
画像処理プロセッサRi2001開発

Development of Image Processor Ri2001

岩永 和彦* 原 和彦* 山浦 慎一*
Kazuhiko IWANAGA Kazuhiko HARA Shinichi YAMAURA

要 旨

MFPのコピー機能では、スキャナーで読み取った画像データに対して、種々の画像処理を行っている。画像処理を実現するために従来は、画像処理専用ASICを開発してきたが、リコーでは多製品への柔軟な対応、新しいアルゴリズムの迅速な投入を目的として、画像処理のミドルウェア化を推進している。

モノクロMFPにおいては、画像処理プロセッサRi10シリーズを用いることで画像処理のミドルウェア化を実現した。

カラーMFPにおいても、画像処理のミドルウェア化は有効手段であり、カラーMFPの画像処理を実現できる画像処理プロセッサの必要があるため、新しい画像処理プロセッサRi2001の開発を行った。

ABSTRACT

Regarding copy function of the MFP, it needs various image processing about the image data read with the scanner. In the past, Ricoh has been developed many ASIC for these image processing. And now, we are promoting the Middleware-ization strongly for flexible correspondence to many product and installing new algorithm swiftly. In fact, we realized the Middleware-ization of B&W MFP by using imaging processor of Ri10 series. And we are sure that the Middleware-ization is effective way for the color MFP also. So, the new imaging processor Ri2001 was developed for the color image processing of the color MFP.

* 電子デバイスカンパニー 画像LSI開発センター
Imaging System LSI Development Center, Electronic Devices Company

1. 概要

MFPのコピー機能では、スキャナーで読み取った画像データに対して、種々の画像処理を行っている。画像処理を実現するために従来は、画像処理専用ASICを開発してきたが、リコーでは多製品への柔軟な対応、新しいアルゴリズムの迅速な投入を目的として、画像処理のミドルウェア化を推進している。

モノクロMFPにおいては、画像処理プロセッサRi10シリーズを用いることで画像処理のミドルウェア化を実現した。

カラーMFPにおいても、画像処理のミドルウェア化は有効手段であり、カラーMFPの画像処理を実現できる画像処理プロセッサの必要があるため、新しい画像処理プロセッサRi2001の開発を行った。

Ri2001の中心となるプロセッサコアRi20は、画像処理に適したアーキテクチャであるSIMD (Single Instruction stream Multiple Data stream) 方式を採用した上で、画像処理に特化した機能を付加した。

SIMD方式は、高い並列性を有しているために通常の画像処理においては性能面で優れているが、いくつかの画像処理では、SIMD方式の並列性を活かさない。Ri2001ではこのような画像処理についても性能を落とさずに処理を行うことができるようにした。

このように、カラーMFPのミドルウェア化を推進するための画像処理プロセッサRi2001を開発したので、その内容を紹介する。

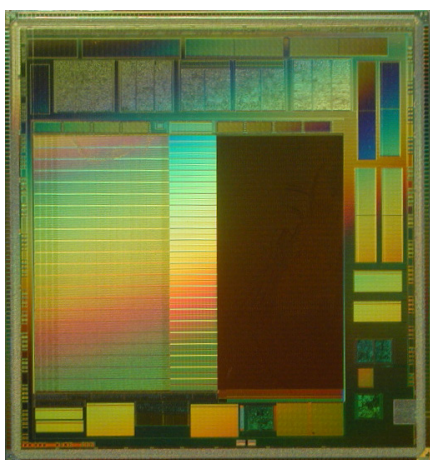


Fig.1 Die photo of image processor Ri2001.

