

A-02

動的再構成可能プロセッサによる JPEG エンコーダの設計とその評価

A JPEG Encoder Design on Dynamic Reconfigurable Processor

鈴木 友規† 中田 靖人‡ 吉田 敦郎† 神戸 尚志‡
Yuuki Suzuki Yasuhito Nakata Aturou Yoshida Takashi Kambe

1. はじめに

動的再構成可能デバイスは、演算器、マルチプレクサ、レジスタなどからなるユニットを基本構成要素とした粗粒度構成をとり、回路の再構成に数十ミリ秒を要する FPGA に対して、ナノ秒オーダーで切り替えることができる。この再構成技術により、限られた面積で複雑な機能を実現でき、容易に修正や拡張を行うことができるデバイスとして近年注目されている。それに伴い、その有効性を定量的に評価することが必要と考えられる。

本研究では動的再構成デバイス的一种であるアイピーフレックス社が開発した DAPDNA-2 を用いて JPEG エンコーダを設計し、その評価を行う。

2. DAPDNA-2

DAPDNA-2 とは、DAPDNA アーキテクチャを採用した二世代目の LSI である。この LSI は、主にシーケンシャルな処理を担当する 32 ビット RISC プロセッサコアの DAP(Digital Application Processor)と、大規模なデータ処理や演算処理を担当する 2 次元アレイ状構造の DNA(Distributed Network Architecture)を内蔵しており、動的に回路構成を切り替えることができる。

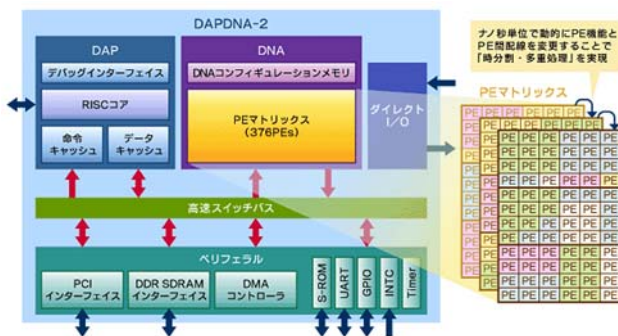


図1 DAPDNA-2 のブロック図

DAP は 32 ビット RISC プロセッサであり、動作周波数は 166MHz で動作する。用途としては、一般的な汎用プロセッサとしての利用や、DNA に内蔵されている PE マトリックスの動的再構成の制御に使用する。

DNA はマルチコンテキスト方式を採用した動的再構成可能であり、32 ビット幅演算器 PE(Processing Element)による粗粒度構成をとる。この PE は様々な処理に対応できるように複数の種類が存在する。動作周波数は DAP と同様 166MHz で動作する。また PE の動作設定と PE 間の接続設定を定義したものを DNA コンフィギュレーションという。

DAPDNA-2 が保持できる DNA コンフィギュレーション・データの数は、実行用バンクで 1 面、DNA 内の DNA コンフィギュレーションメモリであるバックグラウンドバ

ンクに 3 面の計 4 面分である。これらの DNA コンフィギュレーションを動的に切り替えることにより、複雑な処理や大規模な回路を実現することができる。

DAPDNA-2 で処理の実装を行う場合は主に DNA Designer を用いる。DNA Designer は DNA コンフィギュレーションを対話的に入力・編集し、シミュレーションするツールである。ブロック図などの GUI で PE を表現し、PE マトリックスの PE を処理順に配置、PE 間の入出力ポートに信号線を接続することで、データ処理構造を作成する。DNA Designer で作成した DNA コンフィギュレーション・データは DNA Compiler を用いて DAPDNA-2 の物理制限に適合するように自動で配置配線を行う。しかし使用 PE 数が多い場合、DNA コンパイラによる自動配置配線は失敗する可能性が高くなる。このような場合、手動で PE にセグメント位置情報を与えることで、配置配線を行う。

3. JPEG エンコーダ概要

JPEG エンコーダは以下の順で静止画像圧縮を行う。入力画像データを 16×16 画素領域である MCU (Minimum Coded Unit) へと分割する。各 MCU に対し、3 原色の RGB 信号から、輝度と色差を表す YCbCr 信号へと色空間の変換を行う。ここで、符号化効率を高めるためにサブサンプリングによって色差成分の間引きを行い、成分毎に 8×8 画素のブロックを構成する。各ブロックに対して離散コサイン変換 (DCT) を行い、ブロックの画素値を周波数データである DCT 係数へと変換する。DCT 係数 1 つあたりの符号量を削減するために量子化を行い、ジグザグスキャンによって低周波成分から順にブロックの要素を並び替える。最後に、ブロックの係数マトリックスの左上端にある直流 (DC) 成分と残り 63 要素の交流 (AC) 成分に対して順にハフマン符号化を行うことで、静止画圧縮を行う。

