを4個 (ゲートがデュアルのため) 必要とする。このため、メモリセルの回路は図6.5.4となり、レイアウトすると図6.5.5のようになる。ここに、T1、T3がPMOS、その他がNMOSである。レイアウト図から明らかなように、NMOSとPMOSは分離を必要とするため、このようにPMOSとNMOSの素子数比が2:6の場合、極めて面積効率が悪くなってしまふ。そのため、同一容量の1ポートメモリを2面設ける場合との占有面積差が僅かになり、デュアルポートメモリの利点が半減してしまっている。

そこで、4個のゲートトランジスタ (T5~T8) に着目し、1つのゲートをNMOS、もう1つのゲートをPMOSで構成し、メモリセルのPMOSとNMOSの素子数比が、4:4になる場合を検討した。このときの問題は、PMOSをゲートに使用することによって、NMOSに比してゲートのON抵抗が高くなり、読みだし速度が劣化する点である。即ち、デュアルポート読み出しのとき、PMOSゲートとNMOSゲートの読み出し速度が異なり、遅いほうのPMOSゲートの速度で、メモリサイクルが決定される。これに対しては、ビット線のプリチャージ電位を電源電圧VDDまで高め、PMOSのON抵抗を下げ、

メモリセルからのビット線のドライブ能力を、PMOSゲートとNMOSゲートとでバランスさせることで解決した。

この新しいメモリセルの回路は、図6.5.4において、トランジスタ(T5, T7)をPMOSに取り替えたものである(図面省略)。これにより設計したメモリセルのレイアウトは、図6.5.6のようになる。すべてのパターンが完全に対称化でき、無駄面積がなくなっているのがわかる。このメモリセルの面積は、ほぼ1層および2層のアルミ線幅で定まっており、 $1826\mu\text{m}^2$ ($2.5\mu\text{m}$ ルールするとき)である。一方、同一プロセスでの1ポートメモリのメモリセル面積は、 $1440\mu\text{m}^2$ である。即ち、このデュアルポートRAMのメモリセルは、1ポートメモリと比較して、ワード線とビット線の数2倍、トランジスタ数が1.33倍になっているにもかかわらず、面積は1.27倍に留まっている。これより、本技術によって、高密度化が実現できることが実証できた。

その他の改良として、書込み回路とセンスアンプを共用化し、ビット線の浮遊容量を低減してメモリアクセスの高速化を図り[15]、1ポートメモリとほぼ同等のスピードとした。

6.6 オンチップメモリ容量を削減したビデオDSPの構成法

6.6.1 ビデオDSPの背景

動画をリアルタイムで符号化するDSPをビデオDSPと呼んでいる。このDSPは、ITU-Tにおける、テレビ会議、テレビ電話を対象とした通信用符号化方式(H.261勧告)と、ISOにおける、CDやDAT等のパッケージメディアを対象とした蓄積用動画符号化方式(MPEG勧告)とが検討されるのと並行に開発され、87年頃から種々発表されてきている[17][18][19][20]。

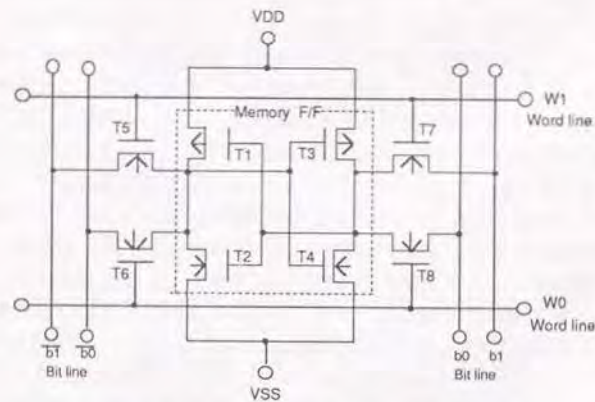


図6.5.4 従来のデュアルポートRAMのメモリセル回路

(注) T5とT7をPMOSに取り替えると、占有面積の小さい新しいデュアルポートメモリセルになる

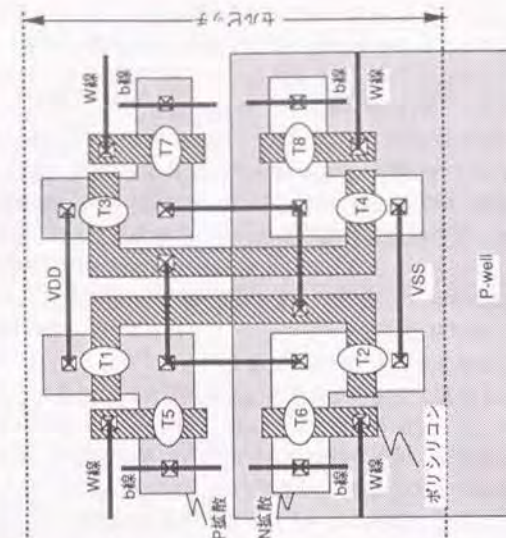


図6.5.6 新しいデュアルポートRAMのメモリセルのレイアウト概念

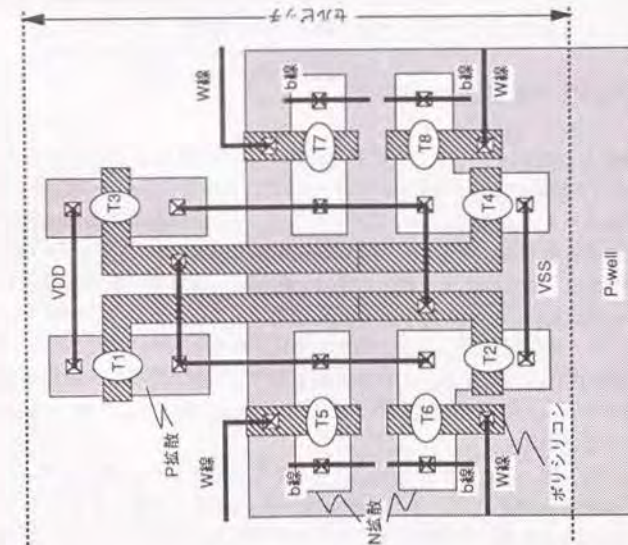


図6.5.5 従来のデュアルポートRAMのメモリセルのレイアウト概念

これに先立ち、80年代の始めから、ノイズ除去やエッジ強調など画像の前処理を高速に処理するプロセッサLSIが開発されていた。これらは、(#1) ISPのような局所近傍型並列プロセッサ[21][22]、(#2) AAP[23]のような低粒度のメッシュ状アレイプロセッサ、(#3) T9506[24]のような画像演算プロセッサの3種に分類できる。いずれも、低レベル画像処理に特化して、LSIの回路を演算器に重点的に配分しているのが特長である。例えばメモリについては、(#1) と (#2) は、処理を近傍データに限定しているため、僅かなレジスタファイルを搭載しているのみである。(#3) は、FFTのような大局処理も実行できるが、メモリはすべて外付けである。またデータ入出力では、(#1) と (#2) は、ビットまたはワード単位のシリアル入出力のみに限定している。即ち、動画の要求処理能力が高いため、汎用的なものを作れるところまで、LSI技術が至っていなかったと言える。

一方、ビデオDSPでは、かなり高度な処理を要求されるのが特長である。例えば、64Kb/sのH.261-CODECを取り上げると、種々の帯域圧縮アルゴリズムが組み合わされるため、一画素当りの処理量が250オペレーションにもなり、かなり複雑である[28]。また、16×16画素を単位とするブロック処理を行うので、DSP内で一括処理するデータ量も多い。このため、性能と汎用性を両立させる構成になっているものが多い。それでも汎用DSPと比べると、データメモリのアドレス生成機能が、2次元データのアクセス用に特化されていたり、演算器が、画像符号化で良く用いられる動き補償やベクトル量子化に強化されている等、適切な専用化が加えられている。

このような流れの中で、従来のビデオDSPの性能を飛躍的に向上させ、1ボードのビデオCODECを実現できるビデオDSPを開発した。要素技術として、4.6節で述べた「多バンクデータメモリのアドレス生成技術」、5.4節で述べた「適応処理用のSIMD型並列アーキテクチャ技術」、そして6.5節で述べた「メモリ容量削減技術」を取り入れた。そして、先端0.8μmBi-CMOS技術と融合させて実現した。

6.6.2 ビデオDSPへの要求条件の分析

H.261標準に準拠したビデオCODECの符号化部の構成を図6.6.2に示す。主に、動き予測補償部(MC: Motion Compensation)、ループフィルタ部(LPF: Loop Filter)、順方向および逆方向コサイン変換部(DCT/IDCT: Discrete Cosine Transform/Inverse DCT)、そして可変長符号化部(VLC: Variable Length Coding)からなっている。

動き予測補償部では、動きベクトル検出の手法を使って、現フレームの対象画素ブロックが、前フレームの探索範囲内のどこから来たかを検出し、フレーム間差分が最も小さくなる差分信号を生成する。差分信号は、DCT部で符号化されると共に、前フレームの再生画像に加えられて、再生画像を更新する。これらの処理の中で、動きベクトル検出が最も演算量が多い。この動きベクトル検出処理では、一乗(L1-norm)または、二乗(L2-norm)の距離計算が重要な演算である。

ループフィルタ部では、各画素に対して、3×3サイズのマスクを使ってローパスフィルタをかけ、ノイズ除去処理を行う。この処理では、積和演算が主体である。

順方向および逆方向コサイン変換部では、2次元DCTとIDCTを行う。この処理では、変換マトリクスB(m,i)を使って、行列積演算を行と列に対して2回実行する。具体的

には、画素ブロックX(i,j)を入力として(6.6.1)式の演算を行い、画素ブロックZ(m,n)を出力する。画素ブロックのサイズは、8×8である。

$$\begin{aligned} Y(m,j) &= \sum_i B(m,i) \cdot X(i,j) \\ Z(m,n) &= \sum_j Y(m,j) \cdot B(n,j)^T \end{aligned} \quad (6.6.1)$$

ここにB(n,j)Tは、行列B(n,j)の転置行列である。この処理でも積和演算が主体である。また演算精度は、H.261勧告で示されている復号時の許容誤差から、24ビット必要なことがわかっている。これは、H.261-CODEC処理で、最も精度を要する演算である。

可変長符号化部では、ビット処理演算、テーブル検索、論理演算等、ベクトル化しにくいスカラー演算が、主たる処理である。

以上により、H.261に準拠したビデオCODECに要求される処理能力を見積ると、(6.6.2)式で表わされる。

$$PE \cdot FR \cdot (\sum_{NA} + \sum_{TA}) + Ts \cdot (\sum_{NB} + \sum_{TB}) \quad (6.6.2)$$

ここに、

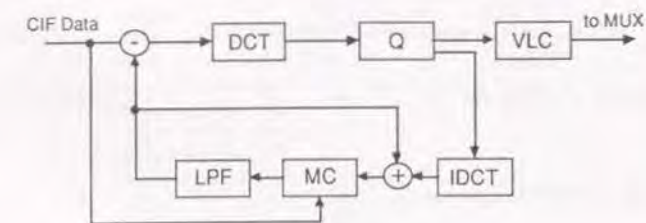
PE : pixels in a frame (352 X 288)

FR : frame rate (10 ~ 30)

Ts : coded data transfer speed (64 Kb/s ~ 384 Kb/s)

\sum_{NA} : total number of numerical calculations per pixel for CIF side
(~135 : coder, ~80 : decoder)

\sum_{NB} : total number of numerical calculations per pixel for MUX side
(~12 : coder, ~10 : decoder)



- (1) MC : Motion Compensation
- (2) LPF : Loop Filter
- (3) DCT : Discrete Cosine Transform
- (4) IDCT : Inverse DCT
- (5) Q : Quantizer
- (6) VLC : Variable Length Coding

図6.6.2 H.261 準拠画像CODECの符号化部

ΣA : total number of data transfers per pixel for CIF side
(~ 10 : coder & decoder)

ΣB : total number of data transfers per pixel for MUX side
(~ 2 : coder & decoder)

第1項と第3項は、算術と論理の全演算量を示している。第2項と第4項は、ビデオCODECをDSP-LSIで構成したときに、LSIと外部データメモリ間で必要なデータ転送量である。これより、演算処理能力とデータ転送能力の両方の改善が必要になることがわかる。また、H261-CODECの符号化および復号化部全体で、約300～1000MOPS (Mega Operation per Second) の処理能力を要することがわかる。

以上に基づき、ビデオDSPの仕様決定に当たって、以下の2点を考慮した。第1点は、ビデオ符号化アルゴリズムは適応処理が非常に多いことである。適応処理には、条件判断が伴うためパイプラインが乱される。このため、DSPの平均処理能力は、ベクトル化された最高処理能力に比べて、かなり低下することを考慮しておかなければならない。第2点は、H261-CODECを1ボードに納めるために、使用するDSP-LSIを4チップ程度に抑えなければならないことである。

