

CAMによる高速パターンマッチング機能を有する 超並列SIMDプロセッサ

幸野 豊[†] 熊木 武志[†] 石崎 雅勝[†] 田上 正治[†] 小出 哲士[†]

マタウシュ ハンス ユルゲン[†]

行天 隆幸^{††} 野田 英行^{††} 黒田 泰斗^{††} 堂阪 勝己^{††} 有本 和民^{††} 齊藤 和則^{††}

[†] 広島大学 ナノデバイス・システム研究センター 〒739-8527 東広島市鏡山 1-4-2

^{††} 株式会社 ルネサステクノロジ 〒664-0005 兵庫県伊丹市瑞原 4-1

E-mail: †mpe_list@sxsys.hiroshima-u.ac.jp

あらまし 近年の大量のマルチメディアデータを効率よく処理するために、データの処理方向と演算器の配置を工夫することにより、大量のデータに対し同一の演算を施すマルチメディア処理を高速に処理することができる超並列SIMDプロセッサをこれまでに開発している。本報告では、超並列SIMDプロセッサの入出力インターフェースであるTSブリッジにCAMを加え、パターンマッチング処理を高速化する方法を提案する。提案アーキテクチャにより、パターンマッチングの一種であるハフマン符号化において、従来のDSPに比べ約60%の処理クロックサイクル数の削減を実現することができた。

キーワード 超並列SIMDプロセッサ, CAM, ハフマン符号化, マルチメディア処理, パターンマッチング

Super Parallel SIMD Processor with CAM based High-Speed Pattern Matching Capability

Yutaka KONO[†], Takeshi KUMAKI[†], Masakatsu ISHIZAKI[†], Masaharu TAGAMI[†], Tetsushi KOIDE[†], Hans Jürgen MATTAUSCH[†], Takayuki GYOHTEN^{††}, Hideyuki NODA^{††}, Yasuto

KURODA^{††}, Katsumi DOSAKA^{††}, Kazutami ARIMOTO^{††}, and Kazunori SAITO^{††}

[†] Research Center for Nanodevices and Systems, Hiroshima University 1-4-2, Kagamiyama, Higashi-Hiroshima, 739-8527, Japan

^{††} RENESAS Technology Corporation 4-1, Mizuhara, Itami City, Hyogo, 664-0005, Japan

E-mail: †mpe_list@sxsys.hiroshima-u.ac.jp

Abstract A super parallel SIMD processor has been developed for handling the increasing amount of multimedia data efficiently. It is able to execute the same operation with a large amount of multimedia data at high speed by changing the conventional word-serial processing to a highly parallel bit-serial processing with a large number of simple computing units. In this paper, we improve the time space conversion bus bridge (TS-bridge), which is the external interface of the super parallel SIMD processor, with a pattern matching capability for table look-up operations. This reduces the number of Huffman coding clock cycles by 60% as compared with DSP because the necessary can be carried out in table look-up clock cycles.

Key words super parallel SIMD processor, CAM, Huffman encoding, multimedia processing, pattern matching

1. はじめに

近年、半導体技術、通信技術の発展、及び通信インフラの整

備により、携帯電話やデジタルカメラなどのモバイル情報機器や家電製品では、高画質の静止画や動画像ならびに高音質の音楽等のマルチメディアデータを取り扱う機会が増えてきて

いる。このようなマルチメディアデータ処理においては、一般的にデジタル信号処理に特化した Digital Signal Processor (DSP) とソフトウェアの組み合わせにより対応されている。

DSP による処理では、ソフトウェアの変更により様々なアプリケーションを取り扱うことができるため、規格や仕様の変更などに対応し易く、汎用性も高い。しかしながら、汎用性を持たせることにより、回路が複雑になり、実装面積が増大するという欠点があった。また最近では、ハイビジョンや地上デジタル放送の普及により画像や音声の高品質化が求められており、一定時間内に処理するデータ量が増大している。このため、従来の DSP とソフトウェアの組み合わせでリアルタイム処理を行うためには DSP コアの動作周波数を GHz 領域まで引き上げ、動作させる必要性があり、この結果消費電力が増大し、モバイル情報機器にて許容される消費電力の制約を超えてしまうため、別のアプローチによる解決策が必要となってきた。