4. JPEG エンコーダ設計

JPEG エンコーダ処理全体を C 言語で記述し、これをもとに JPEG エンコーダの色空間変換からハフマン符号化を DNA 用に設計し動作確認を行なう。サブサンプリングに関しては、複雑なメモリアクセスを必要とし、DNA での設計は困難であるため DAP で処理を行なう。以下に DNA 部の設計について述べる。

4.1 色空間変換

色空間変換式を以下の式(1)~(3)に示す。

$$Y = 0.2990R + 0.5870G + 0.1140B - 128 \quad (1)$$

$$Cb = -0.1687R - 0.3313G + 0.5000B \quad (2)$$

$$Cr = 0.5000R - 0.4187G - 0.0813B \quad (3)$$

式(1)~(3)では小数演算を行っているが、DNA では浮動小数点演算を扱うことが困難であるため、計算の簡略化を行う。

†近畿大学総合理工学研究科エレクトロニクス系工学専攻

‡本田技研工業株式会社

‡近畿大学 理工学部電気電子工学科

式(1)~(3)の各係数を2のべき乗表現とし、シフト除算による乗除算を行う。

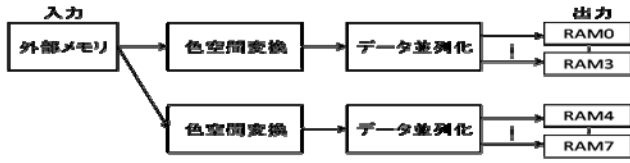


図2 色空間変換処理の流れ

色空間変換を2並列処理で実現する(図2)。まず外部メモリと通信を行い、画像データをDNAに入力する。次に入力された画像データに対して、色空間変換を行う。この後のDCT部で行うバタフライ演算の処理効率化を図るため、変換とデータ出力を並列化し、8個の内部RAMに結果を出力する。1MCUあたりの処理時間は並列化適用前 $2.79 \mu s$ 、2並列処理適用後 $1.59 \mu s$ となり約1.75倍の高速化となる。

4.2 DCT

JPEG エンコードで用いる2次元DCTの式を式(4)に示す。

$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^7 \sum_{y=0}^7 S_{yx} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \quad (4)$$

$$C_u C_v = \begin{cases} \frac{1}{2} & u, v = 0 \text{ のとき} \\ 1 & \text{それ以外} \end{cases}$$

2次元DCTはループ回数が多く、また三角関数や除算を多く含む。これをDNAに実装するため、2回の1次元DCTで実現する。1次元DCT演算を行う式を式(5)に示す。1次元DCTをx方向に処理後、転置し、再度1次元DCTと転置によって、2次元DCTと同様の出力を得る(図3)。

$$C = \frac{1}{2} C_u \sum_{x=0}^7 S_{ux} \cos \frac{(2x+1)u\pi}{16} \quad (5)$$

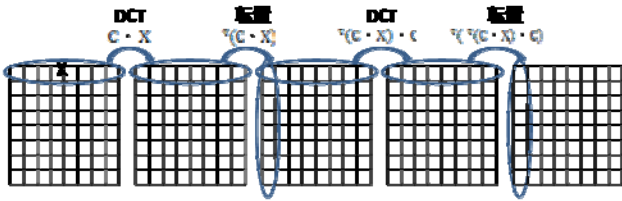


図3 2次元DCTの処理手順

4.2.1 Chen アルゴリズムの適用

DCT 演算には Chen アルゴリズムによる高速バタフライ演算を用いる。Chen アルゴリズムとは零行列が多い疎な行列に分解して、ゼロ行列部分の計算を省くことで、高速な演算するアルゴリズムである。Chen アルゴリズムによる8画素(8×8 ブロックのx成分)に対する1次元DCTの演算を図4に示す。1MCUあたりの処理時間はChen アルゴリズム適用によって $9.54 \mu s$ から $2.65 \mu s$ となり、約3.60倍の高速化となる。