そこで、DSPの性能目標を、最大300MOPS、平均150～200MOPSとした。この性能は、従来DSPの約10倍である。また効率よいマルチDSP動作を行う機能が必要である。このため、3組のパラレルポートを搭載し、それぞれに専用のDMAプロセッサを内蔵させてデータ転送能力を高めることとした。この3組のパラレルポートは、ビデオデータの入出力、符号化されたデータを送受信するデジタルインタフェース、マルチプロセッサ動作での共通バス等に、柔軟に割り当てることができるものとする。

6.6.3 基本アーキテクチャ[25][26]

(1) 基本設計思想

一般的に、ハーバードアーキテクチャのDSPの性能は、(6.6.3)式で表される。

$$\text{MOPS} = \text{EC} \cdot \text{NS} \cdot \text{NP} \quad (6.6.3)$$

ここに、

EC = execution cycle / sec

NS = number of pipeline stages in each execution unit

NP = number of execution units

ECは、一秒間当たりの演算の実行サイクル数であり、パイプライン演算器やデュアルポートデータRAM等データバス上のクリティカルパスで決定される。ECは、これらのバスが高速化されるに伴い向上する。一般にDSPは、RISCタイプのプロセッサであり、

すべての命令を1マシンサイクルで実行するのを基本としている。このため、演算の実行サイクル数は、マシンサイクル数と同じになる。即ち、DSPにおけるECは、(6.6.4)式のように、クロック周波数で簡単に表すことができる。

$$\text{EC} = \text{fc} / n \quad (6.6.4)$$

ここに、

fc = clock frequency

n = machine cycle time / clock cycle time

式(6.6.3)で示した、DSPの性能を表す3つのパラメータより、性能向上の手法を3通り考えることができる。第1は、超高速マシンサイクルDSP (以下高サイクルDSP)、第2は、超長パイプラインDSP、第3は、並列演算器DSP (以下並列DSP)である。これら3手法の中で、超長パイプラインDSPが、ビデオ符号化アルゴリズムに向かないことは明らかである。なぜなら、符号化のアルゴリズムは、 8×8 または 16×16 画素からなる画素ブロックを処理単位としているので、パイプライン長に比例して、ベクタ演算のオーバーヘッドが大きくなるからである。例えば、演算器のパイプラインが12ステージのDSPを考えると、 8×8 画素ブロックの場合のオーバーヘッドが、約50%程度になると見積られる。このため、超長パイプラインの手法は除外することとした。

そこで、残りの2つの手法について、それぞれの利点を定量的に比較した。その結果を表6.6.1に示す。高サイクルDSPでは、演算器の並列度を1でクロックを100MHzとし、並列DSPでは、並列度を4とした替わりに、クロックを4分の1の25MHzと

表6.6.1 300-MOPS DSPを実現する2つの手法の比較
(高サイクルDSPと並列DSP)

比較項目	高サイクルDSP	並列DSP
Execution cycle / sec (Ec)	100MHz	25MHz
Number of stages (Ns)	3	3
Number of parallel (Np)	1	4



消費電力 $P \sim \text{Ec} \times V_{\text{cc}}^2 \times G$	1	$\frac{1}{4} \times \left(\frac{3}{5}\right)^2 \times \frac{2.5}{1.2} \sim 0.2$
Ec	1	1/4
Power supply (Vcc)	5V	3V
Gate counts (G)	1.2	2.5 (#1)
占有面積	1.2	2.5

(#1) (4.6)節で述べたアドレス生成技術による

た。即ち、両者の最高処理能力は同じである。この両者を比較したときの差は、消費電力で顕著に現われており、並列DSPの方が高サイクルDSPよりも、消費電力の点で圧倒的に優れている。この理由は以下の2点である。

- (#1) $0.8\mu\text{m}$ -CMOSプロセスを想定した場合、 25MHz 動作は 3V 電源で実現できるが、 100MHz 動作は、 5V 電源にしなければ達成できない。すなわち、並列DSPは、電源電圧を下げることができる。
- (#2) 並列DSPは4個のパイプライン演算器を搭載するが、全回路規模で見ると4倍にはならない。その理由は、データRAMのアドレス生成ユニット(AAU)に工夫を凝らせることにより、付随するデータRAM容量を、ほとんど同程度に抑ええることができるからである。この技術については、4.6節で詳述した。

このため、並列DSPは、並列度が4にもかかわらず、回路規模を従来DSPの2.5倍程度に抑えることができ、電源電圧低下との相乗効果により、いっそう低消費電力化できるのである。一方高サイクルDSPは、ドライバ系の高速化や遅延調整等、高速動作のための付加回路が必要であり、従来DSPに比して約20%程度の回路規模増を見込まなければならない。消費電力的によりいっそう不利となっている。

結論として、4個のパイプライン演算器を搭載した並列DSPは、高サイクルDSPに比して、消費電力は僅か20%程度である。反面回路規模が大きいため、アクティブ領域(LSIチップは、アクティブ領域とパッド等の周辺領域からなる)の周辺領域占有面積は2倍程度になるが、ビデオ処理の様なハイエンドLSIでは、消費電力の方がより差し迫った問題なので、並列DSPの方を採用することとした。また占有面積は、マクロセルを使った高密度レイアウトにより改善することができる。事実マクロ設計により、従来のスタンダードセル設計に比して30%の改善を行った[27]。

2) 全体アーキテクチャ

提案するDSP(以下IDSP: Image Digital Signal Processor)の全体図を図6.6.3に示す。IDSPは、3段パイプラインの演算器を4個搭載して(以下データプロセッシングユニット: DPU0~3)、並列パイプライン演算をする。このため、理想状態では、1マシンサイクルの間に12演算を同時に実行できる。設計の要点は、 $0.8\mu\text{m}$ -CMOS技術の集積規模制限の中で、この演算器と演算器を効率良く実行させる仕組みを、如何に搭載するかである。即ち、演算器以外の占有面積を如何に削減するかである。

その方策として、以下の4技術を考案した。第1は適応SIMD技術である。これは、4個のDPUの内、内部の状態フラグが同じものを自動的に選択して並列実行させる機能である。それぞれのDPUに、条件が異なるときに自己ストップし、条件が合っているときに自己起動する機能を持たせることで実現している。この利点は、プログラミング上完全なSIMDであり、プログラマが個々のDPUの状態を全く意識しなくても並列処理ができることである。ビデオ符号化では、動き補償フレーム間予測に効果的である。この詳細は5.4

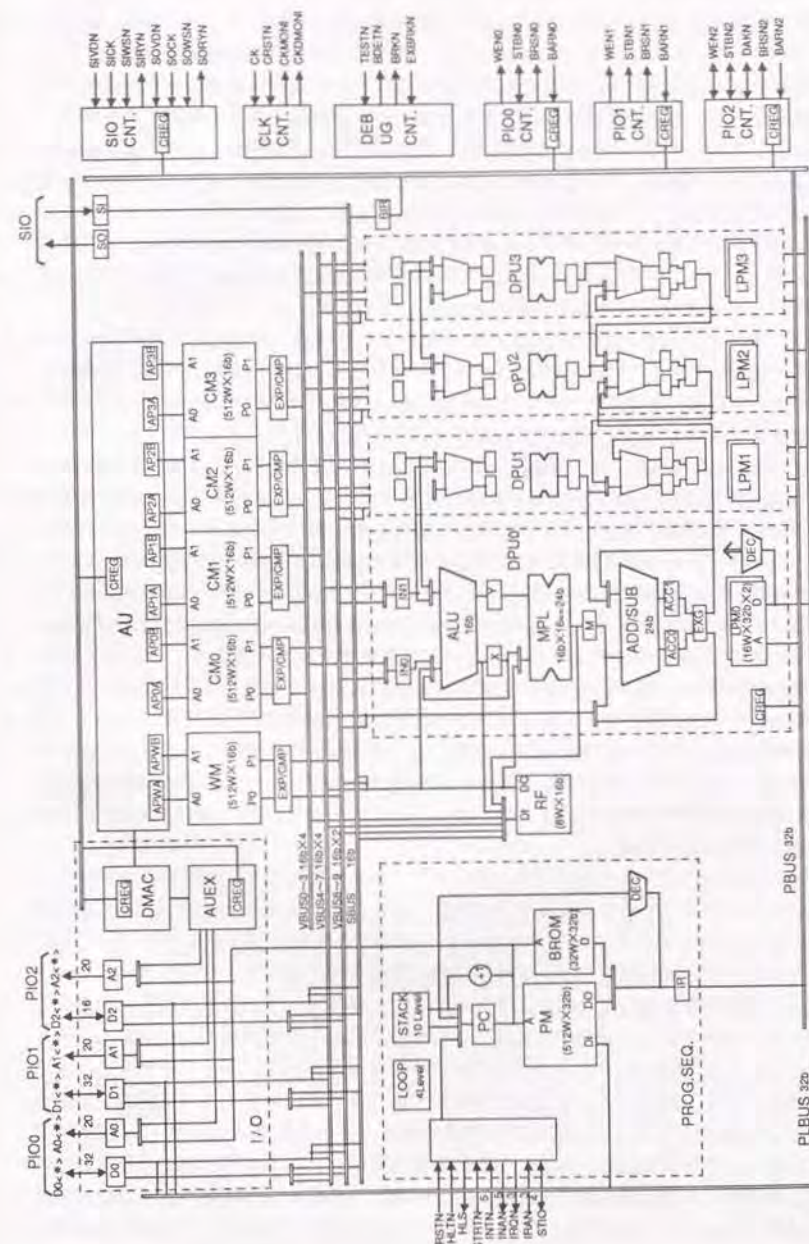


図 6.6.3 ビデオDSPの全体図

節で述べた。

第2は、4個のDPUに5個のデュアルポートRAMを割り振ったことである。5個の内4個をデータキャッシュメモリ(CM0~3)、残りの1個をワークメモリ(WM)と呼ぶ。各CM i ($i=0\sim3$)は、それぞれDPU i ($i=0\sim3$)に対応し、WMは4個のDPUに共通である。WMを共通に使用することにより、2次元フィルタ処理のような3アドレス演算を4並列で実行できる。また4.6節で述べたアドレス機能により、動き補償処理の最重要演算である動きベクトル検出のように、4個のCMに股がって並列に4データを読み出す複雑な処理も、並列に実行することができる。

第3は、DPUと各デュアルポートRAM間のデータ転送を並列に行うため、10組の16ビットデータバス(VBUS0~9)と、もう1組のスカラ演算用の16ビットデータバス(SBUS)を設け、バスネックを回避したことである。

第4は、チップ外部とのデータ転送のボトルネックを回避するため、3組のパラレルポート(PIO0~2)と1チャンネルのシリアルポート(SIO)を設けたことである。各パラレルポートには、1MWのアドレス空間を設け、チップ外部のビデオフレームメモリとの直接接続を可能とした。3組のパラレルポートのうち、PIO0とPIO1は、データ幅を32ビットとし、メモリバンド幅を広げて、ビデオデータのリアルタイム転送ができるようにした。一方PIO2は、主にマイクロプロセッサのシステムバスとして使用するため、チップのピン数制限を考慮して、16ビット幅とした。各パラレルポートにはそれぞれDMAプロセッサ(図中では3個のDMAをまとめてDMACと表現)を内蔵させ、命令による処理と並列に、データ転送を行えるようにした。また外部メモリへのアドレス生成用に、外部メモリアドレスユニット(AUEX: Address Unit for EXternal memory)を搭載した。このユニットから、3組のパラレルポート全てのアドレスを生成する。

以下その他の特長について述べる。まず8Wのレジスタファイル(RF)には、以下の2つの役割を持たせた。第1は、マイクロプロセッサと同じように、1W単位でのレジスタ間処理を効率よく実行するためである。DPU0の中のALUおよびMPLとレジスタファイルとを専用のバスで結合させることにより、パイプライン遅延なしにALU演算およびMPL演算を実行することができる。第2は、デュアルポートメモリ間のパイプラインデータ転送を中継させるためである。

以上述べたすべてのユニットを、マイクロプログラムで制御する。プログラムシーケンスは、512Wのメインプログラムメモリ(PM)、32WのブートストラップROM(BROM)、10レベルのスタックコントローラ、4レベルのループコントローラ(LOOP)からなっている。その他、割り込み制御回路、外部バスのバスアービタ、プログラムデバッグサポート機能等汎用DSPの基本機能のすべてを搭載し、プログラム開発を容易にしている。メインプログラムメモリの容量は、LSI技術制約からの搭載可能規模と、ビデオ符号化を行うときのプログラムダウンロードにより生じるオーバーヘッドとを考慮して、512Wに定めた。即ち、例えば128Kb/sのビデオCODECを4チップで構成したときの、各チップに割り当てられる処理のプログラムステップ数を見積ると、2~3Kステップとなるためオンチップ化困難である。しかし符号化処理は、動き補償予測符号化やコサイン変換等、機能単位の処理に分解でき、そしてそれぞれの処理は、画素ブロック単位の処理をフレームサイズに渡って繰り返すので、スタティックステップに比して圧倒的にダイナ