2. LSIの特徴

今回開発したLSI (Ri2001) の仕様を以下に述べる。

(1) 主な特徴

- 352個のプロセッサエレメント (Processor Element, 以下PEと略する) を持つ専用の超並列SIMDエンジン “Ri20コア” を内蔵
- ピーク演算性能84.5GOPS (Giga Operation Per Second) , ピーク演算ビット幅5,632ビット
- 69種のSIMD命令と43種の制御命令により、リアルタイム画像処理をソフトウェアで実現
- 画像処理に必要な画像メモリを各PEに1Kバイト搭載、周辺機能を含め総計520Kバイトの容量
- 画像データの入出力に専用の画像ポートを8本搭載し、高速画像データ転送を実現
- 拡大・縮小, ミラー, LUT, ステート変換, 機能をサポートする専用ハードウェアを搭載
- 大容量画像, 高速画像処理には、画像ポート直結機能によるマルチプロセッサ構成で対応
- 専用のソフトウェア開発支援ツールにより、ソフトウェアの作成, リアルタイムデバッグを実現

(2) 主要な電気的特性

- 動作周波数：入力30MHz (内部240MHz)
- 電源電圧：コア電源1.275V, I/O電源3.3V
- 製造プロセス：0.13 μ m CMOSプロセス, 銅8層配線, 3200万トランジスタ
- パッケージ：336ピン・プラスチックBGA

2-1 Ri2001全体

画像プロセッサRi2001は、Ri20コアを中心に構成され、Ri20の周辺機能には画像データの転送・保持を行うイメージプロセッシング部と、Ri2001システムを制御するシステムコントロール部に、分類される。画像処理プロセッサとして特徴のイメージプロセッシング部は、画像ポート、ラインバッファ・コントローラ、ラインバッファメモリで構成される。

以下のFig.2にRi2001の構成を示している。

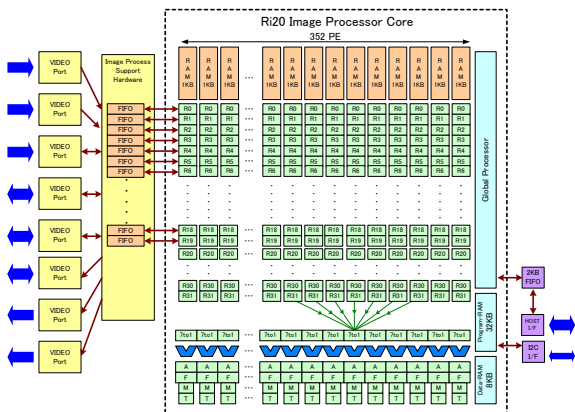


Fig.2 Block diagram of Ri2001.

2-2 Ri20コア

Ri2001の中心となるのはSIMD形式プロセッサであるRi20コアである。Ri20コアは画像プロセッサ全体の制御を行うための制御プロセッサとしてのSISD (Single Instruction stream Single Data stream) 型のプロセッサであるグローバルプロセッサ (Global Processor, 以下GPと略す) と、1つの命令で並列動作する352個のプロセッサエレメント (PE) とから構成されている。

各PEは8ビット32個の汎用レジスタを内蔵している。32個の汎用レジスタの中で24個はRi20コア外部のラインバッファとの間で独立して480Mバイト/秒の転送レートでデータ転送が可能である。さらに各PEにはラインバッファとして使用することを目的としている1Kバイトの画像データメモリ (PE-RAM) を内蔵している。

各PEは8ビットの条件レジスタを備え、ほとんどの命令は条件付き実行命令であるため、PE毎に異なる演算処理が可能である。

Ri20コアは、352個のPEを使用することで並列度の高い演算を行うことが可能な、基本的なSIMD命令を搭載しているが、さらに画像処理で頻出するクリッピング処理に対応するための飽和演算命令や、広範囲な領域での画像データのマスクなどに対応するために任意のPEを選択して条件レジスタに値を設定することが可能な命令などを搭載している。

2-3 画像ポート

画像ポートは、Ri2001内のラインバッファと、Ri2001外部との間で画像データのやり取りを行う、データ転送ポートで

ある。画像ポートは1チャンネルあたり16ビットデータの入出力を、最大で75MHzで転送可能であり、1.2Gビット/秒の転送レートを持つ。Ri2001では入力用の画像ポートを3チャンネル、出力用の画像ポートを2チャンネル、入出力兼用の画像ポートを3チャンネル搭載している。