そこで、我々はマルチメディアデータを高速かつ低消費電力で処理することができる超並列 SIMD プロセッサを開発している [1], [2], [3]。この超並列 SIMD プロセッサでは、従来のプロセッサがワードシリアル・ビットパラレルでデータ処理をするのに対して、ビットシリアル・ワードパラレルでデータを処理する。そのため、大量のデータに対して同一の演算を行うようなマルチメディアアプリケーションでは並列度が 2048 であるため、高速に処理を行うことができる。しかしながら、パターンマッチングやテーブル変換のようなデータ処理をワードごと、逐次的に行う処理に関しては、SIMD プロセッサの稼働率が落ちてしまい高速に処理することが難しくなる。そこで、我々は超並列 SIMD プロセッサに Content Addressable Memory (CAM) を付加し、一致検索処理を高速化することで、パターンマッチングを高速に行う方法を提案する。改良した超並列 SIMD プロセッサにより、従来のマルチメディア処理プロセッサである DSP と比較してハフマン符号化処理にかかるクロックサイクル数を大幅に削減することが可能となった。

以降では、第 2 節で開発している超並列 SIMD プロセッサについて述べ、第 3 節で CAM を用いたパターンマッチングの高速化について説明する。シミュレーションによる評価を第 4 節で述べ、最後に第 5 節でまとめと今後の課題について述べる。

2. 超並列 SIMD プロセッサとパターンマッチング

2.1 超並列 SIMD プロセッサの構成

大量のマルチメディアデータの効率的な処理のために開発した超並列 SIMD プロセッサは [1], [2], [3]、携帯電話に代表されるモバイル情報機器に搭載することを目的としており、動作周波数 200MHz で消費電力が 250mW、処理性能が 40GOPS と従来のプロセッサよりはるかに優れている [1]。超並列 SIMD プロセッサのブロック図を図 1 に示す。超並列 SIMD プロセッサは、演算処理を行う SIMD プロセッサコア、SIMD プロセッサコアを制御するコントローラ、ホスト CPU から送られてきたデータを SIMD プロセッサコアに転送する TS ブリッジで構成されている。

SIMD プロセッサコアは、2,048 個の演算器 (以下、PE) と 1M ビット (左、右共に 256 ビット×2,048 エントリ) の SRAM で構成されている。各々の PE は、左右 256 ビットずつの SRAM に挟まれており、SRAM - PE 間のデータ転送を行う Horizontal Channel (H-ch) と各 PE 間のデータ転送を行う Vertical Channel (V-ch) によって密接に結合している。SRAM は PE の演算用レジスタであり、データの入出力や中間データを保存する RAM として動作する。以上の構成を採ることによって SRAM-PE 間のデータ転送速度を保ったままで超並列を実現し、メモリ内に PE を埋め込むという構造により小面積を実現している [1]。

TS ブリッジは 2 バンク直交 SRAM2 個、SIMD プロセッサコアと TS ブリッジを結ぶ Global Channel (G-ch)、及び MUX/DMUX で構成される [2]。従来のプロセッサはワードシリアル・ビットパラレル処理でデータ処理を行うのに対して、超並列 SIMD プロセッサはワードパラレル・ビットシリアル処理でデータを処理する。ホスト CPU から送られてきたデータを超並列 SIMD プロセッサで処理するにはデータをワードシリアルからビットシリアルへ変換する必要がある。TS ブリッジは、CPU から処理データを受け取り、SIMD プロセッサコアで処理ができるよう、ワードシリアル/ビットシリアルデータ変換処理を直交 SRAM にて行い、G-ch を介して SIMD プロセッサコアにデータを転送する。転送されたデータは SIMD プロセッサコアの SRAM に保存されコントローラからの信号により読み出され、PE で演算処理が行われる。SIMD プロセッサコアのコントローラは、PE の制御信号、SRAM のアドレス、及び制御信号を SIMD プロセッサコアへ出力する。図 2 は SIMD プロセッサにおける処理の概念図である。まず、CPU バスからデータが TS ブリッジに送られ、TS ブリッジ内の直交 SRAM にワードシリアルで書き込まれる。データが一杯になると、TS ブリッジから超並列 SIMD プロセッサ内の SRAM に、ビットシリアルで書き込まれる。図 1 に示しているように TS ブリッジ内の直交 SRAM は、2 バンク SRAM で書き込み用と読み込み用の 2 種類ある。一方の書き込み用メモリバンクにデータが一杯に書き込まれると、もう一方のメモリバンクに同様の書き込み動作が行われる。1 つのメモリバンクは 32 ビット×32 ワードであるため、CPU バスから一方のメモリバンクへの書き込みと、もう一方のメモリバンクから SIMD プロセッサコアへの転送は、同じ 32 クロックサイクルで交互にインタリーブ動作で実現することができる。SIMD プロセッサコアの SRAM に格納されたデータは、コントローラからの演算命令により、各行の PE にビットシリアルで読み出され、演算結果は再び各行の SRAM に格納される。