ミックステップが多い。このため、各機能処理を行うに十分なプログラム容量さえあれば、ダウンロードのオーバーヘッドは無視できる。そこで、符号化のアルゴリズムをすべて解析し、いずれの機能処理も512Wでプログラミングできることを確認した[28]。

また上記プログラムのダウンロードは、チップ内の32ビット幅のプログラムロードバス(PLBUS)を介して行われる。ダウンロードする外部プログラムメモリの最大容量は64KWである。またPLBUSは、各DPU内の32Wのローカルプログラムメモリ(LPM0~3)のダウンロード用バスと共通である。32ビット幅のプログラムバス(PBUS)は、32ビット命令をLSIの各ブロックにあるローカルデコーダに分配するために設けてある。この分散デコード方式により、プログラムシーケンスと各ブロックとの結線数を大幅に減らすことができ、高集積化と高速化を図ることができた。

6.6.4 データバス構造

(1) バスアーキテクチャ

5個の各デュアルポートRAMのそれぞれのポート毎に、データバスを設けた。各バスは独立してデータ転送を行うことができるため、同時に10個のデータ転送が可能である。この構造を図6.6.4に示す。CM0~3のポート0(P0)は、それぞれVBUS0~3に接続され、ポート1(P1)は、VBUS4~7に接続されている。

一方、各DPUの2つの入力ポート(I0とI1)は、それぞれVBUS0~3とVBUS4~7に接続されている。それぞれのI0ポートは、VBUS0~3の中から適切なバスを選択できるし、I1ポートは、VBUS4~7の中から選択することができる。また各DPUの出力ポート(O)は、VBUS4~7を介して、CM0~3の中から適切なRAMを選択して、そのポート1(P1)にデータを転送することができる。即ち、各デュアルポートRAMの各ポートは、すべてのDPUと自由にデータ転送を行うことができる。

3組の外部パラレルポート(PIO0~2)の入力(I)と出力(O)は、すべてVBUS4~7に接続されている。そして各パラレルポートは、それぞれ独立にVBUS4~7

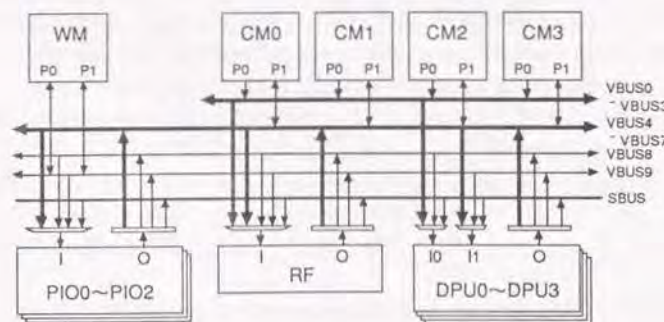


図6.6.4 ベクタバスとスカラバスの構造

を介して、並列にデータを転送することができる。また一般に外部の転送クロックは内部バスの転送クロックよりも遅いので、速度変換回路を設けた。具体的には、32ビット幅のポートに16ビット幅の2個のデータを乗せる。チップ内部に取り込むときは、内部の2クロックを使って2データに分けて取り込み、逆に出力の時には2データを圧縮する。この2:1の圧縮・伸張により、外部クロックの周波数を内部の1/2に下げることができる。

VBUS8とVBUS9はWMのポート0とポート1に接続されている。そしてVBUS8とVBUS9は、DPU0~3およびPIO0~2に接続されているので、WMはCM0~3と同様に、すべてのDPUおよびPIOとデータ転送ができる。さらに、デュアルポートデータRAMと外部パラレルポート間のデータ転送の中継として、8Wの16ビットレジスタファイル(RF)を設けた。RFは10個すべてのデータバスと接続されている。

ベクタ演算に加えて、1ワード単位で処理するスカラー演算についても効率よい処理を行うため、16ビットの共通バス(SBUS)を設けた。VBUSはベクタ演算の高速化を目的とするため、ベクタ演算を行うユニットにのみ接続したが、SBUSは、スカラー演算、開発段階でのチップの機能テスト、プログラムデバックのデータ入出力にも使うため[29]、DSPチップのほとんどすべてのレジスタに接続した。

(2) マルチDSP機能

ここでは、ビデオCODECに適したマルチプロセッサ構成法について述べる。一般にマルチプロセッサシステムは、2つの視点から分けて4種に分類することができる。第1は、メモリ資源の共有の有無により密結合と粗結合に分けられる。第2は、タスク配分の切り口により、機能分散と負荷分散に分けられる。

一般に、システムの要求性能がプロセッサの処理能力を大きく上回り、大規模な並列処理システムでなければ処理できない場合には、疎結合型の負荷分散システムが適している。例えば、既存DSPを使って、機能分割でビデオCODECを構成しようとする場合、DSPの処理能力が低いため、符号化のそれぞれの機能モジュール(動き補償予測符号化やコサイン変換等)さえ、1チップで処理することができない。そこで機能モジュールをさらに機能分割することが考えられるが、分割のオーバーヘッドが大きくなり、実用的でない。そこでこの場合には、画面を分割して、各々の分割画面にDSPを割り当てる負荷分散方式とせざるを得ないのである。次に、負荷分散の中で密結合にすると、空間分割した境界のそれぞれのオーバーラップ領域は、境界の両方のプロセッサからデータを必要とするため頻りにアクセスされる。このため共有メモリでデータ転送ボトルネックを生じやすくなる。以上により、性能の点からは、各DSPに十分なローカルメモリを持たせた疎結合型の負荷分散方式が、ビデオCODECに適していると言える。

しかしこの方式では、ハードウェア規模が大きくなるため、1ボードのビデオCODECを実現するのは難しい。そこで、機能分散処理ができるように、DSP性能を300MOPSに向上させた。このようにすると、以下の理由により機能分散の方が有利となる。

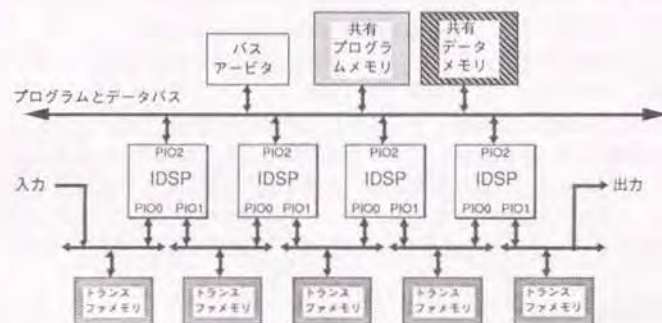
- (#1) 負荷の平均化: 負荷分散システムは、画素ブロックのデータに依存して処理量が大幅に変わるため、プロセッサ間で負荷バランスを取るのが難しい。一方機能分散は、画

素ブロックの処理量バラツキが画面全体で平均化されるため、負荷変動が小さい。

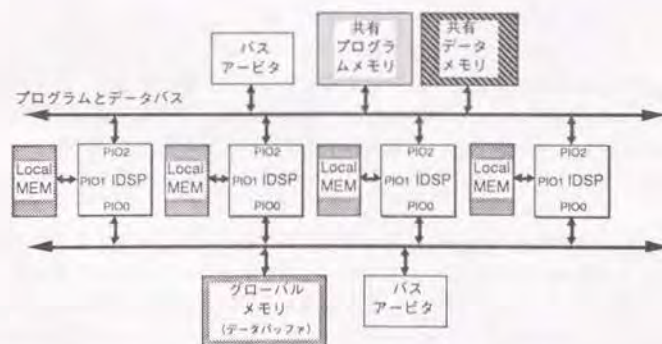
- (#2) プログラム容量が小: 機能分散では、それぞれのDSPは、自分が処理する機能のプログラムのみをロードすればよいので、プログラムメモリ容量を小さくできる。

密結合型機能分散のマルチDSPシステムを図6.6.5(A)に示す。3個の外部パラレルポートの内の1個を、プログラムのダウンロードに用いる。残りの2個のポートは、DSP間データ転送に用いる。このとき、外付けの転送メモリ(トランスファメモリ)を使用する。8×8や16×16サイズの画素ブロックを単位とする転送の場合には、チップ内部のWMとCM0~3で、転送に必要な十分な容量を確保できるので、チップ間ダイレクト転送を行うこともできる。このときには、外付けメモリは不要である。

3組のパラレルポートを備えているため、密結合型負荷分散システムを構成することもできる。その例を図6.6.5(B)に示す。1個のポートを使って、DSP毎にローカルメモ



(A) 密結合型機能分散



(B) 密結合型負荷分散

図6.6.5 IDSPによるマルチプロセッサシステム

メモリを接続し、領域分割で割り当てられた領域のデータをすべて格納する。残りのポートの1個には、プログラムメモリとデータメモリを共有接続で持たせ、各DSPへのダウンロードに使用する。最後の1ポートには共通のワークメモリ（グローバルメモリ）を接続し、DSP間のデータ転送に使用する。このシステムは、HDTV（High-Definition TV）のような大画面を扱う処理に有効である。

6.6.5 データプロセッシングユニット

ここでは、ビデオ符号化アルゴリズムに適合させた演算処理部の構成と高速化技術について述べる。最初の特長は、演算部のパイプラインを3段（1段目は16ビットALU、2段目は16×16ビット乗算器、3段目は24ビットの加減算器）としたことである。既存DSPでは、フィルタ演算が主体であるため、乗算器と加算器とからなる2段パイプラインが一般的であるが、ビデオ符号化では、2乗距離計算の頻度が高いので、3段パイプラインの方が有利である。

演算語長は、ITU-T勧告で定められた演算誤差を基に決定した。具体的には、2次元DCTと逆DCTの有限語長シミュレーションを行い、復元データと原データとの差が推奨値以下となる最小語長を求めた。さらに、ハードウェア量を削減するため、3段目の加減算器（累算に使用）のみを24ビット精度、その他を16ビットとする工夫を行なった。

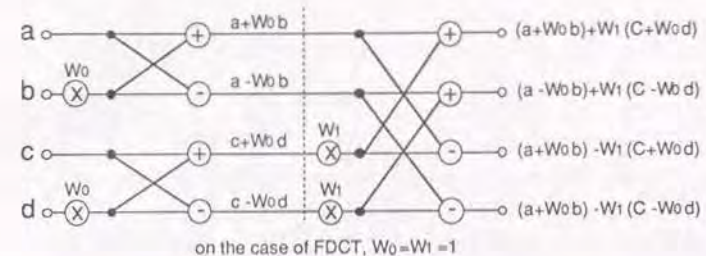
また、論理演算やスカラー演算が、パイプライン遅延なしで処理できるようにした。このため、ALUと乗算器が、それぞれレジスタファイル（RF）と直結されるように、専用バスで接続した。またスカラー演算できるのは、4個のDPUの内の1個（DPU0）のみとした。これは、スカラー演算がベクタ演算に比べて出現頻度が低いので、1データづつの柔軟な逐次処理に合わせた方が有利だからである。以下本演算処理部の特長を述べる。

(1) DPU間結合

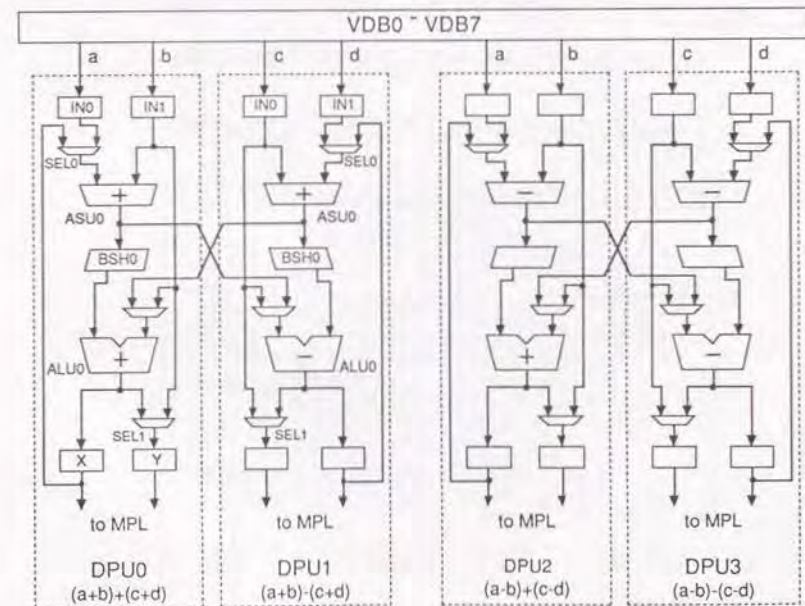
DPUに割り当てるタスクの粒度が大きい場合には、各DPUを独立に並列演算させることで、スループットを向上できる。しかしそのとき、多くの場合、各DPUは、かなりの容量のデータメモリを備えていなければならない。例えば、動きベクトル検出での距離計算の場合には、探索領域を処理の単位とした並列演算となるので、各DPUは、それぞれが探索領域全体のデータを持っていなければならない。これは、ハードウェア規模の大幅な増加につながるため、望ましくない。一方、各DPUが受け持つ演算の粒度を下げると、DPU内に閉じない演算が増えてくる。すなわちDPU間で相互にデータを交換する仕組みが必要となってくる。

