

動的部分再構成技術を用いた JPEG デコーダの機能分割実装

八並泰一郎[†] 吉廣 秀章[†] 久我 守弘^{††} 末吉 敏則^{††}

^{†,††}熊本大学大学院 自然科学研究科

E-mail: [†]{yatsunami,yoshihiro}@arch.cs.kumamoto-u.ac.jp, ^{††}{kuga,sueyoshi}@cs.kumamoto-u.ac.jp

あらまし 動的部分再構成技術とは、回路の他の部分の動作を止めることなく回路の一部を再構成することで、FPGA の柔軟性を高める技術である。著者らは、動的部分再構成技術を用いた動的再構成システムの開発を行ってきた。これまでの我々の研究においては動的再構成システムで実装したアプリケーションは回路規模が小さいものであったため、アプリケーション単位で再構成を行う方法で実現できた。しかし、再構成を行うアプリケーションの回路規模が大きくなると割当てられるリソースが不足し実装できない可能性がある。そこで、アプリケーションを機能毎に分割し、その機能単位での部分再構成を行う機能分割実装に着目し評価を行った。JPEG デコーダの機能分割実装を行った評価の結果、JPEG デコーダ全体を実装するのに必要なリソースの約 5 分の 3 以下のリソースで実装可能であることがわかった。

キーワード FPGA, 部分再構成, 機能分割, 動的再構成

Functionally-partitioned JPEG Decoder for Partial Dynamic Reconfiguration

Taiichiro YATSUNAMI[†], Hideaki YOSHIHIRO[†], Morihiko KUGA^{††}, and Toshinori SUEYOSHI^{††}

^{†,††}Graduate School of Science and Technology, Kumamoto University

E-mail: [†]{yatsunami,yoshihiro}@arch.cs.kumamoto-u.ac.jp, ^{††}{kuga,sueyoshi}@cs.kumamoto-u.ac.jp

Abstract FPGA partial dynamic reconfiguration is the technique which makes FPGA more flexible. We developed the dynamic reconfigurable system using the technique. This paper focuses attention on functionally-partitioning as implementation method for dynamic reconfigurable system. Functionally-partitioning is enable to implement into fewer resource than total logic resource required for application. As an example, functionally-partitioned JPEG decoder is evaluated on partial dynamic reconfigurable system. As compared with implementation the total JPEG decoder, the functionally-partitioned JPEG decoder can be implement with about three-fifth or less resource of the required resources.

Key words FPGA, Partial Reconfiguration, Functionally-partitioning, Dynamic Reconfiguration

1. はじめに

近年、FPGA (Field Programmable Gate Array) に代表されるように、回路の可変構造を持つ LSI の研究が盛んに行われている。またそれに伴い、その可変性を積極的に活用する再構成システム (Reconfigurable System) が注目されている。再構成システムの特徴は、アプリケーションの要求に応じて必要なハードウェア機能を再構成できることであり、これによってハードウェアリソースを増加することなく様々な機能を得ることができる。

そこで、我々は FPGA を用いた動的再構成システムである EXPRESS-2 の開発を行っている [1][2]。我々の開発している動的再構成システムの特徴は、FPGA 内部に組み込まれたプロセッ

サがアプリケーションに必要なハードウェアを再構成させる点であり、この機能によってシステム全体の動作を止めることなく FPGA を再構成し新たな機能へ変更することができる。

EXPRESS-2 の開発には、FPGA の部分再構成技術が必要である。再構成時間の短縮や動的な再構成を可能とする等、部分再構成技術は FPGA の柔軟性をさらに高める技術である。我々は、この部分再構成技術を用いた EXPRESS-2 のプロトタイプシステムを開発し、アプリケーション単位での部分再構成を行い評価した [2]。この評価に用いたアプリケーションは回路規模が小さいこともあり、アプリケーション単位で再構成を行う方法で実現できた。しかし、再構成を行うアプリケーションの回路規模が大きくなると割当てられるリソースが不足し実装できない可能性がある。

そこで、本研究ではアプリケーションを機能毎に分割し、機能単位での部分再構成を行う機能分割実装に着目し評価を行う。機能分割実装では少ないハードウェアリソースでアプリケーションを実装可能となる利点がある。また、回路規模が小さくなることにより再構成時間を削減できる可能性もある。