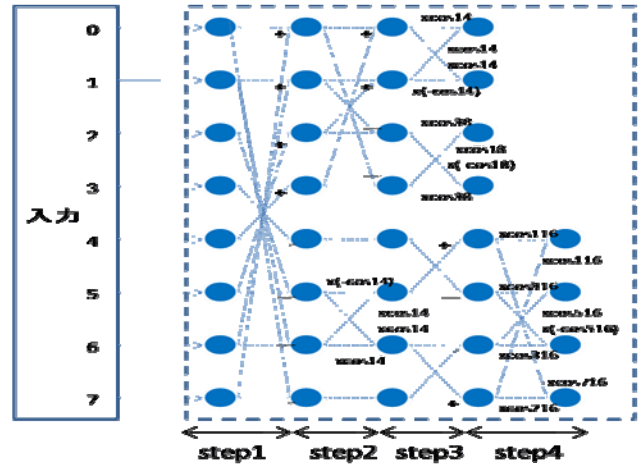


図4 Chenの高速DCTアルゴリズム

4.2.2 1次元DCTの高速化

バタフライ演算はstep1~4の4段階で処理を行う。各ステージをパイプライン化し、かつ入力0~7を並列に処理することで処理の高速化を図る。また1次元DCT中の三角関数は2のべき乗に置換し、シフト処理による乗除算を行う。1MCUあたりの処理時間はこの並列化によって $2.65 \mu s$ から $0.51 \mu s$ となり、約5.20倍の高速化となる。

4.2.3 転置

転置はx方向への1次元DCTを行った後にy方向に1次元DCTを行った後に2回行う。内部メモリ(RAM)に入力データをいったん記憶するが、この内部メモリはデータの書き込みと読み出しを同時に行うことができない。データ競合を避けるために入力データを1MCU間隔を空けて入力すると、処理時間の増大となる(図5_上図)。入力データを記憶テーブルに格納せず、直接x方向からy方向に転置することで、サイクル単位のデータ入力および入力8列の並列化が実現する(図5_下図)。1MCUあたりの処理時間はこの並列化によって $4.85 \mu s$ から $0.82 \mu s$ となり、約5.91倍の高速化となる。

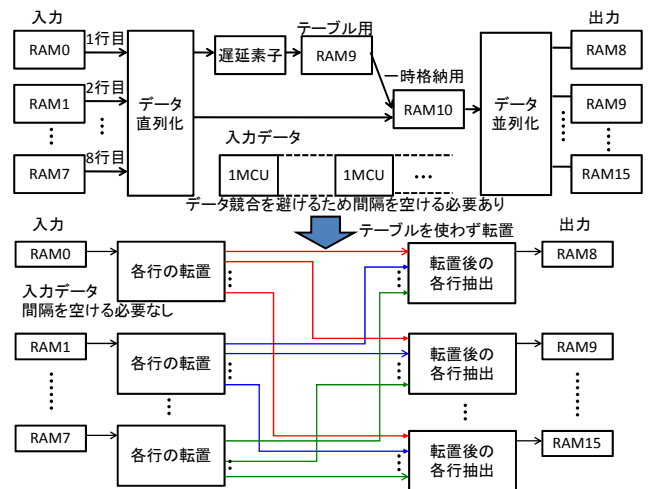


図5 転置の並列化

4.3 量子化

量子化ではDCT出力の周波数成分に対して量子化テーブルを用いた除算を行う。しかし除算は回路が複雑化するので、量子化テーブル値の逆数を取り、2のべき乗を用いて値の整数化を行う。量子化演算後、2のべき乗で除算することで元の数値表現に戻す。DCT演算が8並列化なので、処理の連続性を考えて量子化も入力8列を並列に処理する(図6)。1MCUあたりの処理時間はシフト乗除算適用によって4.91 μs から2.59 μs となり約1.90倍の高速化となる。また8並列化によって2.59 μs 、0.41 μs となり約6.32倍の高速化となる。