2.2 超並列 SIMD プロセッサによるハフマン符号化

前述したように従来のプロセッサはワードシリアル・ビットパラレルで、超並列 SIMD プロセッサはビットシリアル・ワードパラレルでデータ処理を行う。ワード数が多く、一般的に 8~16 ビット長程度で処理することの多いマルチメディアデータにおいては、ワードシリアルでデータ処理を行う従来のプロセッサより、ビットシリアルでデータ処理を行う超並列 SIMD

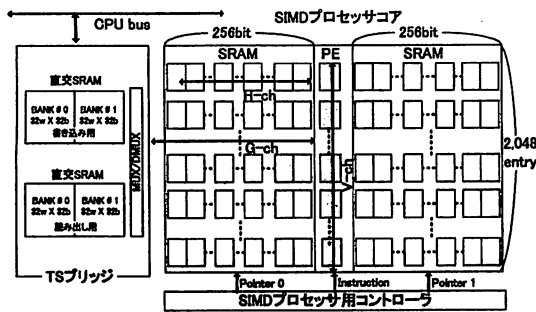


図1 超並列 SIMD プロセッサ。

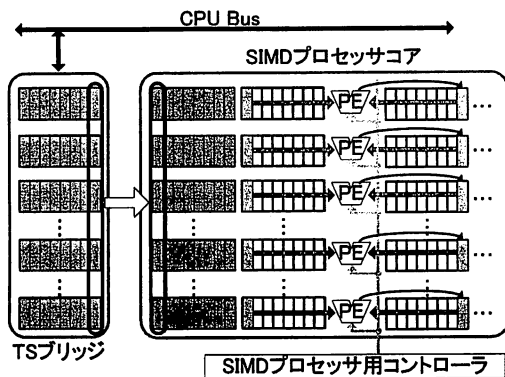


図2 超並列 SIMD プロセッサのデータ処理。

プロセッサの方が高速に処理できることが多い。しかしながら、マルチメディア処理の中には SIMD アーキテクチャに適していない処理が存在する。画像圧縮処理の一つである JPEG 処理中に用いられているハフマン符号化が例として挙げられる [4]。

ハフマン符号化とはデータ圧縮処理の一つであり、出現率の高いデータに短い符号を割り当てていく圧縮方法である。通常、ハフマン符号化を行う場合は、圧縮前データと圧縮後データが 1 対 1 で対応しているハフマン符号化テーブル表によるパターンマッチング処理によって行われる。超並列 SIMD プロセッサにおいてハフマン符号化を行う場合、図 3 に示すように圧縮前データを 1 つずつ SIMD プロセッサコアに格納し、1 ビットずつハフマン符号化テーブルと比較していかねばならず、高速に処理することが難しくなる。超並列 SIMD プロセッサではビットシリアル処理を 2048 並列で行うことで、全体の処理能力を大幅に向上させているが、更に処理能力を向上させるためには、ハフマン符号化のようなデータの逐次的処理を行うアルゴリズムを効率よく処理を行う工夫が必要となる。