そこで、DPUの演算粒度の小さい方法を基本とし、その欠点を解決する方法を検討した。それは、データ交換が円滑に行なわれるように、データフローに適合した転送データバスを設ける方法である。ここでは、ビデオ符号化処理で必須の2種類の演算に対処するために設けた、2種類のDPU間結合バスについて述べる。1つは、バタフライ演算バスであり、もう1つは、4個のDPUをトリー型の接続した累算バスである。それぞれの構成を図6.6.6と図6.6.7に示す。

図6.6.6 (A) は、Leeが提唱した高速DCTアルゴリズム（FDCT：Fast Discrete Cosine Transform）[30]での、実バタフライ演算の2ステージ分を表わしている。この2ステージ分の処理は、(6.6.5)式の処理4個から成っている。この4個の実バタフライ演算を、4個のDPUを使って一括処理するように構成したDPU間結合バスを、図6.6.6 (B) に示す。2個のDPUをクロスで接続させることにより、第1ステージと第2ス



(A) バタフライ演算



(B) DPU間結合

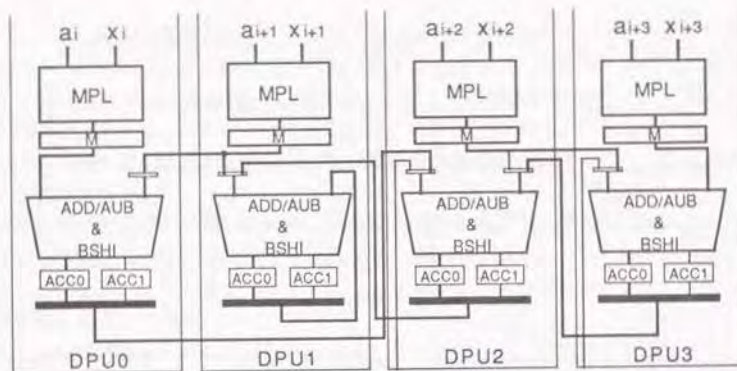
図6.6.6 バタフライ演算のためのDPU間結合

ジのパタフライをパイプラインで連続実行できるようにしている。

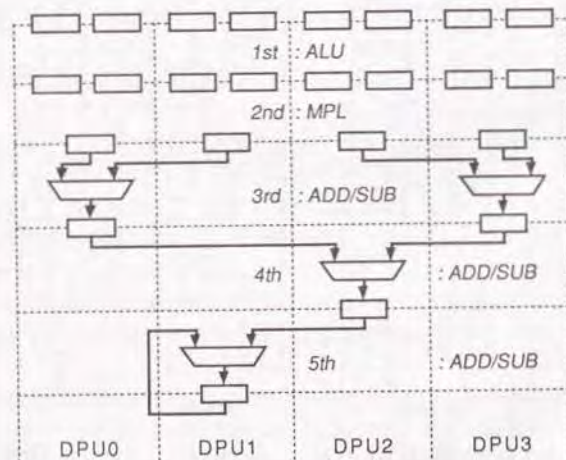
$$x_k = g_k + W_k \cdot h_k \quad (6.6.5)$$

$$y_k = g_k - W_k \cdot h_k$$

$$k=0, 1, 2, \dots, n/2-1, \text{ and } W_k = 1$$



(A) DPU間アキュムレーションのための結合 (太線)



(B) アキュムレーションの等価回路

図 6.6.7 トリー型のDPU間結合 (DPUの累算)

次に、2番目のDPU間結合バスを図6.6.7に示す。これは、4個のDPUの演算結果をまとめて累算するバスであり、DPUの3段目のパイプライン演算器である加減算器をトリートに結合させている。そして、5段パイプラインの乗算器として使用できるようにしている。(A)は実際の接続図であり、(B)はその等価回路図である。この機能は、動きベクトル検出での距離計算や、FIRのフィルタ演算を並列に実行するときの乗加算演算に効果的である。それぞれの演算式を(6.6.6)式と(6.6.7)式に示す。

$$\sum_i \{(x_i - y_i)^2 + (x_{i+1} - y_{i+1})^2 + (x_{i+2} - y_{i+2})^2 + (x_{i+3} - y_{i+3})^2\} \quad (6.6.6)$$

$$\sum_i \{a_i \cdot x_i + a_{i+1} \cdot x_{i+1} + a_{i+2} \cdot x_{i+2} + a_{i+3} \cdot x_{i+3}\} \quad (6.6.7)$$

(2) ALU動作機能

ALUでは、画像符号化の特長に合わせて2点の工夫を行っている。第1点は、演算回路を16b系で構成し、それに加減算の倍精度演算機能を付与したことである。この決定に当たっては、画像符号化処理ではほとんどの処理が16b精度で十分であり、一部DCT演算のみに24b精度が必要という特長を、定量的に分析した。即ち、24b系で構成するという対案と比較して、クロックを高速化できる点と、ハードウェア規模が小さいので演算器の並列度を上げることができる点とから、倍精度演算のオーバーヘッドを帳消しにし、さらに

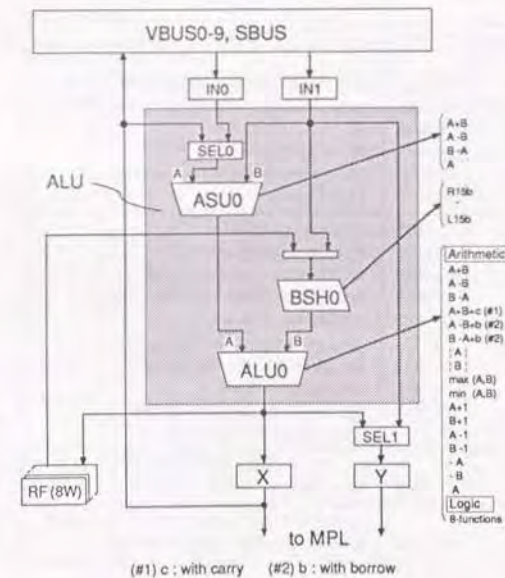


図 6.6.8 ALU動作機能

それを上回る性能となるのである。第2点は、論理演算やスカラ演算等の処理と、距離計算のようなパイプラインベクトル演算処理との、両方をサポートすると構成としたことである。以下、スカラ演算、1段パイプライン演算、3段パイプライン演算について説明する。スカラ演算は、8Wのレジスタファイル(RF)によるレジスタ間演算で行う。この演算ではパイプライン遅延は生じない。また、ループカウンタ更新やデータメモリのアドレス演算でも使用する。1段パイプライン演算では、IN0とIN1を入力レジスタ、Xをアキュムレータまたは出力レジスタとして使用する。この演算モードはDSPで最も良く使われる。本モードを使った累算の例を(6.6.8)式に示す。

$$\sum_i \max(a_i, b_i) : \text{Accumulation of maximum values} \quad (6.6.8.a)$$

$$\sum_i |a_i - b_i| : \text{L1-norm calculation} \quad (6.6.8.b)$$

3段パイプライン演算では、ALUを初段のステージに割り当てている。後段の乗算器ステージと加減算器ステージとにより、2乗距離計算等の3段パイプラインベクトル演算を実行する。以上の特長を持つALUの構成を図6.6.8に示す。

31		26	25	0
op. code		operand		

command	op. code	operation
Execution Scalar	00 0000	Data transfer between RF to CM0-3 and WM
	00 0001	Data transfer between RF to external memories
	00 0010	RF → ALU operation → RF
	00 0011	RF → MPL operation → RF
	00 0100	Data transfer between two registers
Vector	00 1 -	Vector operations
System set-up	01 0000	Initial data set in AU
	01 0001	Address mode set in AU
	01 0010	VBUS0-9 inter-connections set
	01 0011	Data-flow-pass set in DPU0-3
	01 0100	Immediate data set
	01 0101	Set parameter for PM loading
	01 0110	Set parameter for LPM loading
	01 0111	Interrupt priority set
	01 1000	DSP operating mode assignment
	01 0110	Program memory selection
Sequence control	10 xxxx	NC-JMP, B-JMP, LOOP, CALL, HALT, NOP, Command for adaptive SIMD cont.
I/O control	11 xxxx	Data-load start, PM-load start, LPM-load start, Interrupt, Bus-arbitration, Status-out

図6.6.9 メインプログラムの4種のマイクロ命令

6.6.6 マイクロ命令

メインシーケンサのマイクロ命令の一覧を図6.6.9に示す。マイクロ命令は、システムセットアップ命令、実行命令、シーケンスコントロール命令、そしてI/O制御命令の4種類の命令から構成されている。以下、システムセットアップ命令と実行命令の概要と特長を述べる。

システムセットアップ命令は、実行命令に先だってDSPの種々の動作モードやデータバスの指定を行うもので、以下の7種類の命令を設けている。

- (#1) アドレス生成ユニット(AU)の、アドレスモードと初期値を設定する。
- (#2) 各機能ブロック(DPU0~3, CM0~3, WM, PIO0~2)をデータバス(VBUS0~9)を介して接続するときの指定と、DPU内部のバスの指定とを行なう。演算毎に最適なバスを選択するときを使用する。
- (#3) 制御レジスタおよびデータレジスタへのデータロードを起動する。
- (#4) プログラムメモリ(PMおよびLPM)へプログラムをダウンロードするとき、転送用のDMAプロセッサへパラメータをセットする。
- (#5) 各種割り込みの優先順位(プライオリティ)をセットする。
- (#6) IDSPの動作モードセット: マルチプロセッサでのマスタ/スレーブの指定、外部バスのデータ転送スピードの指定、シリアルポートの転送モード(同期/非同期)と転送ワード長の指定を行なう。
- (#7) プログラムシーケンサがアクセスするメモリの指定(BROM/PM/External):

31		29	27	21	17	11	6	4	0
OP		CM0-3		WM	EX-MEM	P-Port	VBUS F/B		LPM
001		P0 P1		P0 P1	PIO0 PIO1 PIO2	PIO0 PIO1 PIO2	F/B		0-3

SEQ : Return Control
P-Port : Parallel Port Control
VBUS F/B : Forward/backward exchange
LPM0-3 : 16w-LPMs addressing

(A) メインフィールド: PM

31		29	27	22	17	14	11	4	0
IN0 IN1 SEL0		ASU -0	BSH0	ALU0	SEL1 X Y	MPL	ADD/SUB & BSH1		ACC0 ACC1

各フィールドの意味は図6.6.3参照

(B) ローカルフィールド: LPM

図6.6.10 ベクタ命令

DSPを起動するときに、起動の手続き（ブートストラップシーケンス）が書かれているBROMから、実際のアプリケーションプログラムが書かれているPMへ、シーケンスを移すときに使用する。またBROMの代わりに、外部メモリから起動するときには、Externalを選択する。

実行命令は、スカラー命令とベクタ命令とからなる。スカラー命令は、レジスタファイルを使ったレジスタ命令であり、ALU演算、乗算、データRAMとの転送、外部データメモリとの転送からなっている。すべて1W単位で処理を行う。ベクタ命令は、4個のDPUによる並列パイプライン処理を基本とする。この命令の、メインフィールド（メインプログラムの命令）と、ローカルフィールド（ローカルプログラムの命令）を、それぞれ図6.6.10の（A）と（B）に示す。図から明らかなように、メインフィールドは、ブランチ制御（SEQ）、データRAM制御（CM0-3, WM）、外部メモリ制御（EX-MEM, P-Port）を同時に行う。またローカルフィールドは、それぞれのDPUの演算制御を独立に行う。

ベクタ命令の特長として、2種類のベクタ演算を交互に行うときに、機能ブロック間の接続替えのオーバーヘッドをなくす機能がある。これは、実行命令のマイクロフィールド長を削減する技術のひとつに位置づけることができる。即ち、ベクタ命令に先だって、予めデータバスの接続をフォワード側とバックワード側の2種類設定（システムセットアップ命令を使用）しておき、それぞれを、フォワード側、バックワード側としておく。そしてベクタ命令実行時に、ベクタ命令のフォワード/バックワード切り替え指定フィールド（VBUS F/B）を使って、ベクタ演算と並列に切り替えるのである。この機能は、例えば適応量子化の様に、画素ブロックの量子化手法が2種類あって、画素ブロックの状態に応じて処理形式を