また画像ポートを介して、複数のRi2001を接続することで、画像処理のボリュームに応じてマルチプロセッサ構成をとることが可能となっている。

2-4 ラインバッファ・コントローラ

Ri2001は、Ri20コアの汎用レジスタと、コア外部のラインバッファとの間で、Ri20コアの命令実行のバックグラウンドでデータ転送を行うためのラインバッファ・コントローラを搭載している。

ラインバッファ・コントローラは、転送の機能別にタイプA、タイプB、及びタイプC+の3種類のタイプが存在しているが、いずれもベースとなる機能として空間フィルタなどで必要となる1ライン前、2ライン前などの画素データをラインバッファに転送するためのFIFO機能を有している。

個別の転送機能としては、タイプAは画像処理における変倍処理 (拡大・縮小) に対応するための画素データの間欠書き込み・重複読み出し機能、およびミラーリング (鏡像) 処理のためのLIFO機能を有している。タイプBは画像ポートとラインバッファとの間のデータ転送、ラインバッファと汎用レジスタとの間のデータ転送の調停を行っている。タイプC+は、タイプAと同様の画像処理に対応するのに加えて、ラインバッファをLUTとして利用することを可能にしている。

また、ラインバッファは8K/9Kバイトのメモリを18本搭載しており、複数のメモリを組み合わせることで処理を行う画像データのサイズに応じてラインバッファの容量を最大で主走査画素数が49Kバイトまでフレキシブルに拡張可能である。

2-5 ホスト・インターフェース

システム全体をコントロールするホストマイコンから、画像処理を担当するRi2001を制御するためのホスト・インターフェースを備える。このインターフェースにより、ホストマイコンは、Ri2001へのプログラム転送や割込みなどの制御が可能になる。

インターフェースは、16ビットまたは8ビット幅のバス構

造で、最大50MHzで転送可能である。

ホストマイコンからRi2001へは16ビット1024ワード相当のFIFOバッファが、Ri2001からホストマイコンの方向には16ビット16ワードのFIFOバッファが備えられており、Ri2001が実行するプログラムをブートするほか、各種データの送受信に使用できる。

さらにホストマイコンからRi2001に対して、リセット/NMI/IRQの要求が可能となっている。

2-6 その他の機能ブロック

その他では、 μ Cバスインターフェース、汎用IOポート23本、インターバルタイマ4本、ICEとの接続を行うエミュレーションI/Fを搭載している。 μ Cバスに接続されたシリアルEEPROMからのブートも可能としている。

3. SIMDプロセッサ

3-1 SIMDプロセッサアーキテクチャ

一般にデジタル画像データ（画素）は8ビットあるいは16ビットで扱われており、現在PCで主流になっている32ビットや64ビットのデータを処理するCPUほどの演算精度は必要としない。一方で画像は多くの画素の集合体であることから、同一処理のデータ数そのものが多いという特徴がある。

画像処理に適したプロセッサという観点では、単純に単位時間に処理するデータ数量（画素数）で性能を評価できる。その高性能へのアプローチは1回の処理を時間的に圧縮する方法と空間的に分散する方法の2通りがある。前者は1回の処理に有する時間の短縮、すなわちクロック周波数の向上であり、後者は1回の処理で複数のデータを処理する並列処理である。単純な性能指標はこれらを積算したものと見なすことができる。

SIMDとは、並列処理のアプローチの一種であり、単一の命令により複数のデータが処理されるプロセッサアーキテクチャである。同一の処理を多くの画素に対して行う画像処理の特徴に、よく適しているため、画像処理向けといわれるプロセッサの多くは、基本的にSIMDアーキテクチャを採用している。

SIMD型のプロセッサで、ひとつのデータを処理するユ

ニットをPEと呼び、Ri20コアではPEを352個搭載している。これにより主走査方向に連続した352個の画素データを同時に演算処理可能である。

3-2 画素データの転送

SIMDプロセッサでは、演算処理は並列性を活かして効率が向上するが、各PEへ入力画像データを送りこむこと、および処理結果の画像データを各PEから取り出すことは、並列性では解決しない。