2.3 既存のハフマン符号化アーキテクチャ

ハフマン符号化は、もっとも出現率の高いデータに少ないビットの符号を割り当てるため、常に最適な符号を構成できるという特長があり、マイクロプロセッサベース、SRAM ベース [5]、Programmable Logic Array (PLA) ベース [6]、CAM ベース [7] など多くのハードウェアやソフトウェアにて実装されている。

マイクロプロセッサベースのアーキテクチャはソフトウェアで符号化が実現されるため、汎用性は高いがハードウェアと比較して処理効率は低い。また、SRAM ベースのアーキテクチャでは符号化する際、符号表から変換データを降順で検索するためアドレスが大きくなるにつれてクロックサイクル数が増大する。PLA ベースアーキテクチャではハフマン符号化テーブルが AND や OR の組み合わせ回路として構成されているため、テーブルの変更が難しく汎用性も低い。CAM は内部にコンテンツテーブルと呼ばれるメモリを持ち、そこにデータを格納している。これらは比較対象データとなり、比較データが入力されたならば、CAM は全比較対象データの中が 1 クロックサイクルで一一致検索し、一致しているものがあれば一致信号とそのデータアドレスを出力する [8]。CAM の一一致検索処理は並列に行われるため、他のハードウェアやソフトウェアに比べ高速に検索結果を得ることができる。CAM を利用したテーブル変換方法のブロック図を図 4 に示す。初めに、CAM にテーブル変換前のデータを、RAM にテーブル変換後のデータを格納しておく。テーブル変換前のデータとテーブル変換後のデータは各々のメモリのアドレスにて 1 対 1 で対応している。CAM に比較データが入力されると、CAM は格納してあるテーブル変換前データとビットパラレルで一一致検索処理を行う。一致したデータがあれば、そのデータのアドレスが RAM に入力され、格納されてあるデータを RAM にて出力すればテーブル変換が完了する。このように、CAM ベースのアーキテクチャは高速にテーブル変換を行うことができる。

代表的な画像圧縮方法である JPEG ではハフマン符号化が使用されている。図 5 に JPEG 処理のフローを示す。JPEG 処理ではハフマン符号化処理のほかに、離散コサイン変換 (DCT)、量子化、ジグザグスキャン、ランレングス処理などの処理から構成されている。JPEG の各処理は DCT、量子化のように全てのデータに対し同一の演算処理を行うものと、ハフマン符号化のようにデータを逐次的に処理するものの 2 つに大別できる。SIMD 演算処理は DCT や量子化は高速に処理できるため、ハフマン符号化を高速に行うことができれば JPEG 処理の更なる高速化が図れる。図 5 からわかるように、ハフマン符号化は JPEG 処理の最後の処理であるため、ハフマン符号化前までの処理を SIMD プロセッサコアで行い、ハフマン符号化を TS ブリッジで行い、CPU バスにデータを出力すればバイライン処理が可能となり、性能劣化を最小限に抑えることができる。

上記の理由から、本研究では超並列 SIMD プロセッサのインターフェース部である TS ブリッジに着目し、CAM を付加した TS ブリッジ (以下、パターンマッチング TS ブリッジ) においてハフマン符号化を高速に行う方法を提案する。

3. CAM による高速パターンマッチング

3.1 CAM を用いた TS ブリッジ

本研究では超並列 SIMD プロセッサでのハフマン符号化の高速化を実現するために、機能メモリの一つである CAM を TS ブリッジに付加する。図 6 にパターンマッチング TS ブリッジのブロック図を示す。パターンマッチング TS ブリッジは、TS