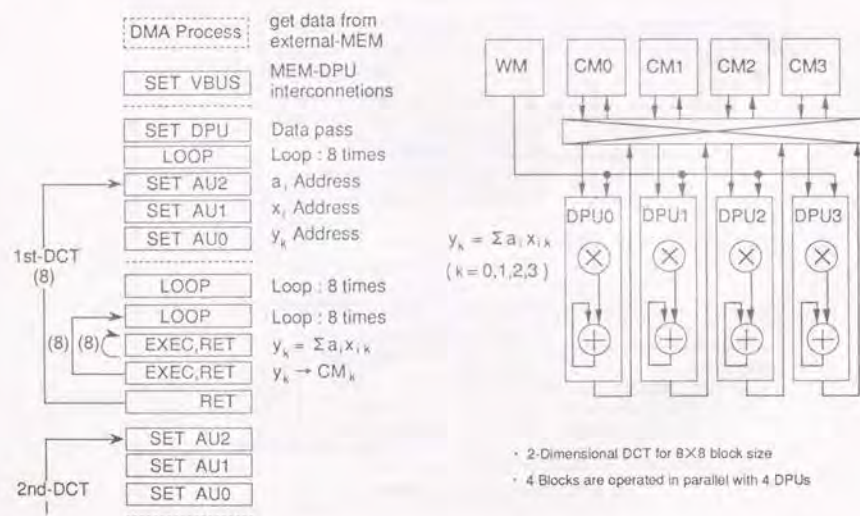


図6.6.11 ベクタ命令による実行例 (DCT)

選択する場合に、切り替え処理でのオーバーヘッドをなくすることができるので、極めて有効である。

上記ベクタ命令の応用例を図6.6.11に示す。これは、2次元DCTを3重ループのベクタ演算で処理している例である。最初のループで積和演算によるポイント周波数の計算、次のループで1次元DCT、最上位のループで2次元DCTを実行している。このときそれぞれのDPUに、8×8サイズの画素ブロックが割り当てられ、それぞれが、SIMD制御で並列に実行される。4ブロック単位の空間分割処理である。

6.6.7 IDSPの性能

(1) VLSIデザイン[31][32]

以上の特長を有するDSPを、0.8μm-CMOS技術とマクロセルによる高密度設計技術とにより1チップに集積した。クロックドライバの負荷バランスを均一化し、かつ段数を一様に3段とすることにより、クロックスキューを1ns以下に抑えたことと、高密度マクロセル技術とにより、演算サイクル25MHz、最大処理能力300MOPSを達成した[27][33]。またチップサイズは15.2×15.2mm²である。チップ写真を図6.6.12に示す。

またDSPのハードウェアの諸元を表6.6.2に示す。マクロセル70KG、ポリセル40KGからなり、全ロジックは110KGである。メモリも含めた全素子数は910K素子であり、0.8μmで製造した最大級のLSIである。

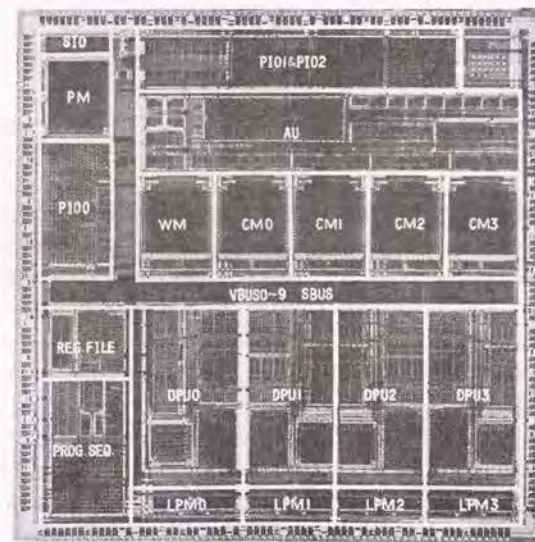


図6.6.12 IDSPのチップ写真

表 6.6.2 I D S P の諸元

Data processing unit (DPU)	
No. of DPUs	4 - parallel
Pipeline stage	3 - stage / DPU
ALU and MPL data length	16b
Accumulator data length	24b
Data RAM	
Dual-port cash	512W×16b×4bank
Dual-port work	512W×16b
Register file	16W×16b
Extended	1MW / port
Micro-program memory	
Main program RAM	512W×32b
Local program RAM	16W×32b×2bank / DPU
Boot ROM	32W×32b
Extended	64KW / port
Internal BUS	
Vector data BUS	16b×10
Scalar data BUS	16b
Program BUS	32b
Program load BUS	32b
Sequencer	
Loop	10 level
Stack	4 level
Interrupt	5 level
I / O port	
32b / 16b parallel	3 pair
Serial	1 pair
Instruction cycle	> 25MHz
Pin counts	280 (205 signal pins)
Power dissipation	1.0W max at 25MHz
No. of elements	910K
Die size	15.2mm×15.2mm

表 6.6.3 I D S P のビデオ符号化の性能

(for 16×16 pixel-size and 25MHz operation)

	Exec. Time (μ sec)	Performance (MOPS)	Ave / Max (%)
• 3×3 2-D FIR filter	23.0	288	96 (#1)
• 8×8 2-D DCT and inverse DCT	41.8	196	98 (#2)
• Motion compensation	35.4	178	59 (#1)

(#1) Max performance = 300 MOPS (ALU + MPL + ADD / SUB)

(#2) Max performance = 200 MOPS (MPL + ADD / SUB)

(2) 性能評価

アーキテクチャ評価のために、画像符号化の代表的なベクタ処理を取り上げ、マイクロ命令の実ステップ数を算出した。そのステップ数から計算したビデオ符号化の性能を、表 6.6.3 に示す。当然のことであるが、2次元フィルタや2次元DCTのような定型処理の平均処理効率は、非常に高い。このオーバーヘッドは、システムセットアップ命令とパイプライン遅延のみである。代表的適応処理である動き補償では、動きベクトル検出、フレーム内／フレーム間適応符号化、および適応量子化を含んでいるが、この場合にも、最高性能と比較して、59%という高い平均処理効率を達成している。これは、モディファイドSIMDの並列処理技術と、機能ブロック間の接続替えを実行命令と並列に行う機能との効果である。また本DSP-LSIを4チップ用いて、H.261標準でCIF画面サイズの1ボードCODECを試作した[28][34]。このCODECは、毎秒12.5フレームまでの処理性能を有し、世界初のDSPによる標準化仕様のCODECである。現在テレビ電話、テレビ会議等のビジュアル通信のキーデバイスとして、広く使用されている。

6.6.8 まとめ

最後に、本ビデオDSPのデザインコンセプトをまとめる。このDSPは、以下の2つの目標を両立させた点が最大の特長である。第1点は、すでに標準化されたH.261に従う標準CODECを、コンパクトに構成できる性能を持たせたことである。第2点は、上記CODECだけではなく、ビデオ処理や新しい符号化アルゴリズムに対応できる柔軟性を持たせたことである。この目標実現のために考案した技術とその特長を、以下に列挙する。

- (A) 4個のパイプライン演算器を搭載した並列処理を基本とした。またモディファイドSIMD技術を提案して、定型処理ばかりでなく適応処理に対しても並列処理ができるようにした。この技術により、同一分岐の演算器をグループ化して並列処理できるようになり、適応アルゴリズムに対しても、処理効率が向上した。
- (B) 3組のパラレルポートを搭載させ、階層的な負荷分散型および機能分散型のマルチプロセッサシステムを構成できるようにした。また各ポートにDMA機能を付与し、命令と並列に外部とのデータ転送ができるようにした。これにより、画素ブロック単位でのパイプライン処理が可能となり、処理効率が向上した。
- (C) 2次元アドレス生成技術と、並列プロセッサの2階層マイクロ命令制御技術とにより、それぞれデータメモリ容量とプログラムメモリ容量を大幅に削減した。この技術により、4並列演算器を搭載した大規模DSPを、1チップに集積できた。
- (D) ブートストラップ機能、オンチッププログラムデバック機能、プログラムのダウンロード機能、マルチDSPのバスアービトレーション機能、2チャンネルのシリアルポート等、DSP単独で動作する機能を搭載し、システム構築を容易とした。

本章では、LSIの技術制約をアーキテクチャまたは回路技術で解決して、実際に高速の信号処理LSIを実現する手法について述べた。

第2節では、究極的な高速加算器である冗長2進加算器に着目し、回路規模を大幅に削減できる手法を検討した。冗長2進コードの割り振り方を改善して回路の共通化を図り、回路規模を通常2進加算器に比較して、17%の増加に押さえることができた。一方冗長2進加算器の加算速度は、例えば20ビット語長のとき、CLA回路に比して約2倍高速である。このため、上記冗長2進回路の実用化は、今後信号処理LSIを高速化する上で、飛躍的な進歩を期待できる。

第3節では、上記の冗長2進加算器をLSIに適用して効果を実証した。その例として、LSI化の利点を活かしやすいCORDICアルゴリズム（加算とシフトの繰り返し演算で初等関数を生成できる）に着目し、パイプライン型冗長2進加算器アレイでCORDIC関数発生器を構成し、幾何学変換のアドレスを生成をするLSIを取り上げた。性能目標は、高精細動画の幾何学変換を実時間処理できることとした。冗長2進加算器技術に加えて、データ転送でのクロックスキュー低減に効果的なコントロールストリーム技術を考案して適用し、CMOS 1.2 μ mプロセスで、演算語長20ビット、演算レート70MHzと、既存の幾何学変換LSIに比較して2.5～10倍の高速性を実現した。この性能は、1.2 μ mプロセスでは、世界最高の演算サイクルである。これにより、冗長2進加算器の高密度・高速技術と、コントロールストリームによる高速制御技術の優位性を実証した。

第4節では、並列乗算器のハードウェア削減に効果的な「1加算法」を取り上げ、符号ビットを拡張した部分積の加算と1加算法に基づく加算とが、等価であることを証明した。

第5節では、信号処理LSIで大きな占有面積を占めるオンチップのプログラムメモリとデータメモリの容量を削減する技術について検討した。プログラムメモリの使用効率を高める技術として、並列プロセッサにおける2階層マイクロプログラム制御方式を提案した。この方式は、マイクロフィールド長が長く、スタティックステップが短く、そしてダイナミックステップの長いベクタ演算に有効である。試作したビデオDSPの例では、メモリ容量を150Kbから20.5Kbへ削減することに成功した。一方データメモリでは、入出力の原データと演算過程での加工データとの、ダイナミックレンジの差を利用し、データにコンパクションをかけてストアする手法を検討し、実際のビデオDSPで有効なことを示した。また高密度デュアルポートRAMの構成法として、PMOSとNMOSの数が等しくなる相補型メモリセル構造を提案し、従来のデュアルポートRAMと比較して、セル面積を約30%削減できることを示した。

第6節では、前節でのオンチップメモリ容量削減技術を使って、実際に大規模なビデオDSPを開発し、技術の有効性を実証した。このDSPは、動画CODECを1ボードで実現する上でのキーデバイスであり、従来DSPの1桁上の性能が要求されていた。このため、オンチップメモリ容量削減技術と共に、3～6章で述べてきた種々の信号処理LSI高速化技術を適用した。そして、0.8 μ m技術で、DSPでは世界最高性能である300MOPSの性能を実現した。

- [1] 山内、"DSPの技術動向と応用事例"、1992年春期全国大会併催事業講習会予稿「デジタル信号処理プロセッサ」、pp.2-21, (Mar., 1992).
- [2] N.Takagi, H.Yasuura, S.Yajima, "AVLS-oriented high-speed divider using redundant binary representation", IECE Trans. J67-D, No.4, pp.451-457, (1984).
- [3] A.Avizienis, "Signed-digit number representation for fast parallel arithmetic", IRE Trans. EC, Vol. EC-10, pp.389-400, (1961).
- [4] T.Nakanishi, H.Yamauchi, "CMOS Radix-2 Signed-Digit Adder", IEICE Trans. Vol. E69, pp.261-266, (Apr., 1986).
- [5] 仲西、山内、吉村、"CMOS高速SD加算器の検討"、昭和61年度電子情報通信学会総合全国大会、No.429, (1986).
- [6] Z. Stroll, et al., "VLSI-based Image Resampling for Electronics Publishing", VLSI for Pattern Recognition and Image Processing, pp. 209-229, Springer. Verlag.
- [7] H. Yoshimura, et al., "A 50 MHz CMOS geometrical mapping processor", ISSCC Technical Digest, pp. 162-163, (1988).
- [8] J. S. Walther, "A unified algorithm for elementary functions", SJCC, pp. 379-385, (1971).
- [9] J. E. Volder, "The CORDIC trigonometric computing technique", IEEE Trans. Electron. Comput, EC-8, pp. 330-334, (Sep., 1959).
- [10] 田端光男、置川尚、"画像メモリ用アドレス計算LSI"、昭和60信学総会大、1264.
- [11] 仲西、山内、吉村、"CMOS 50MHz幾何変換LSI"、電子情報通信学会論文誌、Vol. J72-C-2 No.5, pp.354-361, (1989).
- [12] H.Yoshimura, T.Nakanishi, H.Yamauchi, "A 50-MHz CMOS Geometrical Mapping Processor", IEEE trans. on Circuits and Systems, Vol.36, No.10, pp.1360-1364, (Oct., 1989).
- [13] H.Yamauchi, T.Nikaido, T.Nakashima, Y.Kobayashi, T.Sakai, "10ns 8x8 Multiplier LSI Using Super Self-Aligned Process Technology", IEEE J. Solid-State Circuits, Vol. SC-18, No.2, pp.204-210, (Apr., 1983).
- [14] 山内、二階堂、中島、小林、酒井、"完全拡張機能を内蔵した超高速8x8ビット乗算器LSI"、電子情報通信学会論文誌、C, Vol.C-37, pp.353-360, (1983).
- [15] 松谷、山内、"DSP1の2ポートデータRAM"、昭和60年度電子情報通信学会総合全国大会、No.470, (Mar., 1985).
- [16] 山内、金子、渡辺、"音声のフレーム処理に適した信号プロセッサのメモリ構成"、昭和58年度電子情報通信学会総合全国大会、No.517, (Mar., 1983).
- [17] S.Nakagawa, H.Terane et al., "A 24-bit 50ns Digital Image Signal Processor", IEEE Journal of Solid-State Circuits, vol.25, no.6, pp.1484-1493, (Dec., 1990).
- [18] K.Kikuchi, N.Yasuaki et al., "A Single-Chip 16-bit 25ns Realtime Video/Image Signal Processor", in proc. IEEE-ISSCC'89, pp.170-171, (Feb., 1989).
- [19] I.Tamitani, H.Harasaki et al., "A Real-Time Video Signal Processor Suitable for Motion Picture Coding Applications", IEEE Trans. on Circuits and Systems, vol.36, no.10, pp.1259-1266, (Oct., 1989).