各PEへの画像データを取り込む場合に通常のプロセッサでは、メモリからのデータロード命令が352回必要となり、この命令処理にかかる時間とメモリへのアクセス時間が性能上の大きなネックとなる。Ri2001では前述のRi20コアと接続されたラインバッファ・コントローラが、Ri20コアの汎用レジスタに直接アクセスすることが可能であるため、画像データ転送をRi20コアの演算処理と同時に実行でき、画素データの各PEへの転送中でも、Ri20コアの画像処理能力は最大限発揮される。

3-3 空間フィルタ

空間フィルタでは、近接画素を参照して対象画素が算出される。Fig.3に示す 7×3 フィルタを例にすると、主走査方向に前後それぞれ1つ隣/2つ隣/3つ隣の画素、副走査方向に前後1つ隣の画素を参照し演算する。主走査方向はPEの並びに、副走査方向はレジスタの並びに沿って画素データが配置される。副走査方向の参照は同一PE内でのレジスタなので特別なことはないが、主走査方向の参照は別のPE内のレジスタへのアクセスとなり、適した仕様が必要になる。Ri20コアではレジスタアクセス時に、左右それぞれに1つ隣/2つ隣/3つ隣のPEのレジスタに対しアクセス可能で、主走査方向の隣接画素参照にも対応している。

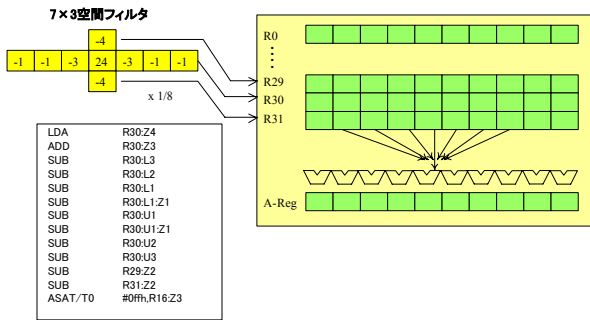


Fig.3 Example of image filter by using 7×3 matrices.

プログラム例の3～10番目の命令で隣接PEのレジスタを参照している。

また、上記の例にもあるが、空間フィルタでは画素データの整数倍の値を多用するが、Ri20はレジスタアクセスの際にデータをビットシフトする機能を持つため、2倍、4倍、8倍、あるいは1/2倍、1/4倍、1/8倍などが簡単に実現できる。上記の例では、1番目の命令で4ビットシフトの読み出しで16倍を作り、次の命令の読み出しで3ビットシフトによる8倍を作り、加算して合計24倍を得ている。

4. 各種画像処理対応

4-1 大規模画像とSIMD

SIMDプロセッサにおける画像処理は、並列処理の特性にマッチしているが、実際にSIMDプロセッサで画像処理を効率的に実施するには、いくつかの課題を解決しなければならなかった。

Ri20コアで一度に処理できる画素数は、最大でPE数と同じ352画素であるが、MFPで扱う画像データは、例えばA4用紙の縦方向600DPIで約7500画素であるため、352画素単位の処理を繰り返し実行することで実現している。（以下、これをSIMD単位と呼ぶ） Fig.4は、対象画像をSIMD単位での分割するイメージ図である。ここでのSIMD単位は“320”であるが、その理由は後述する。

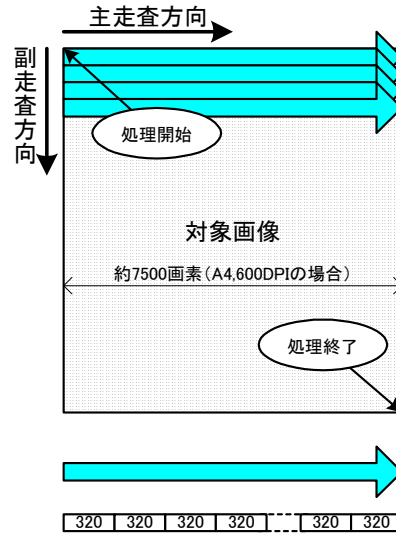


Fig.4 Processing of large scale image data.