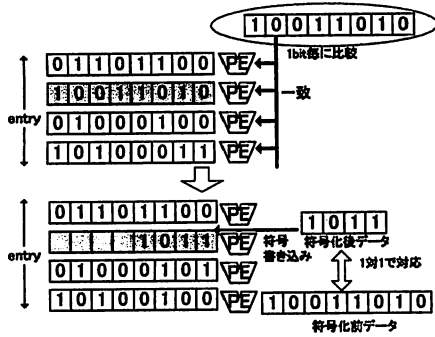


図3 SIMDプロセッサによるハフマン符号化例。

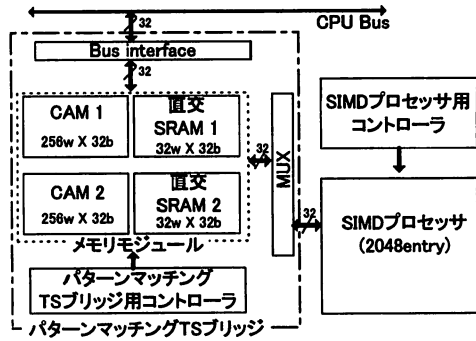


図6 パターンマッチングTSブリッジのブロック図。

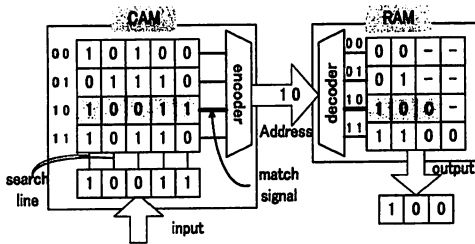


図4 CAMを使用したテーブル変換。

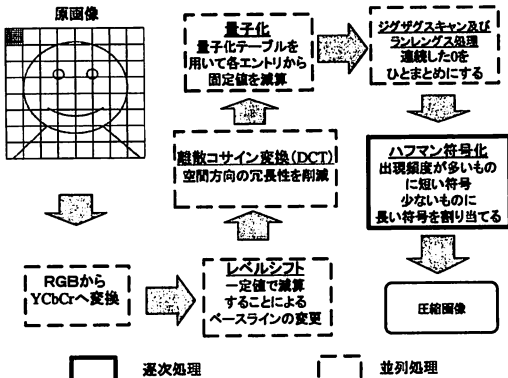


図5 JPEG処理フロー。

ブリッジ (直交 SRAM2 つで構成) に CAM を 2 つ付加した構成を採っている。2 つの CAM は図 4 で表しているように、1 つを CAM、もう一方を RAM として使用することにより高速なテーブル変換を実現する。またこれらの CAM は、CPU 側から 32 ビット × 512 ワードの 1 つの CAM として動作させる事が可能であり、一致検索の用途にも使うことができる。

3.2 パターンマッチング TS ブリッジによるハフマン符号化

本節ではパターンマッチング TS ブリッジによるハフマン符号化の概要について説明する。ハフマン符号化の前処理として、CAM1 にはハフマン符号化前データ、CAM2 にはハフマン符号化後データを格納しておく。また、SIMD プロセッサコアに

てハフマン符号化前まで処理されたデータは、直交 SRAM1 にビットシリアルで読み出されているものとする。図 7 にパターンマッチング TS ブリッジによるハフマン符号化処理を示す。パターンマッチング TS ブリッジは 2 つの直交 SRAM と 2 つの CAM で構成されており、図 7 の矢印及び番号はハフマン符号化におけるデータ処理の順番を示す。

[TS ブリッジによるハフマン符号化処理]

- (1) パターンマッチング TS ブリッジ用コントローラより直交 SRAM1 に読み出しデータのアドレスが入力される。
- (2) ビットシリアルで直交 SRAM1 に格納されたデータを、ワードシリアルで比較レジスタに読み出す。ここでデータはビットシリアルからワードシリアルに変換されている。
- (3) 直交 SRAM1 から読み出されたデータは CAM1 の比較データレジスタに入力される。これと同時に、SIMD プロセッサコアからハフマン符号化前データが、ビットシリアルで直交 SRAM2 に入力される。
- (4) CAM1 のコンテンツテーブルに格納されてあるハフマン符号化の全データパターンと、比較データレジスタに入力されたデータの一致検索処理を行い、一致したデータのアドレスを出力する。
- (5) CAM2 のアドレスレジスタに CAM1 から出力されたアドレスデータを入力する。
- (6) 入力したアドレスデータに対応したハフマン符号化後データを出力する。