- [20] M.Murayama, H.Uwabu et al., "VLSI Architecture and Implementation of a Multi-Function, Forward/Inverse Discrete Cosine Transform Processor", in Proc. The International Society for Optical Engineering Visual Communications and Image Processing 90, pp.1343-1360, (Oct., 1990).
- [21] T.Fukushima, Y.Kobayashi, "An Image Signal Processor", in Proc. IEEE-ISSCC'83, pp.258-259, (Feb., 1983).
- [22] Y.Kobayashi, T.Fukushima et al., "A BiCMOS Image Signal Processor with Line Memories", in Proc. IEEE-ISSCC'87, pp.182-183, (Feb., 1987).
- [23] T.Kondo et al., "An LSI Adaptive Array Processor", IEEE Journal., Vol. SC-18, No.2, pp.147-156, (Apr., 1983).
- [24] A.Kanuma, M.Noda et al., "A 20MHz 32b Pipelined CMOS Image Processor", in Proc. IEEE-ISSCC'86, pp.102-103, (Feb., 1986).
- [25] H.Yamauchi, Y.Tashiro, T.Minami, Y.Suzuki, "Architecture and Implementation of a Highly-Parallel Single-Chip Video DSP", IEEE Trans. on Circuit and Systems for Video Technology, Vol. CSVT., No.2, pp.207-220, (Jun., 1992).
- [26] H.Yamauchi, Y.Tashiro, T.Toshihiro, Y.Suzuki, "A Highly-Parallel Single-Chip DSP Architecture for Video Signal Processing", Proc. IEEE-ICASSP'91, pp.1197-1200, (May., 1991).
- [27] T.Minami, R.Kasai, H.Yamauchi, Y.Tashiro, J.Takahashi, "A 300-MOPS Video Signal Processor with a parallel Architecture", IEEE J. Solid-State Circuits, Vol. SC-26, No.12, (Dec., 1991).
- [28] Y.Suzuki, T.Tajiri et al., "Single Board Video Codec using VLSIs for 64/128 Kbit/s CIF Video", in Proc. Third International Workshop on 64 Kbit/s coding of Moving Video, (Sep., 1990).
- [29] Y.Tashiro, H.Yamauchi et al., "An organized Firmware Verification Environment for the Programmable Image DSP", in proc. IEEE-ITC'91, pp.1034-1041, (Sep., 1991).
- [30] B.G.Lee, "A New Algorithm to Compute the Discrete Cosine Transform", IEEE Trans. on ASSP, vol. ASSP-32, no.6, pp.1243-1245, (1984).
- [31] H.Fukuda, S.Horiguchi et al., "A BiCMOS Channelless Masterslice with On-Chip Voltage Converter", in Proc. IEEE-ISSCC'89, pp.176-177, (Feb., 1989).
- [32] K.Takeya, S.Horiguchi et al., "A Generator for High-Density Macrocells with Hierarchical Structure", in Proc. IEEE-CICC'89, pp.23.5.1-23.5.4, (May., 1989).
- [33] T.Minami, H.Yamauchi et al., "A 300-MOPS Video Signal Processor with a Parallel Architecture", in Proc. IEEE-ISSCC'91, pp.252-253, (Feb., 1991).
- [34] T.Tajiri, S.Suzuki et al., "Single Board Video CODEC for ISDN Visual Telephone", in Proc. IEEE-ICASSP'91, p.30.M10.7, (May., 1991).

第7章 結論

7.1 本研究で得られた成果

デジタル信号処理LSIは、リアルタイム制御、音声処理、画像処理、ビデオ処理等広範囲に普及してきているが、いずれも時代の先端技術を使った最も高速なLSIであり、LSIの高速化技術の最先端を歩んで来たと言える。そして常により高速な技術を要求し続けている。本研究では、この「デジタル信号処理LSIの高速化技術」に焦点を絞って研究した。そして種々の高速化技術を提案した。さらに提案した高速化技術の効果を検証するため、実際にいくつかの先端信号処理LSIを開発して確認し、そして種々の装置に応用して実証した。以下に本研究で得られた成果のうち重要な項目をまとめて示す。

[第2章]

- (1) 高速化の方針について検討し、高速演算回路構成技術、データメモリアドレス生成技術、並列処理技術、およびLSI技術制約の解決技術の4点を、研究項目として明確化した。

[第3章]

- (2) 並列乗算器の高速化技術を検討し、モディファイドCSAアレイ方式を提案した。また上記アレイ方式を、BoothデコードおよびCLA加算器と組み合わせて、高速乗算器を設計・試作し、世界最高速を実証した。設計にあたっては、安定化容量を内蔵したL-CML回路技術、デバイスパラメータ自動測定技術、およびSSTトランジスタの回路設計技術を確立した。また並列乗算器の完全拡張機能を提案し、チップをアレイ状に配置するだけで、任意の語長の乗算器を構成できる方法を示した。
- (3) バイブライン型の浮動小数乗算器と浮動小数加減算器の高速化手法を検討した。浮動小数乗算器では、異常処理を含めてバイブラインの切れ目を最適化して高速化する設計法を示した。浮動小数加減算器では、指数部の減算と仮数部の桁合わせシフトとを並列に実行する高速化手法、仮数部のシフト減算を桁上がり伝搬のない1の補数演算で近似する高速化手法、およびキャリー選択加算器を使った高速正規化手法を提案した。そしてそれぞれ、16%から40%高速化できることを示した。また上記高速化技術を確認するため、具体的に6段バイブラインの複素バタフライ演算回路を試作し、世界最高速を実証した。

[第4章]

- (4) 音声信号処理に必要なアドレスを汎用的に生成できるアドレス生成ユニットの構成法を検討した。まず音声信号処理に必要なアドレスモードを抽出し体系化した。次に体系化に基づき、バタフライ、モジュロ、イテレーション等の各モードを、統一した回路で構成するハードウェアアルゴリズムを提案し、実現した。
- (5) 上記アドレスユニットを搭載した高速音声DSPを開発した。まず、複雑なベクタ演算もバイブラインを崩さずに実行できることを示し、本アドレスユニットの高機能性

を実証した。また有限語長シミュレータによる演算語長の最適化、メモリマップの解析によるオンチップメモリ容量の最適化等、汎用DSPの設計手法を確立した。さらに繰り返し処理のオーバーヘッドをなくす、リピートやディレイドジャンプ等、命令レベルでも高速化した。そして、1.6 kb/sのAPC-AB音声CODECを1チップで実現できることを、世界で最初に示し、アドレス技術を始めとする高速化技術の効果を実証した。

- (6) 複素バタフライ演算を1マシンサイクルで実行できる、パイプライン型のバタフライ演算器を実現するため、核となる技術である、データメモリ構造とそのアドレス生成について検討した。そしてアドレス生成では、1次元フーリエ変換毎に外部メモリとのインタリーブ処理を連続的に行う方法を提案した。またデータメモリ構造では、バタフライを処理単位とするパイプライン演算を、すべてのバタフライステージについて連続して行える4バンク4ポートRAMを提案した。この成果を、項目(3)で述べた浮動小数点演算器の高速化技術と融合させ、400MOPSという最高速の2次元フーリエ変換LSIを開発して、技術の有効性を実証した。
- (7) ビデオ符号化処理で重要なアドレス生成について検討し、2つの技術を提案した。1つは、任意サイズの画素ブロック(2次元)を切り出してスキャンするアドレス生成技術であり、もう1つは、複数の演算器が、バンク分けされた多ポートメモリを共通に使用して並列処理するためのアドレス生成技術である。特に後者は、オンチップメモリ容量の制限を解決して並列処理する技術であり、LSI化に効果的である。

[第5章]

- (8) 電子ビーム描画装置のリアルタイム制御を取り上げ、パイプラインサイクルがデータ依存性を持っている場合の適切なパイプライン制御技術を検討した。そして、計算ステージと描画ステージの処理時間が共にデータ依存性を持っている場合や、描画ステージの処理時間にデータ依存性がありかつ計算終了と同時に描画ステージに入ることが要求される場合について、それぞれ適切な処理法を提案した。また実際に、パイプボラとBiCMOSとによりLSIを試作し、1.5 ns分解能で効率良く動作することを実証し、技術の有効性を示した。
- (9) 複数のプロセッサエレメントが、それぞれの内部状態によって異なる分岐をする場合の効果的な並列処理法を提案した。これは、SIMD制御を改良した方法で、プログラマが全く意識しなくても、同一分岐のプロセッサを自動的にグループ化して並列処理する方法であり、モディファイドSIMD技術と名付けた。この技術は、僅かな回路の増加で実現できるのが特長である。この方式を、実際にビデオDSP-LSIに搭載して適応量子化処理を行わせ、有効であることを実証した。

[第6章]

- (10) 冗長2進加算器を実用に供する上での最大の問題である、回路規模削減法を検討した。そして、加算器回路の論理の共通化を図れる新たなコード化を提案し、回路規模を約2分の1に削減した。そして実際にレイアウトを行い、通常2進のCLA加算器と比較して、回路面積が同等になることを示した。

- (11) 提案した冗長2進加算器の高速性と高密度性を実証するため、本加算器を使って、高精細動画の幾何学変換処理を実時間で行うLSIを開発した。CORDICアルゴリズムを冗長2進加算器アレイで構成する技術と、パイプラインステージ間のクロックスキューを低減するコントロールストリーム技術を提案し、1.2 μ m CMOSで集積した。70 MHz動作を確認し、技術の有効性を実証した。
- (12) 複数の演算プロセッサを搭載した並列DSPの、プログラムメモリ容量を削減する技術を提案した。長いマイクロフィールドの中から、演算プロセッサで行う制御を切り出してローカルメモリ化する技術であり、並列プロセッサにおける2階層マイクロプログラム制御技術と名付けた。そしてこの技術をビデオDSPで使用し、メモリ容量を画期的に低減できることを示した。
- (13) 上記のメモリ容量削減技術を始め、種々の高速化技術を取り入れて、動画画像符号化を実時間処理する、非常に処理能力の高い並列ビデオDSPの構成法を検討し、そのアーキテクチャを提案した。そして具体的に0.8 μ m CMOS技術で試作し、300MOPSの性能を達成した。またこのDSPを使って、世界で始めて、DSPによる1ボードビデオCODECを実現した。

7.2 今後の信号処理LSIの研究課題

LSI技術は今後も発展を続け、西暦2000年には0.2 μ mの技術が実用に供され则认为られている。この時の信号処理LSIの性能を類推すると、汎用DSPで5 GOPS (Giga Operation Per Second)、専用DSPで50 GOPS程度を期待することができる。

表 7.2.1 ビデオ符号化サービスの動向

サービス	符号化レート (b/s)		方式課題	標準化動向	DSP課題	実現可能時期	
	現状	2000年				743' 92' 4	DSP技術
携帯テレビ電話	-	8 k	知的符号化	-	10-GOPS, 100-mW 大容量メモリ	-	2000年
テレビ電話	64 k	64 k	高品質化	H.261	従来技術の延長	○	○
テレビ会議	384 k	384 k	高品質化	H.261		○	○
マルチメディア CATV	-	1.5 M	-	MPEG1		○	○
デジタルビデオ ディスク	1.5 M	6 M	-	MPEG2 H.262		△	△
デジタルCATV ビジュアルテレホン	32 ~ 45M (NTSC)	6 M	B-ISDN	MPEG2 H.262	100-GOPS, 1ボード	△	△
映像会議 ビデオシスター	100 ~ 130M (HVS)	15 ~ 30M	B-ISDN 臨場感	CCIR/CMITT で検討中		△	2000年
超高精細品質通信	~ 600M	~ 100M	B-ISDN 質感通信	-		-	2005年