以下、2. では我々が提案している動的再構成システムである EXPRESS-2 について、およびシステム構築に用いた FPGA である Virtex-4 の部分再構成技術について説明する。次に、3. で機能分割実装について述べ、今回実装した JPEG デコーダの機能分割について述べる。4. で Virtex-4 を用いて実際に構築したシステムについて述べ、5. にて機能分割実装の評価を行う。そして、6. でまとめと今後の課題について述べる。

2. EXPRESS-2 と部分再構成技術

EXPRESS-2 には、部分再構成可能なプロセッサ混載 FPGA である米国 Xilinx 社製の Virtex シリーズ [3] を用いることを考えている。しかし、部分再構成技術を用いる際にはそのデバイス特有の制約を満たす必要がある。

そこで、本章では EXPRESS-2 の特徴的な機能やシステムの概観について述べ、使用デバイスである Virtex-4 での部分再構成について述べる。

2.1 動的再構成システム EXPRESS-2

当研究室では、これまでに EXPRESS-1 と呼ぶ動的再構成システムを開発してきた [4]。EXPRESS-1 では、ARM プロセッサを混載した全体再構成型の FPGA を用いてシステムを開発した。現在、部分再構成が可能な FPGA の登場に伴い、我々は EXPRESS-1 からの発展的なシステムである EXPRESS-2 と呼ぶ動的再構成システムの開発を行っている。

図 1 に、我々が提案している動的再構成システムである EXPRESS-2 の概観について示す。EXPRESS-2 には、部分再構成やセルフリコンフィギュレーション、ネットワークを用いた機能拡張といった特徴的な機能がある。以下に、その機能について述べる。

部分再構成： FPGA の部分再構成とは、FPGA のある特定の領域の回路のみを再構成させる機能である。その際に、再構成と関係のない部分の回路はその動作を止める必要はない。EXPRESS-2 では、システムの制御を行う Fixed-module と部分再構成を行う Reconfigurable-module がある。Fixed-module は再構成を行わずその動作も止めない。Fixed-module からの制御により Reconfigurable-module 内を部分再構成し、新たな機能への変更を行う。

セルフリコンフィギュレーション： FPGA に混載されたプロセッサにより FPGA の一部を内部から再構成することで、1 チップ内での再構成の制御を可能にする機能である。本研究で使用する Virtex-4 には、混載プロセッサとして PowerPC が組込まれている。よって、この PowerPC から再構成を行うことになる。また、内部からの再構成については FPGA デバイスによって仕様が異なるが、Virtex-4 では ICAP (Internal Configuration Access Port) インタフェースを介して再構成を行う。

ネットワークを用いた機能拡張： EXPRESS-2 では部分再構成

を行うための構成データが必要となる。その構成データはチップ外のメモリにおくことになるが、そのメモリ容量により搭載できる構成データの数は制限される。これは、変更できる機能を制限することになり、システムの柔軟性を損なうことになる。そこで、FPGA の遠隔再構成技術を用いて、ネットワーク上のデータベースから構成データをダウンロードすることで新たな機能の追加を行う [5]。

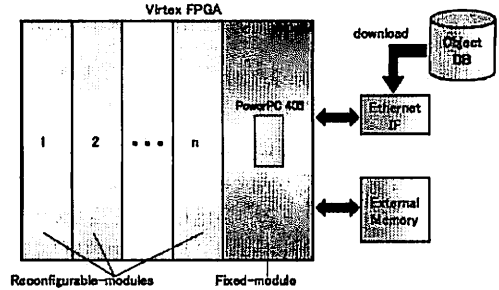


図 1 EXPRESS-2 の概観

2.2 Virtex-4 FPGA での部分再構成

図 2 に、米国 Xilinx 社製 Virtex-4 FPGA のアーキテクチャを示す。図 2 の CLB (Configurable Logic Block) 内に、LUT (Look-up table) および FF (Flip Flop) を備える。Virtex-4 では、内部からの再構成を行うための ICAP をデバイス中心部に備えている。