直交 SRAM を 2 つ実装することにより、上記の (3) に示している直交 SRAM に SIMD プロセッサコアからデータを読み出すことと、ワードシリアルからビットシリアルへ変換したデータを CAM の比較データレジスタに格納することを同時に行うことができ、インターリーブ動作を実現している。

SIMD プロセッサコアは、ハフマン符号化前までの処理を全て終了してから、パターンマッチング TS ブリッジにデータを出力している。そのため、パターンマッチング TS ブリッジでハフマン符号化を行っている場合は、SIMD プロセッサコアはデータを送っているのみで処理は行われていない。また、上記の各処理をパイプラインステージとすることにより、ハフマン

符号化の高速化を図っている。

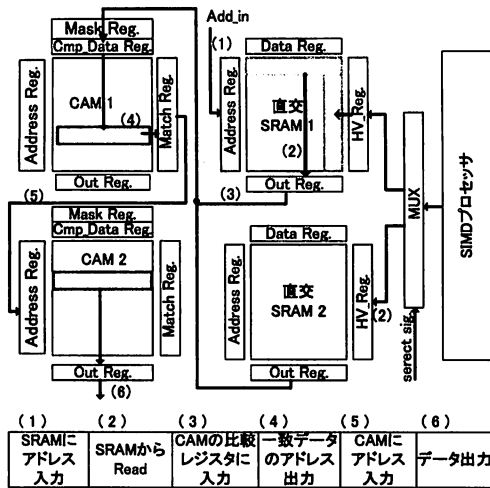


図 7 パターンマッチング TS ブリッジによるハフマン符号化ブロック図。

4. 性能評価

パターンマッチング TS ブリッジを FPGA に実装動作検証し、ハフマン符号化処理に必要なクロックサイクル数をシミュレーションにて算出した。実装動作検証には Xilinx 社 VirtexII (xc2v6000) の FPGA を使用した。シミュレーションには ModelSimSE 5.8c を用いた。シミュレーションに使用した画像の一部を図 8 に示す。パターンマッチング TS ブリッジによるハフマン符号化の処理クロックサイクル数は画像サイズに比例するため、それぞれの画像に対してハフマン符号化を行い、クロックサイクル数を算出し、平均値をパターンマッチング TS ブリッジのハフマン符号化の処理クロックサイクル数とした。

図 9 は JPEG 処理とハフマン符号化のサイクル数を比較したグラフである。比較対象として、16 ビット DSP、超並列 SIMD プロセッサ、の 2 つを用いる。

まず、JPEG 処理全体にかかるクロックサイクル数を見てみ

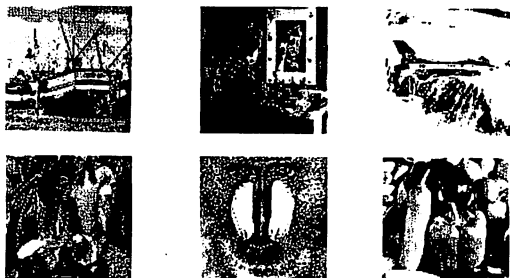


図 8 シミュレーションに使用した画像 (512×512 画素)。

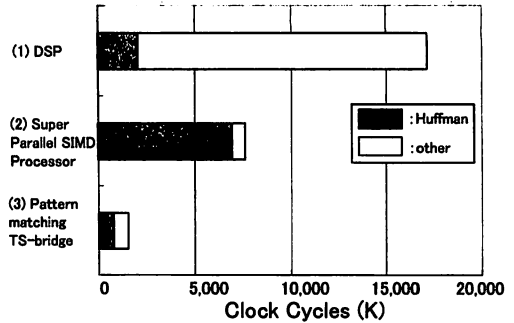


図 9 各プロセッサにおける JPEG 処理とハフマン符号化のクロックサイクル数の比較。

ると、超並列 SIMD プロセッサは DSP に比べ約 55% 削減している。しかしハフマン符号化だけに限るとクロックサイクル数が 3.4 倍に増加している。これはハフマン符号化が、データを逐次的に処理していくテーブル変換のため、SIMD 処理に適していないからである。そこでハフマン符号化を TS ブリッジで処理した場合、符号化にかかる処理クロックサイクル数は 813,312 となり、SIMD プロセッサのみに比べ約 88%、DSP に比べ約 60% 処理クロックサイクル数を削減することができた。これより、超並列 SIMD プロセッサと TS ブリッジを組み合わせると、JPEG 処理にかかるクロックサイクル数は、DSP に比べ約 90% 削減することができる。この結果より、超並列 SIMD プロセッサとパターンマッチング TS ブリッジの組み合わせは、従来の DSP に比べマルチメディア処理に適していることがわかる。