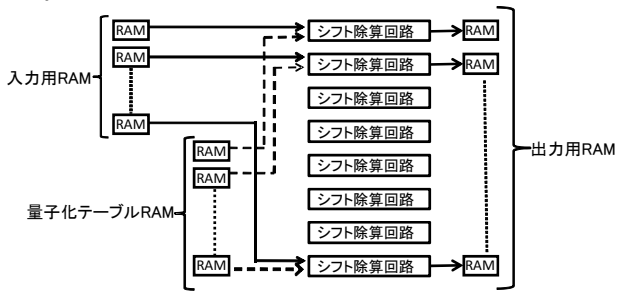


図6 8並列量子化回路

4.4 ジグザグシーケンス

転置処理を削減するため、ジグザグシーケンス処理の方向をx方向からy方向へのシーケンスに変更する(図7)。

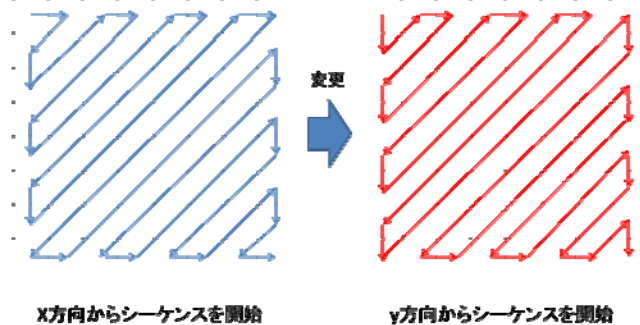


図7 ジグザグシーケンスの処理方向変更

これにより、ジグザグシーケンスと転置を同時に行うことができ、2回目の転置を削減できる。

これらの変更により、2回目の1次元DCTと量子化を1枚のDNAコンフィギュレーションに納め、パイプライン化による高速化を実現する。1MCUあたりの処理時間はDCTと量子化をあわせたパイプライン化によって3.07 μs から1.89 μs となり約1.62倍の高速化となる。

ジグザグシーケンス処理は転置と同様にテーブルを用いて行う。データの競合を避けるためには、テーブルの行列情報から入力データの出力タイミングを調整する(図8上図)。図8上図より入力RAM0~7は行0~7と対応し、RAMのアドレスは列情報に対応する。ここでジグザグシーケンスに従いブロックから0番目の値を読み出す場合、値は0行0列目に格納されている。つまりRAM0のアドレス0に格納されているデータを読み出すことに等しい。同様の手順をブロック0~63番目に行いブロック単位で繰り返すことでジグザグシーケンスを実現する。ブロック図を図8下

図に示す。サイクル単位パイプライン化が可能となり、1MCUあたりの処理時間は4.87 μs から2.51 μs となり約1.94倍の高速化となる。

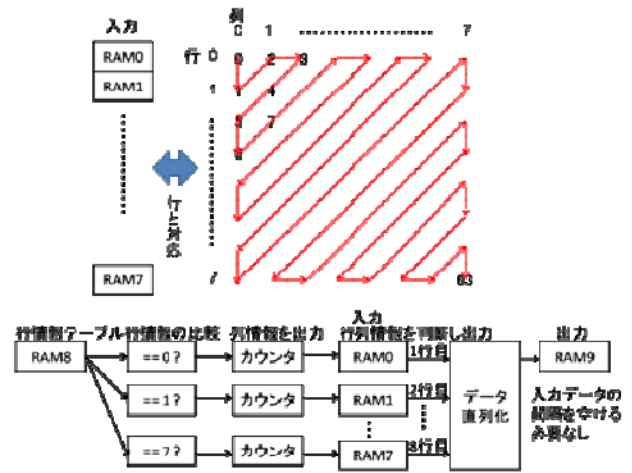


図8 ジグザグシーケンスのパイプライン化

4.6 ハフマン符号化

ハフマン符号化は、直流(DC)成分と交流(AC)成分に対して順に符号化を行う。

DC成分では前のブロックのDC成分値との差分値を計算する。内部メモリを2つ用意し、1ブロック分データをずらして読み出すことで差分値計算を行う方法がある。しかし、内部メモリはデータの書き込みと読み出しを同時に行うことができないため、ジグザグシーケンス処理結果を内部メモリに格納した後に差分値計算を開始するにはDNAコンフィギュレーションを切り替える必要である(図9上図)。これを解決するため、差分値計算するPEでDC成分値をx、yの2ポートから入力し、PE内部でレジスタを用いてyのポートの値のタイミングをずらし、次のブロックの値と減算することで内部メモリを使用せずに行う。これによりジグザグシーケンスとハフマン符号化を1枚のDNAコンフィギュレーションに収め、パイプライン化を実現する(図9下図)。