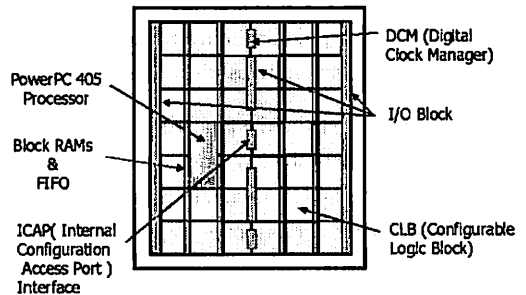


図 2 Virtex-4 FPGA のアーキテクチャ

このような Virtex-4 のアーキテクチャが、部分再構成を用いた設計する際の制約になる。以下、本節ではその制約について述べる。なお、この設計制約は部分再構成を必要とする設計をする際のみ気を付けなければならない。

- モジュール同士が再構成時の境界を越えて通信しなければならない場合、専用のマクロを使用しなければならない。このマクロを Bus Macro と呼ぶ。Virtex-4 では、スライススペースの Bus Macro が Xilinx 社より提供されている。Bus Macro は縦 4 個、横 4 個で合計 16 個の LUT を配線済みのマクロとして使用する。1 つの Bus Macro で上下または左右の一方方向に 8 ビットの通信が可能である。

- Reconfigurable-module の高さは、Virtex-II Pro までの Virtex シリーズにおいてはすべてデバイスの高さと同じでなければならない。一方、Virtex-4 においては 16CLB の高さにクロックリージョンの境界があるため、高さを 16CLB の倍数で設

定できる。これにより任意の長方形で Reconfigurable-module を作成することが可能となった。横幅は 2 スライス (1 列の CLB) の倍数でなければならない。

- 個々の Reconfigurable-module は、直接使用できる I/O がその領域にあるものだけに限られている。
- クロック用のロジック (BUFGMUX, CLKIOB 等) は、Reconfigurable-module にあってはならない。そのため、Virtex-4 では Reconfigurable-module がデバイスの中心を横断することは避ける必要がある。
- すべての Reconfigurable-module は初期配置のエリアで固定されており、変更することはできない。つまり、大きさを変えたり場所をずらしたりすることはできない。

2.3 部分再構成用構成データの生成

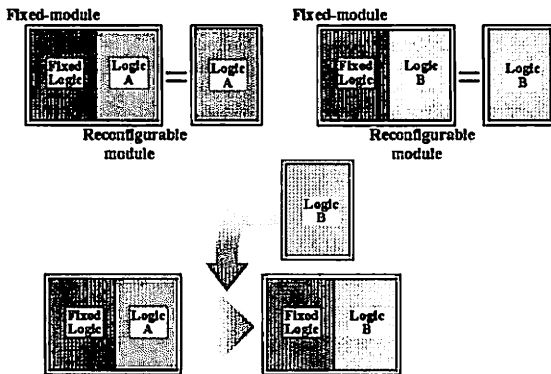


図 3 部分再構成用構成データの生成法

図 3 は、Xilinx 社より提供されている部分再構成用再構成データの作成方法である EAPR (Early Access Partial Reconfiguration) を用いて、部分再構成用再構成データの作成例を示している。この EAPR においては部分再構成の際に再構成前後の関係に依存せずに構成データの作成が可能である。部分再構成用の構成データは Reconfigurable-module に設定した回路部分を抜き出したものである。この部分再構成用構成データは、その生成法の特徴から全体再構成を行う際の構成データよりもデータ量が少なくなる。図 3 に示すように回路の再構成の際には、作成した構成データを使用することにより行う。

3. 機能分割実装

本章では、動的再構成システムにおける実装方法について説明し、その後今回使用した JPEG デコーダおよび JPEG デコーダの機能分割について説明する。

3.1 動的再構成システムにおける実装方法

動的再構成システムにおける実装方法は、アプリケーション毎に動的に再構成する実装方法とアプリケーションを複数の機能に分割してそれぞれの処理単位で動的に実装する方法の 2 つの方法がある。前者の方法は回路規模の小さな回路に対して有効な方法であるが、後者はデバイスへ 1 度に実装できないような大規模のアプリケーションを実装可能となる利点を持つ。また、1 度にデバイスへ実装する回路規模も削減できるために、採用するチップサイズを削減することができる。

我々は先行研究の文献 [2] においては、アプリケーションは回路規模が小さいものであったため、アプリケーション単位で再構成を行う方法で実現できた。しかし、開発している動的再構成システムは大規模のアプリケーション実装も想定している。そこで、本研究では動的部分再構成システムにおける機能分割処理の有効性の確認を行うため、JPEG デコーダを例として機能分割処理による実装評価を行う。