○ 開発済み △ 開発中 - 研究中

この処理能力を活かす分野は、マルチメディアハンドリングである。特に高精細画像の符号化は、マルチメディアに共通する基盤技術であり、LSI技術にける期待は大きい。表7.2.1は、現在および将来のビデオ符号化の動向を示したものであり、2000年にはHDTV-CODECが、2005年にはUHTV (Ultra High-definition TV) -CODECが、それぞれ1ボードで実現できると期待される。またそのときのLSI技術は、それぞれ0.2 μ mと0.1 μ mになると思われる。

表7.2.2 2000年のSIMD型並列DSPの予測諸元

項目	内容
最大性能	64 GOPS
内部クロック	250 MHz
並列度	256 (3段X64並列)
ベクタプロセッサ	(64並列)
スカラープロセッサ	(64並列)
画像データ入力	2Gb/s (80b x 25MHz)
画像データ出力	2Gb/s (80b x 25MHz)
メモリ	
フレームメモリ	64Mb (2MW x 16b x 2)
レジスタファイル	6Mb (2KW x 24b x 2 x 64)
プログラムメモリ	1Mb (16KW x 64b)
ローカルプログラム	8Mb (2KW x 32b x 2 x 64)
電源電圧	2 V
消費電力	5 W
素子数	480M (1.5MG+79Mb)
アーキテクチャ	
チップ内並列	SIMD/MIMD
チップ間並列	密結合 (数十並列)

表7.2.3 2000年の疎結合型並列DSPの予測諸元

項目	内容
最大性能	3GFLOPS + 10GOPS
内部クロック	250 MHz
並列度	52 (13 x 4) 要素プロセッサ4個並列
要素プロセッサ	
ベクタプロセッサ	3 演算 (浮動小数点)
スカラープロセッサ	10 演算 (固定小数点)
リンクプロセッサ	4 リンク
データ入出力	3.2Gb/s (32b x 25MHz x 4)
メモリ	
レジスタファイル	16Mb (500KW x 32b)
プログラムメモリ	1Mb (16KW x 64b)
電源電圧	2 V
消費電力	2 W
素子数	413M (1.2MG+68Mb)
アーキテクチャ	
制御	MIMD
結合	疎結合 (スケラブル)

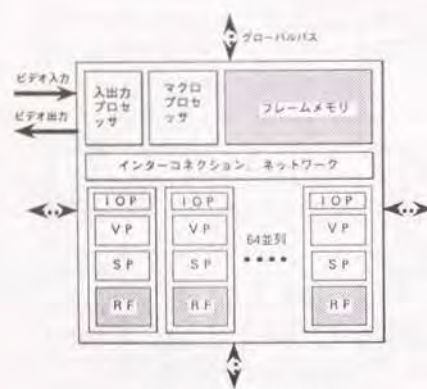


図7.2.1 SIMD型並列DSP

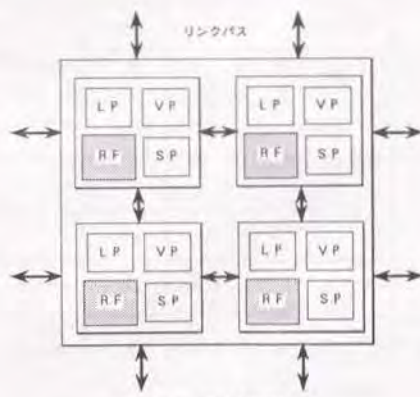


図7.2.2 疎結合型並列DSP

上記目標に向かって、信号処理LSIは一層の高速化技術を開発していかなければならない。その技術は、消費電力を抑えながら演算サイクルを高速化することと、オンチップ化された数十のプロセッサを効率よく並列処理することである。このうち演算サイクルは、配線負荷と寄生効果の増大、および低電圧化によるトランジスタのドライブ能力の低下のため、500MHz程度で飽和することが予想されるので、超並列処理技術による向上が一層重要となる。

並列DSPでは、以下の3つの方法が有望である。第1は、専用のコアDSPをアルゴリズムのフローにそって配置した、「コアDSPによる並列処理ASIC」、第2は、「SIMD型並列DSP」、第3は、「疎結合型並列DSP」である。このうち、第1の並列処理ASICは、従来の信号処理LSI技術の延長上で専用設計することで、解決できると思われる。そこで、以下第2と第3の方法について見解を述べる。

SIMD型並列DSPは、1つのプロセッサエレメントに数百W程度のデータメモリを搭載して、画素ブロック単位での並列処理を行わせることになる。そして西暦2000年には、1チップに256プロセッサ程度を搭載し、50GOPS以上の性能を実現するようになると思われる。その推定諸元と構造をそれぞれ表7.2.2と図7.2.1に示す。この構造の技術課題の第1は、大局処理と適応処理に如何に対応するかである。特に、プロセッサ間のデータ交換を如何に効率良く行うかが重要である。第2の課題は、チップ外部と各プロセッサ間のデータ入出力の高速化である。即ち、内部処理の高速化に比例してデータ入出力の高速化を図らねばならない。第3の課題は、ウェハインテグレーション技術との融合である。消費電力の分散とチップの面積化の方向に添って、CODECや端末等のシステム全体を最適に集積する技術が必要となる。

疎結合型並列DSPは、メッセージパッシング型の通信機能を備えたプロセッサによる、独立性の高い並列システムである。それぞれのプロセッサが独立して動作し、各々が、数十～数百の画素ブロックを処理するので、独立したコンピュータ機能と100KW以上のデータメモリを必要とする。その推定諸元と構造をそれぞれ表7.2.3と図7.2.2に示す。この構造は汎用性に富んでおり、大局処理や適応処理への対応も比較的容易である。しかし、資源の共有を行わない分SIMD並列に比して回路規模当りの処理能力が低い。即ち、ハードウェアの利用効率の悪さを解決する技術開発が望まれる。また、大局処理の効率をあげるため、大規模な高速通信バスの設定法も検討しなければならない。さらに、ユーザのプログラミングを容易にするため、ソフトウェア開発環境も重要である。特にビデオ処理の場合には、多くのライブラリを揃えて効率の良いコンパイラを作る必要がある。

以上、信号処理LSIは、今後も時代を開くキーデバイスであり続けるであろう。そして、一層の高速化が進められていくと期待される。

謝辞

本論文をまとめるにあたり、終始懇切なる御指導、御教示を賜りました、東京大学工学部浅田邦博助教授、羽鳥光俊教授、田中英彦教授、原島博教授、東京大学生産技術研究所喜連川優助教授、並びに東京大学先端科学研究センタ岡部洋一教授に謹んで感謝の意を表します。

本研究は著者が日本電信電話公社（1985年以降株式会社）において研究実用化の一環として行ったものであり、本論文をまとめることをお進めくださると共に多大なる御尽力を頂いた、企業通信システム本部安田浩部長、新分野事業推進部星野坦之博士に心から感謝致します。

また本研究の機会を与えてくださると共に御鞭撻を賜った、沖電気工業株式会社取締役向井久和技師長、NTTエレクトロニクステクノロジー株式会社代表取締役鈴木敏正社長、常務取締役井川彪総合企画本部長、取締役大和田充彦LSI生産本部長、株式会社村田製作所専務取締役有吉昶技術本部長、LSI研究所酒井徹志主席研究員、岩田穆部長、堀口勝治部長並びに酒井保良部長に感謝致します。

研究遂行にあたり信号処理アルゴリズムの立場から多くの御教示、御助言を頂いた、名古屋大学板倉文忠教授、金沢大学橋本秀雄教授、長崎大学黒田英夫教授、知的財産部青山友紀部長、伝送システム研究所小野定康博士、基礎研究所菅田雅彰博士、企業通信システム本部谷中一寿博士、NTTエレクトロニクステクノロジー株式会社鈴木豊部長、LSI研究所小林勉主幹研究員、光エレクトロニクス研究所渡部直也主幹研究員に感謝致します。

また高速演算回路構成技術について多くの有益な御討論、御助言を頂いた、NTTエレクトロニクステクノロジー株式会社の中島孝利部長、二階堂忠信部長、服部三四郎部長、木村隆部長、石谷恒八部長、LSI研究所小林由治主幹研究員、ヒューマンインタフェース研究所鈴木義武博士、高橋淳一主任研究員、三樹聡研究主任、および、データメモリアドレッシング技術について御討論、御助言を頂いた、日本A&T株式会社ジャパンテクノロジーセンタ宮永博史課長、ヒューマンインタフェース研究所金子孝夫主幹研究員、伝送システム研究所金山之治主任研究員、LSI研究所南俊弘主任研究員、さらに、並列処理技術について御討論、御助言を頂いた、LSI研究所渡辺隆主幹研究員、両沢哲男主幹研究員、ヒューマンインタフェース研究所田代豊主任研究員、そして、LSI技術制約の解決技術について御討論、御助言を頂いた、北海道大学雨宮好仁教授、NTTエレクトロニクステクノロジー株式会社宮原則男設計副本部長、ヒューマンインタフェース研究所吉村寛博士、仲西正主任研究員、LSI研究所武谷健主幹研究員、笠井良太博士、浜口重建

主任研究員、山田順三博士、北沢仁志博士、星野民夫主幹研究員、松谷康之主幹研究員に心から感謝申し上げます。

最後に信号処理LSIの設計、試作、評価において色々な形で御協力くださった、NTTエレクトロニクステクノロジー株式会社、LSI研究所、伝送システム研究所、ヒューマンインタフェース研究所の方々に心から感謝致します。

本研究に関する発表文献

1. 論文 (学会論文誌)

- (1) T.Sakai, Y.Kobayashi, H.Yamauchi, M.Sato, T.Makino, "High Speed Bipolar ICs Using Super Self-Aligned Process Technology", JJAP, Vol. 20-1, pp.155-159, (1981).
- (2) H.Yamauchi, T.Nikaido, T.Nakashima, Y.Kobayashi, T.Sakai, "10ns 8×8 Multiplier LSI Using Super Self-Aligned Process Technology", IEEE J. Solid-State Circuits, Vol. SC-18, No.2, pp.204-210, (Apr.,1983).
- (3) 山内, 二階堂, 中島, 小林, 酒井, "完全拡張機能を内蔵した超高速 8×8 ビット乗算器 LSI", 電子情報通信学会論文誌, C, Vol. C-37, pp.353-360, (1983).
- (4) 金子, 山内, "複素アドレス演算に適したアドレス制御法", 電子情報通信学会論文誌, Vol. J69-D No.4, pp.634-636, (Apr.,1986).
- (5) 金子, 山内, 岩田, "ビットリバースアドレスの制御法", 電子情報通信学会論文誌, Vol. J69-D No.7, pp.1124-1126, (Jul.,1986).
- (6) T.Nakanishi, H.Yamauchi, "CMOS Radix-2 Signed-Digit Adder", IEICE Trans. Vol. E69, pp.261-266, (Apr.,1986).
- (7) 金子, 山内, "高速浮動小数点 VLSI シグナルプロセッサ: DSSP1", 電子情報通信学会論文誌, Vol. J72-B-I No.1, pp.67-73, (Jan.,1989).
- (8) 仲西, 山内, 吉村, "CMOS 50MHz 幾何変換 LSI", 電子情報通信学会論文誌, Vol. J72-C-2 No.5, pp.354-361, (1989).
- (9) H.Yoshimura, T.Nakanishi, H.Yamauchi, "A 50-MHz CMOS Geometrical Mapping Processor", IEEE trans. on Circuits and Systems, Vol.36, No.10, pp.1360-1364, (Oct.,1989).
- (10) 永島, 増田, 山内, "地下埋設物探知技術におけるパターン認識技術とその信号処理の高速化の研究", 電子情報通信学会論文誌, Vol. J74-C-2 No.5, pp.317-324, (May.,1991).
- (11) H.Yamauchi, H.Miyanaga, "Architecture of a Floating-Point Butterfly Execution Unit in a 400-MFLOPS Processor VLSI and its Implementation", IEICE Transactions, Vol. E74, No.12, pp.3852-3860, (Dec.,1991).
- (12) H.Miyanaga, H.Yamauchi, "A 400-MFLOPS FFT Processor VLSI Architecture", IEICE Trans. Vol. E74, No.12, pp.3845-3851, (Dec.,1991).
- (13) T.Watanabe, T.Morosawa, N.Shimazu, H.Morita, H.Yamauchi, A.Iwata, "Reliability Enhancements for the Direct Wafer Exposure Electron Beam System EB60", J.Vac.Sci. Technol., B 9 (6), pp.3028-3032, (Nov./Dec.,1991).
- (14) T.Minami, R.Kasai, H.Yamauchi, Y.Tashiro, J.Takahashi, "A 300-MOPS Video Signal Processor with a parallel Architecture", IEEE J. Solid-State Circuits, Vol. SC-26, No.12, (Dec.,1991).
- (15) H.Yamauchi, Y.Tashiro, T.Minami, Y.Suzuki, "Architecture and Implementation of a Highly-Parallel Single-Chip Video DSP", IEEE Trans. on Circuit and Systems for Video Technology, Vol. CSVT., No.2, pp.207-220, (Jun.,1992).
- (16) S.Miki, H.Miyanaga, H.Yamauchi, "Design of a 4000-tap Acoustic Echo Canceller Using the Residue Number System and the Mixed-Radix Number System", IEICE Trans. Electron., Vol. E75-C, No.10, pp.1232-1240, (Oct.,1992).
- (17) H.Miyanaga, H.Yamauchi, Y.Nagashima, T.Hosaka, "A High-Speed Special Purpose Processor for Underground Object Detection", IEICE Trans. Electron., Vol. E75-C, No.10, pp.1250-1258, (Oct.,1992).
- (18) H.Yamauchi, T.Morosawa, T.Watanabe, A.Iwata, T.Hosaka, "Real-time Feed-Forward Control LSIs for Direct Wafer Exposure Electron Beam System", IEICE Trans. Electron., Vol. E76-C, No.1, pp.124-135, (Jan.,1993).