空間フィルタ処理では、隣接する画素を参照して結果を算出するため、両端付近に位置するPEでは隣接画素を参照できず、正しくない処理結果（無効画素）が出る。これは参照を繰り返すごとに無効画素が増える。仮に16画素が無効画素となった場合、一度の処理で生成される正しい結果（有効画素）は、352-16*2で320画素である。7500画素のラインを処理するためには、7500/320で24回の繰り返しとなる。

また、ラインバッファからPEへ画像データを転送する際、無効画素の発生を考慮する必要がある。これはラインバッファとPEのレジスタ間のデータの転送を実行するラインバッファ・コントローラに、無効データ分をオーバーラップして転送管理する機能を持たせる事で解決した。

Fig.5にラインバッファからPEレジスタへのデータ転送時のイメージを示す。

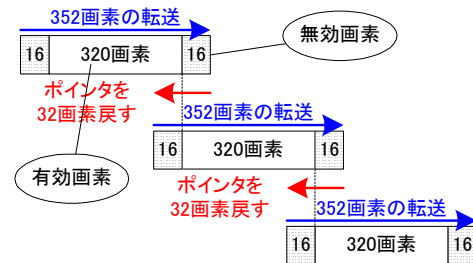


Fig.5 Forwarding of image data.

Ri20コアでは、PE-RAMを備えているため、主走査方向の画素データ（ライン）をその内部に取り込むことができる。

取り込まれたラインは、SIMD単位で分割された状態となり、ラインはPE-RAMのアドレスで管理される。PE-RAMに対するアドレッシングでは、複数アドレスで1ラインとなる配置で便利のように、アクセスポインタが次のラインの配置されているアドレスへスキップしたり、最後の処理ラインが終わると自動的に最初のラインの配置しているアドレスへ戻る機能を備え、専用のハードウェアポインタを16組用意した。これによりPE-RAMに格納された画像データの読み出し・格納の効率が向上する。

4-2 非線形処理

MFPでは、スキャナーの読み取り特性や、プロッタの書き込み特性によって、出力された画像の階調が元の原稿の画像の階調とは異なるものになることを補正するために γ 補正といわれる非線形の画像処理を行っている。 γ 補正は、以下のFig.6のように、変換後のデータを、変換前のデータの関数で表した場合に、2次式や根号を含むような式で表される非線形な処理であり、通常はLUT（ルック・アップ・テーブル）を使用してテーブル変換を行うことで γ 補正を行う。

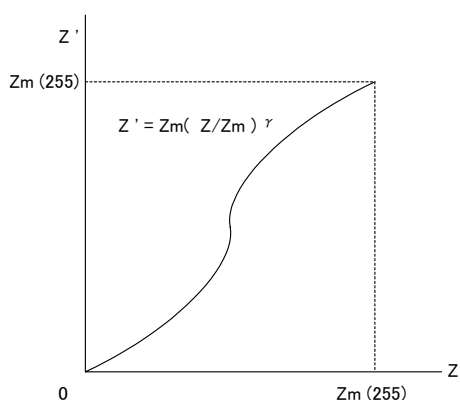


Fig.6 Image curve of gamma correction.

通常のSISD型（Single Instruction stream Single Data stream）のプロセッサでは、 γ 補正後のテーブルデータをデータメモリに格納しておき、補正前の画素データをデータメモリのアドレスとして、データメモリから補正後のデータを読み出すことで γ 補正を実現する。

SIMD型のプロセッサでも1画素ずつ逐次で、データメモリにアクセスすることをPEの数だけ、すなわち352回繰り返せば、非線形処理を行うことは可能であるが、それでは折角の

並列性が全く活かせないことになる。

SIMD型プロセッサの持つ並列性を発揮するためには、各PEが全てデータメモリを備え、それぞれ独立して自身の画素値をアドレスとしてメモリにアクセスするように構成すれば良いのであるが、 γ 補正などのテーブルは全く同一の内容であるため、非常に無駄が多い。