3.2 JPEG デコーダ

JPEG デコーダの処理は Huffman 復号化、逆ジグザグスキャン、逆量子化、逆 DCT の部分である。各処理の説明を以下に記す。

Huffman 復号化: 8×8 画素ブロックの 64 個の入力されたビットストリームに対し、量子化 DCT 係数に変換する。

逆ジグザグスキャン: ハフマン復号化によって復号化されたデータはジグザグスキャンによって 1 次元に変換されたデータであるため、逆ジグザグスキャンによって 2 次元データに並び替えを行う。

逆量子化: 逆ジグザグスキャン後の 2 次元データに量子化ステップサイズを掛けて DCT 係数に変更する。

逆 DCT: 逆量子化処理を施したデータは DCT 変換により輝度成分と色差成分を周波数成分へと変換した 2 次元 DCT データである。そこでこの逆 DCT 変換で輝度成分、色差成分に変換する。

3.3 機能分割

JPEG デコーダを、Huffman 復号化 (HUFF)、逆ジグザグスキャン・逆量子化 (IQNT)、逆 DCT (IDCT) の 3 つの機能モジュールに分割した。機能分割実装を行った各モジュールと JPEG デコーダ全体のモジュールとの論理合成後の回路規模を表 1 に示す。

表 1 全体実装と機能分割実装の回路規模

実装方法	全体実装		機能分割実装		
	JPEG デコーダ	HUFF	IQNT	IDCT	
LUT	8,359	3,993	696	2,817	
FF	2,335	1,438	367	581	
DSP	6	-	6	6	
RAM	12	-	6	6	

LUT と FF, DSP の合計数に全体実装と機能分割実装では差があるがこれは論理合成中の最適化における差であると考えられる。機能分割実装において LUT と FF の割合を比較すると HUFF が JPEG デコーダの半数近くを占めている。また、IQNT, IDCT では DSP と RAM が使用されていることがわかる。なお、機能分割実装を行った際の構成データサイズに関する評価、考察については 5. において述べる。

図 4 に全体実装と機能分割実装のそれぞれにおいて画像の 1 マクロブロックを PowerPC を用いてメモリ~モジュール間のデータ転送により処理させた場合の処理時間の比較のグラフを示す。

図 4 より分かるように、全体実装では前の処理でデータ出力が始まり次第、次の処理を開始していたためにデータ転送の時間を隠蔽することができる。しかし、機能分割ではデータ転送の

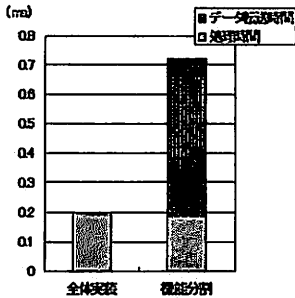


図4 機能分割した際の処理時間の変化

時間が隠蔽できないためにデータ転送の時間が大幅に増えている。しかし、5. で示すようにソフトウェアのみでJPEGデコーダを実行した場合には10ms以上の処理時間が必要であるため、機能分割した際のデータ転送時間の増加を含めても十分な速度向上を得ている。

4. 構築した動的再構成システム

2. にて、EXPRESS-2やVirtex-4の仕様について、およびそれを用いて部分再構成を行う際の制約等について述べた。本章では、2. で述べた制約等を考慮しつつ実際に構築した動的再構成システムや、部分再構成を行う際の構成データの流れについて述べる。

4.1 システムの概要

図5にシステムの概要図を示す。本研究においてはFixed-moduleを1つ、Reconfigurable-moduleを1つまたは2つ備える2種類のシステムを構築した。

Fixed-moduleは、混載プロセッサであるPowerPC 405プロセッサ、プロセッサからの再構成を行うためのICAP、そして、メモリとしてBRAMとSDRAMを持つ。SDRAMはFPGA外にあり、Fixed-moduleにはSDRAMのコントローラがある。各回路はPLB (Processor Local Bus) に接続している。また、OPB (On-Chip Peripheral Bus) にReconfigurable-moduleに載った回路とのデータ転送を行うためのBus Macro、およびその制御回路が接続されている。Reconfigurable-moduleが2つの場合はOPBに両Reconfigurable-moduleが接続された状態となる。