パターンマッチング TS ブリッジを Verilog-HDL にて設計し、論理合成を行った結果を表 1 に示す。使用したテクノロジーは 90nm CMOS、7 層配線、Synopsys Design Compiler^(注1)により論理合成を行った。表より 2 つの直交 SRAM と 2 つの CAM を組み合わせた直交メモリモジュールが 0.535mm²、メモリモジュールを制御しパターンマッチングを行うためのコントローラが 0.210mm²、パターンマッチング TS ブリッジの総面積が 0.749mm² となった。

表 1 パターンマッチング TS ブリッジの論理合成結果。

使用テクノロジー	90nm CMOS 7LM
TSブリッジ全体	0.749mm ²
TSブリッジコントローラ	0.210mm ²
CAMと直交SRAM	0.535mm ²
動作周波数 (シミュレーション)	200MHz

CAM を付加しない TS ブリッジの面積はフルカスタムで設

(注 1)：東京大学大規模集積システム設計教育研究センターを通し、シノプス株式会社の協力で行われたものである。

計を行った場合、 0.32mm^2 [2] となる。これより、パターンマッチング TS ブリッジの面積は論理合成の場合、 0.749mm^2 で TS ブリッジの約 2.34 倍である。表 1 は論理合成の結果であるため、直交 SRAM と CAM のメモリセルはフリップフロップで構成されている。従ってメモリセルをフルカスタムで設計することにより、面積を約 1/10 程度で実現することが可能である。これより、直交メモリモジュールの面積を 0.176mm^2 、パターンマッチング TS ブリッジの面積を 0.39mm^2 と見積もることができ、TS ブリッジの約 22% の面積増加で実現できる。CAM を付加することにより消費電力が増大することが予想されるが、パターンマッチング TS ブリッジで処理を行っている間は SIMD プロセッサコアはデータ転送のみを行っており、SIMD プロセッサコアで処理を行っている間は、パターンマッチング TS ブリッジは一致検索処理を停止しているため、SIMD プロセッサコアの最大消費電力である 250mW 程度で動作可能と予想される。また処理クロックサイクル数は前述のように、約 88% 削減できることがシミュレーションによりわかっている。従って、面積は 22% 程度増加するがそれ以上に処理クロックサイクル数を削減でき、消費電力も同等と予想できるため、パターンマッチング TS ブリッジを SIMD プロセッサコアに付加し、JPEG 処理などのマルチメディアアプリケーションの処理を行うことは有効である。

5. まとめ

本研究では超並列 SIMD プロセッサの TS ブリッジに CAM を付加し高速パターンマッチングを可能とするアーキテクチャを提案した。パターンマッチングの一例としてデータ圧縮処理の主流であるハフマン符号化処理を挙げ、提案アーキテクチャであるパターンマッチング TS ブリッジにおいてハフマン符号化を行う方法を提案した。シミュレーション結果より超並列 SIMD プロセッサに比べ約 88%、DSP に比べ約 60% 処理クロックサイクル数を削減できるという結果を得た。また論理合成の結果、パターンマッチング TS ブリッジは動作周波数 200MHz で動作し、フルカスタム設計を行うことにより面積を 0.39mm^2 まで削減できると予想され、パターンマッチング TS ブリッジの面積増加を抑えることができる。従って、パターンマッチング TS ブリッジを超並列 SIMD プロセッサに付加することにより、画像に対する JPEG 処理を高速に行うことが可能であると考える。