そこでRi20コアでは両者の間を取って、1つの命令で16バイトのデータに一度にアクセスできる構成のデータメモリを搭載し、その命令を16回だけ繰り返すことでテーブル変換を行うことが可能なテーブル変換用の命令セットを搭載している。

テーブル変換を行う場合には、あらかじめ変換前の画素データの値をデコードして条件レジスタを設定する命令を行っておく。そして、実際にテーブル変換を行う命令ではデータメモリから16バイトのデータが全PEに対してブロードキャストされる。各PEは自身の条件レジスタの値に応じてデータを選択してロードを行う。

これを以下のFig.7に記載のように16回繰り返すことによってテーブル変換が完了する。

```

1  LDI      #table.head, G2 ; テーブルデータの格納先頭番地をG2に設定
2  TSET    R0, [G2]+      ; R0の値(変換前データをデコードしてT.Mレジスタ設定、Dへのロード)
3  LD      [G2]+, D      ; Dへのロード
4  TLD/F7/M0 [G2]+      ; if ( T7 = 0 && M0 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
5  TLD/F7/M1 [G2]+      ; if ( T7 = 0 && M1 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
6  TLD/F7/M2 [G2]+      ; if ( T7 = 0 && M2 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
7  TLD/F7/M3 [G2]+      ; if ( T7 = 0 && M3 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
8  TLD/F7/M4 [G2]+      ; if ( T7 = 0 && M4 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
9  TLD/F7/M5 [G2]+      ; if ( T7 = 0 && M5 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
10 TLD/F7/M6 [G2]+      ; if ( T7 = 0 && M6 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
11 TLD/F7/M7 [G2]+      ; if ( T7 = 0 && M7 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
12 TLD/T7/M0 [G2]+      ; if ( T7 = 1 && M0 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
13 TLD/T7/M1 [G2]+      ; if ( T7 = 1 && M1 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
14 TLD/T7/M2 [G2]+      ; if ( T7 = 1 && M2 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
15 TLD/T7/M3 [G2]+      ; if ( T7 = 1 && M3 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
16 TLD/T7/M4 [G2]+      ; if ( T7 = 1 && M4 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
17 TLD/T7/M5 [G2]+      ; if ( T7 = 1 && M5 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
18 TLD/T7/M6 [G2]+      ; if ( T7 = 1 && M6 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
19 TLD/T7/M7 [G2]+      ; if ( T7 = 1 && M7 = 1 ) then D[ T6~T3 ] -> AH, Dへのロード
20 STA     R0, Z8        ; AHのデータ(変換後データ)をR0にストア

```

Fig.7 Sample code of gamma correction by using LUT.

こうしてRi20コアでは、256階調（8ビットデータ）の γ 補正を、20サイクルという短いサイクルで行うことが可能となっている。

さらにRi20コアでは、テーブルデータの容量を低減するために差分データでのテーブル変換にも対応できる命令セットを用意している。

4-3 変倍（拡大・縮小）

MFPでは、画像データの拡大や縮小といった画像処理を行う必要がある。こうした変倍処理においては、処理を行う

前の画像データを主走査方向、および副走査方向の両方向に対して、拡大や縮小を行う。

拡大を行う場合には、画素データを重複させて画素数を増やすことを行う。例えば200%に拡大する場合には、元の画素データを2つずつ並べている。逆に縮小を行う場合には、画素データを間引いて画素数を減らすことを行う。例えば50%に縮小する場合には、元の画素データの2つのうち1つを間引いている。この画素数を増やしたり、減らしたりする操作のことを速度変換と呼んでいる。

速度変換だけでは、画素の値が滑らかに変化しないために、拡大時は、速度変換の後でバイキュービック法による補間演算を行っている。逆に縮小時は速度変換の前に補間演算を行っている。

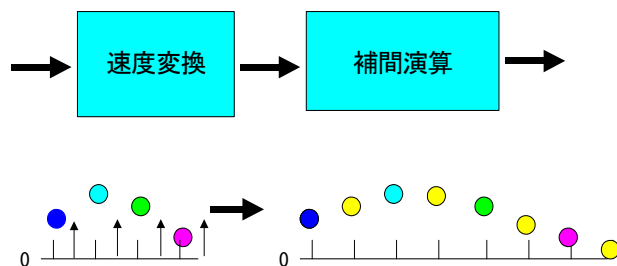


Fig.8 Workflow of magnification processing.