Reconfigurable-moduleとFixed-module間は、Bus Macroにて接続される。バスとの間に、入出力合わせて合計で15個のBus Macroを使用している。Reconfigurable-moduleはOPBに接続されたモジュールと見なし制御することが可能である。そのため、Reconfigurable-moduleにはOPBとのインターフェースとなる回路も組込む必要がある。

Reconfigurable-module内には、部分再構成により様々な回路が実装可能である。しかし、制約に記したように、Reconfigurable-moduleの領域の変更ができないため回路規模がReconfigurable-module内のリソースを超えてしまうものは実装できない。

4.2 部分再構成の流れ

前述したように、ICAPを用いたプロセッサからの再構成で1

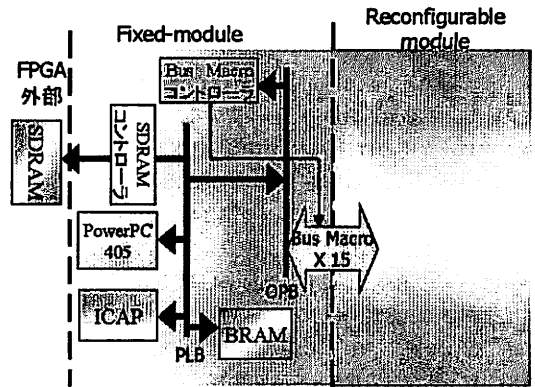


図5 システム (Reconfigurable-module 1つ) の概要

チップでの再構成が可能となる。Virtex-4におけるICAPでは1度に32ビット単位のデータを処理することが可能である。

本研究では、部分再構成を行うための構成データはそのサイズのためSDRAMに置いている。部分再構成を行う際には、PLBを通してPowerPCがSDRAMから構成データを32ビット単位で取得し、ICAPへ送る。これを構成データ量だけ繰り返すことで、再構成が完了する。

この際に再構成を実行する前にOPBに接続しているBus Macroの制御回路を用いReconfigurable-moduleからの入力信号を制限する必要がある。これは再構成の際にOPBへ予期せぬデータが入力されるのを防ぐためである。

5. 評価

本章では、4. で述べたシステムにおいてJPEGデコーダの機能分割の評価を行う。評価を行うにあたり以下の3つの場合における評価を行う。

RMJPEG: JPEGデコーダを全体を1つのReconfigurable-moduleに実装。

RM1: JPEGデコーダの機能分割を行いReconfigurable-moduleが1つのシステムで実装。

RM2: JPEGデコーダの機能分割を行いReconfigurable-moduleが2つのシステムで実装。

また、機能分割の有効性の調査のために以下の項目での調査を行う。

各実装における回路規模と構成データサイズ: 機能分割実装に必要なReconfigurable-moduleの領域および各機能を再構成する際に必要となった構成データのサイズを比較する。

各実装における実行時間と再構成時間: 各実装における1マクロブロックあたりの実行時間と再構成時間の合計を処理時間とし、処理時間とソフトウェア処理を比較し有効性の確認を行う。

5.1 開発環境と評価環境

表2に開発環境を示す。開発ボードは、Xilinx社製ML405評価プラットフォームを用いた。このボードにはXilinx社のVirtex-4 XC4FX20が載っており、混載プロセッサとしてPowerPC 405ブ

ロセッサがある。設計ツールには、同じく Xilinx 社の EDK 9.1i (Embedded Development Kit) と EAPR 用のパッチをあてた ISE 9.1i SP2 PR2 (Integrated Software Environment) を用いた。また、EAPR 用の開発プラットフォームとして PlanAhead 9.1.4 を使用した。

表 2 開発環境

開発ボード	ML405 評価プラットフォーム FPGA : Virtex-4 Pro XC4FX20 混載プロセッサ : PowerPC 405
設計ツール	Xilinx EDK 9.1i, ISE 9.1i SP2 PR2 PlanAhead 9.1.4

5.2 各実装における評価

5.2.1 各実装における回路規模および再構成データサイズ

Reconfigurable-module を実装する時に領域として定義されるものがスライスである。スライスは 1 スライスあたりに 2 つの LUT と FF を含んでいる。評価における回路規模には Reconfigurable-module に割当てたスライス使用率で評価する。まず、今回実装に用いたデバイスである XC4FX20 のスライス数、LUT、FF、DSP、RAM のリソース数について表 3 に示す。このデバイスにおいて、2. で述べたクロック用ロジックの制約よ