本研究で使用したハフマン符号化処理は、ハフマン符号化テーブルを更新しない静的ハフマン符号化 [4]、であった。近年、ハフマン符号化テーブルを動的に更新する方法が提案されている [9], [10], [11]。動的にハフマン符号化テーブルを更新する方法をパターンマッチング TS ブリッジに搭載することによる、ハフマン符号化の圧縮率の向上の実現は今後の課題である。

マルチメディア処理には、大量のデータに対して同じ操作を行う処理とパターンマッチング処理が組み合わさっていることが多い。そのため、高速パターンマッチング機能を有する超並列 SIMD プロセッサにてマルチメディア処理を行った場合、超並列 SIMD プロセッサの処理能力を大幅に向上することができ、

従来の DSP に比べ高速に処理することができる。更に、[12] で提案されている最小距離検索連想メモリを融合することにより、より高度な情報処理の実現も今後の課題である。

謝辞

本研究の一部は、21 世紀 COE プログラム “テラビット情報ノエレクトロニクス” の支援により行われた。

文 献

- [1] M. Nakajima et al., “A 40GOPS 250mW massively parallel processor based on matrix architecture”, ISSCC Dig. Tech. Papers, Paper 22.5, pp. 410-411, Feb. 2006.
- [2] 谷崎哲志, 行天隆幸, 野田英行, 中島雅美, 水本勝也, 堂坂勝己, “マトリクス型超並列 SIMD プロセッサのためのデータ変換バスブリッジの開発”, 信学技報, vol. 106, no. 207, ICD2006-79, pp. 1-6, 2006 年 8 月
- [3] H. Noda, T. Tanizaki, T. Gyohten, K. Dosaka, M. Nakajima, K. Mizumoto, K. Yoshida, T. Iwao, T. Nishijima, Y. Okuno, and K. Arimoto, “The circuits and robust design methodology of the massively parallel processor based on the matrix architecture.” in Symp. VLSI Circuits Dig. Tech. Papers, June 2006, pp. 260-261.
- [4] 越智宏, 黒田英夫, “図解でわかる画像圧縮処理技術” 日本実業出版, pp. 76-79, Jan. 1999.
- [5] S. J. Lee, et al., “An efficient memory allocation schema for Huffman coding of multiple sources,” Signal Process.: Image Commun., Vol. 14, pp. 311-323, Mar. 1997.
- [6] S. M. Lei, M. T. Sun, “An entropy coding system for digital HDTV applications,” IEEE Trans. Circuits Syst. Video Technol., Vol. 1, No.1, pp. 147-155, Mar. 1991.
- [7] “Data compression with MUSIC CAMs,” MUSIC Semiconductors Application Note AN-N6, Nov. 1998.
- [8] K. Pagiamtzis, A. Sheikholeslami, “Content-addressable memory (CAM) circuits and architectures: A tutorial and survey,” IEEE J. Solid State Circuit, vol. 41, No. 3, Mar. 2006.
- [9] T. Kumaki, Y. Kuroda, T. Koide, H. J. Mattausch, H. Noda, K. Dosaka, K. Arimoto, K. Saito, “Multiport CAM based VLSI architecture for Huffman coding with real-time optimized code word table”, Proc. IEEE International Midwest Symposium on Circuit And Systems (MWSCAS'05), pp. 55-58, Aug. 2005.
- [10] Y. Kuroda, T. Kumaki, T. Koide, H. J. Mattausch, H. Noda, K. Dosaka, K. Arimoto, K. Saito, “Highly parallel Huffman encoding by exploiting multiple-matches in content addressable memory”, Proc. The International SoC Design Conference, pp.313-316, Oct. 20-21, 2005
- [11] T. Kumaki, Y. Kuroda, M. Ishizaki, T. Koide, H. J. Mattausch, H. Noda, K. Dosaka, K. Arimoto, K. Saito, “Real-time Huffman encoder with pipelined CAM-based data path and code-word-table optimizer”, IEICE Trans. Inf. & Syst. (in press), 2007.
- [12] H.J. Mattausch, T. Gyohten, Y. Soda and T. Koide, “Compact associative-memory architecture with fully-parallel search capability for the minimum hamming distance”, IEEE J. Solid State Cir., 37, 218-227, 2002