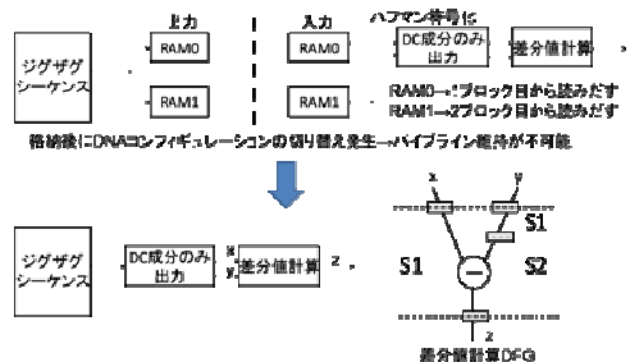


図9 ハフマン符号化のパイプライン化

AC成分は0が連続する数ランレングスを用いて符号化を行う。AC成分が0か判断し、0の場合にランレングスをカウントする。0でない場合はランレングスが中断されるため

パイプラインを維持できない(図 10_上図)。これを解決するため、0 とそうでない場合の処理を各々並列に処理し符号化を行う(図 10_下図)。

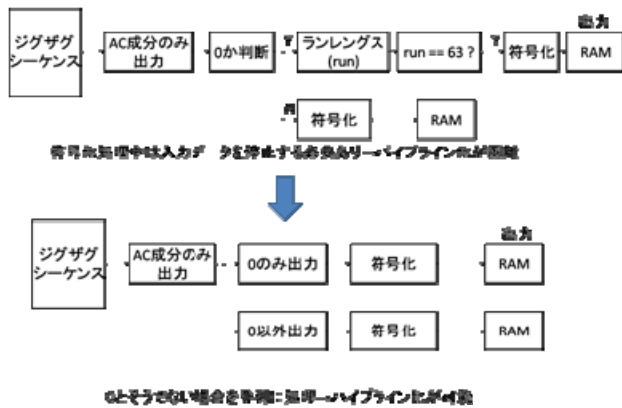


図 10 ハフマン符号化の並列化

1MCUあたりの処理時間はパイプライン化により $10.84 \mu s$ から $2.94 \mu s$ となり約 3.69 倍の高速化となる。ジグザグシーケンスとハフマン符号化を1枚のDNAコンフィギュレーションに収めたことで $5.45 \mu s$ から $3.24 \mu s$ となり約 1.68 倍の高速化となる。

JPEG エンコーダ全体の動作は、各 MCU に対し、色空間変換からハフマン符号化まで、計 5 面の DNA コンフィギュレーションを切り替えることで一つの画像全体の圧縮を行う。

5. 設計結果と評価

4章で述べた設計結果とその評価については、現在確認中であるため、発表時に述べる

6. まとめと今後の課題

JPEG エンコーダを動的再構成可能デバイスである DAPDNA-2 で実現した。DNA コンフィギュレーションの設計では、DAPDNA-2 の特徴的方法であるサイクルパイプライン処理と並列化処理を行い、高速化と回路面積の向上を行った。

色空間変換、DCT、量子化では、シフト演算を用いた小数点演算の簡略化及び処理の並列化、転置ではデータを直接 x 方向から y 方向に転置することで並列にデータを転置することを可能にした。順序的な処理が中心であるジグザグシーケンスとハフマン符号化についてはデータについての並列化や処理方法の変更などにより高速化を図った。

DNA Designer による設計は動的再構成可能デバイスに対する専門知識が要求されることから開発に時間がかかる。C 言語などの高級言語からサイクルパイプライン処理を最大限に可能とするリコンフィギュラブルアーキテクチャ用コンフィギュレーションを自動生成する技術が重要である。

謝辞

DAPDNA 設計ツールの利用法についてサポート頂いたアイピーフレックス株式会社に厚く御礼申し上げます。

参考文献

- [1] 末吉敏則, 天野英晴ほか, リコンフィギュラブルシステム, オーム社, 東京, 2005
- [2] DAPDNA-2 リファレンス, アイピーフレックス(株)
- [3] Takayuki, S., et al. “Dynamically Reconfigurable Processor Implemented with IPFlex’s DAPDNA Technology”, IEICE Trans. Inf. & System., Vol.E87-D, No.8, 2004
- [4] 橋本晋之介, JPEG 概要から C++での実装まで, ソフトバンクパブリッシング, 2005
- [5] 吉田敦郎, 東裕司, 宮崎渉, 田中照人, 神戸尚志, リコンフィギュラブルプロセッサを用いたリードソロン符号復号化回路の設計, 電子情報通信学会技術研究報告書, VLD2007-167, pp.65-68, Mar. 2008.
- [6] 古島直樹, 渡邊誠也, 名古屋彰, 動的再構成可能プロセッサへの JPEG エンコーダの実装と評価, 電子情報通信学会技術研究報告書, RECONF2008-24, pp.7-12, Sep. 2008.