表 3 XC4FX20 リソース数

スライス数	8,544
LUT	17,088
FF	17,088
DSP	32
RAM	68

り Reconfigurable-module に使用可能なスライスの最大リソース数は、4,352 である。JPEG デコーダをすべて実装するのに必要なスライス数は、5,086 であるために JPEG デコーダすべてを Reconfigurable-module に実装することはできない。

次に RM1 については図 6 に示すように実装した。この時の Reconfigurable-module での各モジュールのスライス使用率および部分再構成に必要なビットサイズを表 4 に示す。

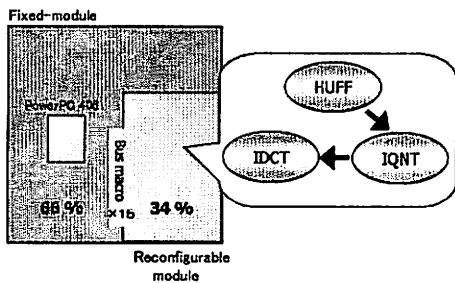


図 6 RM1 の実装方法

表 4 RM1 の実装におけるスライス使用率と再構成データサイズ

モジュール	スライス使用率 (%)	再構成データサイズ (Byte)
HUFF	84.58	197,724
IQNT	14.58	120,528
IDCT	59.44	198,700

Reconfigurable-module の使用可能なスライス数は、2,880 である。RM1 では JPEG デコーダ全体の約 5 分の 3 以下のリソースで実装可能である。HUFF と IDCT では Reconfigurable-module でスライス使用率が 25%以上の差があるが再構成データサイズは IDCT の方が大きくなっている。これは、表 1 に示したように IDCT において DSP と RAM を使用しているためであると考えられる。IQNT も DSP と RAM が使用されているが他のモジュールとのスライス使用率の差が大きいため構成データサイズにも差が生じる。

続いて、RM2 の実装方法を図 7 に示し、Reconfigurable-module での各モジュールのスライス使用率および部分再構成に必要なデータサイズを表 5 に示す。

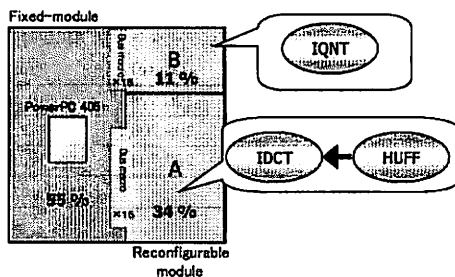


図 7 RM2 の実装方法

表 5 RM2 の実装におけるスライス使用率と再構成データサイズ

モジュール	スライス使用率 (%)	再構成データサイズ (Byte)
HUFF	84.58	198,392
IQNT	43.75	69,388
IDCT	59.44	202,704

RM2 の Reconfigurable-moduleA は RM1 の Reconfigurable-module と同様の領域に設定し、上部に Reconfigurable-moduleB を作成した。Reconfigurable-moduleB に割り当てたスライス数は 960 である。RM2 においては JPEG デコーダ全体の約 4 分の 3 程度のリソースで実装可能である。

Reconfigurable-moduleB に割り当てた領域については、表 1 に示した IQNT の LUT 数のみで見ると表 5 ではスライス使用率が 43.75%であるために割り当てた領域が冗長にみえる。これは IQNT には DSP と RAM が用いられているために DSP と RAM のデバイスでの配置を考慮したためである。しかし、再構成ビットサイズは RM1 の時と比較すると半分近い値になった。これは RM1 の Reconfigurable-module と RM2 の Reconfigurable-moduleB では領域に約 3 倍程度の違いがあるためにそれが影響していると考えられる。また、Reconfigurable-moduleB を上部に配置した影響で HUFF と IDCT の再構成データサイズが変化している。

5.2.2 各実装における実行時間および再構成時間

RM1 と RM2 での各モジュールの処理と再構成を含めた JPEG デコードの流れを図 8 に示す。

RM1 においては、各処理が終了次第、再構成をかけて次の処理を行う。RM2 では、IQNT の処理が終了すると HUFF を IDCT に再構成し、処理を開始する。