SIMDプロセッサを用いて変倍を行うことを考えると、補間演算に関しては、各PEに補間用の係数の転送を行えば並列処理を行うことが可能であるため、SIMDに適した処理であると言える。

速度変換に関しても、副走査方向の速度変換に関しては、SIMDプロセッサに供給する1ラインのデータを間引いたり、重複させたりすることで対応が可能である。

しかし主走査方向の速度変換は、SIMDプロセッサの各PEで扱う画素の数を変更する必要がある、これは逐次で処理するしかないため、並列性の利点を活かさない。

そこでRi2001では、主走査方向の速度変換を実現するのに、Ri20コア外部のラインバッファ・コントローラに速度変換機能を持たせることで、通常のラインバッファの転送のように、Ri20コアで行う画像処理のバックグラウンドで、速度変換を行うようにした。

以下のFig.9は2つのラインバッファ・コントローラを用いて、変倍処理を行うことが出来るようにした時の構成を示し

ている。

2つのラインバッファ・コントローラをマスター・スレーブとして使用し、マスター側のラインバッファ・コントローラが、速度変換のための1ビットの変倍制御データと、補間演算のための補間データをバックしたデータの転送を行う。

スレーブ側のラインバッファ・コントローラは画素データのデータ転送を行うが、マスター側のラインバッファ・コントローラの変倍制御データが“1”の場合には、間引き（縮小の場合）、重複（拡大の場合）を行い、“0”の場合には、画素データをそのまま転送するという具合に、連動して動作するようにしている。

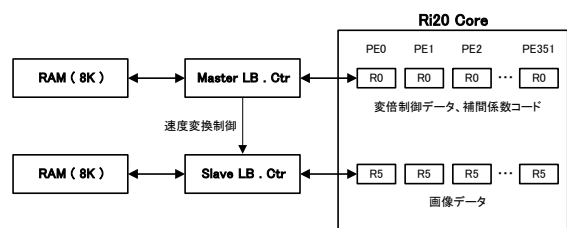


Fig.9 Construction of buffered memory in line in case of magnification processing.

5. Ri2001の処理能力

Ri2001を用いて画像処理を行った場合の性能は以下のとおり。

画像処理能力(モノクロ8ビット画像を対象)	
シェーディング補正	0.13ns / 画素
γ補正(8ビット→8ビットのLUT)	0.27ns / 画素
7行×7列の平滑フィルタ(係数は固定値)	0.66ns / 画素
3行×3列のラプラシアンフィルタ(係数は固定値)	0.16ns / 画素
399%のバイキュービック法による拡大(主走査方向)	0.66ns / 画素
ディザ法2値化	0.05ns / 画素
マスク(主走査2領域)	0.27ns / 画素

数値は@240MHz

Fig.10 Processing speed of Ri2001.

6. 現状と今後の予定

画像処理プロセッサRi2001の現状の開発ステータス及び今後の予定は以下の通りである。

- ・画像処理LSI Ri2001
- ES : 出荷済み
- 量産 : 量産製品RB5V870 2005 Q2量産予定
- ・ミドルウェア開発環境

クロスアセンブラ, シミュレータ, ICEデバッガ, 評価
ボードは全て開発完了.

以下のFig.11はRi2001を実装した評価ボードである.



Fig.11 Evaluation board of Ri2001.

7. まとめ

画像処理プロセッサRi2001を開発し, モノクロMFPで実績のあるミドルウェアによるMFP開発手法は, カラーMFPでも実現可能となった. 今後は, 90nmプロセス採用とそれによる高性能化を検討している.

注1) FCはフィリップス社の登録商標です.