2. 論文 (査読付き国際会議)

- (1) S.Kozuka, N.Watanabe, S.Ono, H.Yamauchi, "Signal Processor Architecture for High Complex Coding", IEEE Global Conference, pp. 45-1-1, (Dec.,1983).
- (2) H.Yamauchi, T.Kaneko, T.Kobayashi, A.Iwata, S.Ono, "An 18-bit Floating-Point Signal Processor VLSI with an On-chip 512W Dual-port RAM", Proc. IEEE-ICASSP'85, pp.204-207, (Mar.,1985).
- (3) H.Yamauchi, T.Kaneko, J.Takahashi, A.Iwata, "Speech Signal Processor VLSI Family", Proc. International Workshop on Digital Signal Processing, pp.2a 1-4, (Jun.,1985).
- (4) T.Kaneko, H.Yamauchi, A.Iwata, "A 50ns Floating-Point Signal Processor VLSI", IEEE-ICASSP'86, pp.1600-1603, (May.,1986).
- (5) H.Yoshimura, T.Nakanishi, H.Yamauchi, "A 50MHz CMOS Geometrical Mapping Processor", IEEE-ISSCC'87, pp.162-163, (Feb.,1987).
- (6) Y.Suzuki, T.Tajiri, T.Takahashi, H.Yoshimura, Y.Kato, H.Yamauchi, "Single Board Video Codec using VLSIs for 64/128Kbit CIF Video", Third International Workshop on 64Kbit/s Coding of Moving Video, (Sep.,1990).
- (7) T.Minami, H.Yamauchi, Y.Tashiro, R.Kasai, J.Takahashi, "A 300-MOPS Video Signal

- Processor with a Parallel Architecture", IEEE-ISSCC'91, pp.252-253, (Feb.,1991).
- (8) H.Yamauchi, Y.Tashiro, T.Toshihiro, Y.Suzuki, "A Highly-Parallel Single-Chip DSP Architecture for Video Signal Processing", Proc. IEEE-ICASSP'91, pp.1197-1200, (May.,1991).
- (9) M.Miyanaga, H.Yamauchi, K.Matsuda, "A Real-Time 256×256 Point Two-Dimensional FFT Single-Chip Processor", IEEE-ICASSP'91, pp.1193-1196, (May.,1991).
- (10) T.Watanabe, T.Morosawa, N.Shimazu, H.Morita, H.Yamauchi, A.Iwata, "Development of Reliable and Comprehensive Direct Wafer Exposure Electron Beam System", The 35th international symposium on electron, ion & photon beams, R6, (May.,1991).
- (11) Y.Tashiro, H.Yamauchi, T.Minami, S.Hamaguchi, Y.Suzuki, "An Advanced SIMD Control Technique for Concurrent Adaptive Processing", Proceedings of JTC-CSCC'91, pp.252-255, (Jul.,1991).
- (12) Y.Tashiro, H.Yamauchi, T.Minami, Y.Suzuki, "An Organized Firmware Verification System for the Programmable Image DSP", IEEE-ITC'91, (Nov.,1991)

3. 解説論文 (学会誌等)

- (1) 宮永、山内、" 剰余数演算アルゴリズムのLSI応用への可能性"、電子情報通信学会誌 学生のページ Vol.73, pp.1003-1007, (Sep.,1990).
- (2) 中野、山内、大庭、" 画像処理・画像ファイル"、電気通信, Vol.54, No.540, pp. 25-33, (Mar.,1992).
- (3) 吉村、山内、" 画像処理用LSIの動向"、テレビジョン学会誌, Vol.46, No.3, pp.233-238, (Mar.,1992).
- (4) 山内、" DSPの現状と将来展望"、電子材料, pp.38-44, (June,1992).

4. 論文 (研究会等)

- (1) 山内、小中、酒井、" バイポーラトランジスタ、デバイスパラメータ評価システム"、電子情報通信学会技術研究報告、SSD81-20, pp.93-100, (1981).
- (2) 酒井、小林、山本、山内、" 高速バイポーラLSI技術: SST"、電子情報通信学会技術研究報告、EDD81-18, pp.53-60, (1981).
- (3) 山内、二階堂、中島、小林、酒井、" 完全拡張機能を内蔵した超高速乗算器LSI"、電子情報通信学会技術研究報告、SSD81-52, pp.7-14, (1981).

- (4) 加本、小原、山内、酒井、首藤、" モノリシックIC化超高速増幅器"、電子情報通信学会技術研究報告、ED81-126, pp.75-80, (Jan.,1982).
- (5) 渡部、小野、鶴沢、山内、" 音声符号化用プロセッサのアーキテクチャ"、電子情報通信学会技術研究報告、CS82-87, pp.89-96, (Nov.,1982).
- (6) 金子、山内、" デジタル音声信号処理に適した汎用アドレス制御法"、電子情報通信学会技術研究報告、CAS82-194, pp.57-62, (Mar.,1983).
- (7) 宮永、山内、松田、" 250MFLOPS 2次元FFTプロセッサVLSIアーキテクチャ"、電子情報通信学会技術研究報告、VLD89-100, pp.23-30, (Mar.,1990).
- (8) 山内、両沢、渡辺、平田、岩田、" フィードフォワード制御用ASICs"、電子情報通信学会技術研究報告、ICD91-9, pp.1-8, (Apr., 1991).
- (9) 山内、笠井、南、田代、鈴木、" 蜜結合型高並列ビデオプロセッサ"、回路とシステム軽井沢ワークショップ予稿, pp.355-360, (Apr.,1991).
- (10) 南、山内、田代、鈴木、笠井、高橋、遠藤、浜口、" ビデオシグナルプロセッサIDSPのデータフロー制御"、電子情報通信学会技術研究報告、ICD91-12, pp.25-32, (Apr.,1991).
- (11) 山内、" DSPの現状と将来展望"、電子情報通信学会技術研究報告、ICD91-98, pp.1-8, (Sep.,1991).

5. 論文 (講演会)

- (1) 山内、酒井、" SET構造によるラテラルトランジスタの計算結果"、昭和52年度電子情報通信学会総合全国大会、No.321, (1977).
- (2) 二階堂、石谷、山内、" 超高速LSIにおける電流切り替え型回路の設計"、昭和54年度電子情報通信学会総合全国大会、(Mar.,1978).
- (3) 山内、小林、酒井、" SST-2によるLCMLゲートの特性"、昭和56年度電子情報通信学会総合全国大会、No.334, (Mar.,1981).
- (4) 山内、酒井、二階堂、" NTLゲートのスピードアップコンデンサの最適化の解析"、昭和56年度電子情報通信学会総合全国大会、335, (Mar.,1981).
- (5) 小中、山内、小林、" IC化超高速トランジスタのパラメータ評価"、昭和57年度電子情報通信学会総合全国大会、No.433, (Mar.,1981).
- (6) 山内、二階堂、中島、小林、酒井、" SST技術を用いた超高速 8×8 乗算器"、昭和56年度電子情報通信学会半導体材料部門全国大会、No.96, (Sep.,1981).
- (7) 山内、小中、酒井、" 高性能安定化容量を内蔵したLCMLゲートの特性"、昭和5

- 7年度電子情報通信学会総合全国大会、No.385, (Mar.,1982).
- (8) 金子、山内、岩田、"音声信号処理に適した汎用アドレスユニットの構成"、昭和57年度電子情報通信学会通信部門全国大会、No.65, (Sep.,1982).
- (9) 鈴木、山内、"高速ビット検出回路を備えた浮動小数点加算器"、昭和57年度電子情報通信学会通信部門全国大会、No.66, (Sep.,1982).
- (10) 山内、鈴木、岩田、"デジタル信号プロセッサに適した除算法"、昭和58年度電子情報通信学会通信部門全国大会、No.67, (Sep.,1982).
- (11) 山内、金子、渡辺、"音声のフレーム処理に適した信号プロセッサのメモリ構成"、昭和58年度電子情報通信学会総合全国大会、No.517, (Mar.,1983).
- (12) 金子、山内、岩田、"適応ビット配分浮動小数点演算方式"、昭和58年度電子情報通信学会半導体・材料部門全国大会、No.150, (Sep.,1983).
- (13) 山内、金子、"信号プロセッサLSIのマルチ接続機能"、昭和58年度電子情報通信学会半導体材料部門全国大会、No.164, (Sep.,1983).
- (14) 小林、山内、岩田、"APC-AB CODEC-LSIにおける有限語長演算の評価"、昭和58年度電子情報通信学会半導体・材料部門全国大会、No.199, (Sep.,1983).
- (15) 金子、山内、"複素ベクトル演算に適したアドレス制御法"、昭和59年度電子情報通信学会総合全国大会、No.337, (Mar.,1984).
- (16) 山内、金子、小野、"リアルタイムプロセッサにおける専用DMAを用いたI/O制御方式"、昭和59年度電子情報通信学会総合全国大会、No.441, (Mar.,1984).
- (17) 山内、金子、"冗長コードを用いた浮動小数点加算器の検討"、昭和60年度電子情報通信学会情報システム部門全国大会、No.429, (Sep.,1985).
- (18) 松谷、山内、"DSSP1の2ポートデータRAM"、昭和60年度電子情報通信学会総合全国大会、No.470, (Mar.,1985).
- (19) 金山、小野、山内、"高性能DSP" DSSP1"のCODECの適用に関する検討"、昭和60年度電子情報通信学会総合全国大会、(Mar.,1985).
- (20) 岩田、山内、金子、小野、"汎用音声信号プロセッサLSI (DSSP1)"、昭和60年度電子情報通信学会総合全国大会、No.468, (Sep.,1985).
- (21) 金子、山内、岩田、"DSSP1の浮動小数点ALU"、昭和60年度電子情報通信学会総合全国大会、No.469, (Sep.,1985).
- (22) 仲西、山内、吉村、"CMOS高速SD加算器の検討"、昭和61年度電子情報通信学会総合全国大会、No.429, (Mar.,1986).
- (23) 宮永、山内、吉村、"剰余数系を用いた離散コサイン変換回路の検討"、昭和61年度電子情報通信学会総合全国大会、No.1158, (Mar.,1986).
- (24) 金子、山内、岩田、"高速正規化浮動小数点方式VLSI信号プロセッサ"、昭和6

- 1年度電子情報通信学会通信部門全国大会、No.S10-1, (Sep.,1986).
- (25) 宮永、山内、吉村、"マルチFFTプロセッサにおけるオンチップメモリ容量削減法"、昭和61年度電子情報通信学会通信部門全国大会、No.55, (Sep.,1986).
- (26) 山内、堀口、"超高速CMOS/BiCMOS専用プロセッサの諸技術と将来展望"、昭和63年電気情報関連連合大会、No.18-7, (1988).
- (27) 山内、"LSI設計技術"、平成元年電子情報通信学会生涯教育秋季講座、(Nov.,1989).
- (28) 山内、"デジタル信号処理LSIの設計"、第20回記念インターネブコン技術セミナー予稿、No.AD8-1, (Jan.,1991).
- (29) 南、山内、田代、笠井、高橋、浜口、遠藤、"300MOPS動画処理用DSP"、平成3年度電子情報通信学会総合全国大会、No.SC-8-8, (1991).
- (30) 山内、"DSPの技術動向と応用事例"、1992年春期全国大会併催事業講習会予稿「デジタル信号処理プロセッサ」、pp.2-21, (Mar.,1992).