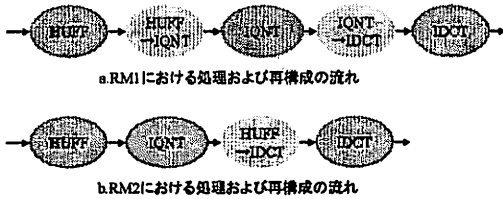


図 8 各実装での処理の流れ

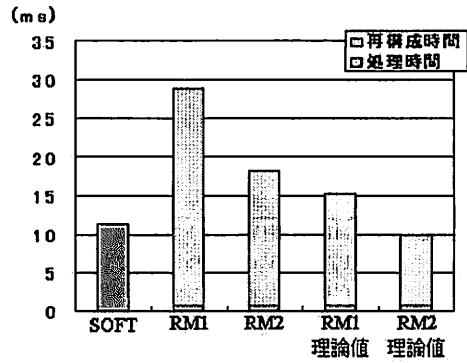


図 9 再構成時間を含めた処理時間の比較

各実装における再構成時間と再構成時のビットレートを表 6 に示す。

表 6 再構成時間についての評価

システム	種類	構成データ量 (Byte)	再構成時間 (ms)
RM1	HUFF → IQNT	120,528	10.48
RM1	IQNT → IDCT	198,700	17.22
RM2	HUFF → IDCT	202,704	17.63
再構成ビットレート			11.5MByte/s

再構成時間は再構成データのサイズに比例して増加する。RM1 においては 2 度の再構成が必要であるために合わせて約 28ms もの再構成時間が必要となる。しかし、RM2 では再構成が 1 度で済むために約 18ms ほどの再構成時間の削減を行うことができる。

再構成時間と各処理時間をもとに混載プロセッサである PowerPC においてソフトウェアで JPEG デコーダを処理した際の HUFF → IDCT までの処理との比較を行う。表 7 に各機能モジュールの動作周波数を示す。

表 7 各機能モジュールの動作周波数

動作周波数	機能モジュール
50MHz	ICAP
100MHz	Reconfigurable-module, BRAM, SDRAM PLB, OPB
300MHz	PowerPC 405

SOFT の処理時間および RM1, RM2 の JPEG デコーダの処理時間を算出したグラフを図 9 に示す。

再構成時間を含めた評価では、ソフトウェア処理と比較して約 2 倍から 3 倍の処理時間が必要であった。しかし、現在のデータ転送はすべて PowerPC405 を介してプログラムの処理となっている。そのために SDRAM ~ ICAP 間のデータ転送での PowerPC の処理がボトルネックとなっている。

そこで、図 9 には PowerPC を介さずに DMA (Direct Memory Access) によるデータ転送を想定した際の、処理時間を計算した値 (理論値) を示す。この処理時間の理論値は、作成したシステムをもとに SDRAM コントローラでの 32 ビット転送の READ/WRITE サイクルタイム、および PLB とモジュール間通信回路の READ/WRITE 時にかかるクロックサイクルと動作周波数からデータ転送にかかる時間を計算し再構成時間を求め

たものである。理論値と SOFT を比較すると RM1 では約 4ms 程度速度が劣っているが、RM2 では約 1ms ほど SOFT よりも速度向上が見込めることがわかった。

6. まとめと今後の課題

本稿では、動的部分再構成技術を用いた JPEG デコーダの機能分割実装の評価を行った

本研究では、JPEG デコーダの Huffman 復号化、逆量子化、逆ジグザグスキャン、逆 DCT 処理を機能分割実装を行い、3 つのシステムにおける評価を行った結果、JPEG デコーダを機能分割せずに実装しようとするとリソース不足のために実装できなかったが、機能分割実装で最小で約 5 分の 3 以下のリソースで実装可能であることが分かった。また、処理時間に対しては、再構成時間を加えるとソフトウェア処理と比較して約 2 倍から 3 倍の処理時間が必要であった。しかし、その原因は再構成時間によるものであるため、将来マルチコンテキストで高速に切り替え可能なデバイスが登場すればソフトウェアと比較して 10 倍以上の速度向上を得た上で、かつ少ないリソースでの実装が可能である。

今後の課題としては、本稿では計算により理論値として求めていたハードウェアでの処理時間を実測することがある。構築したシステムに DMA 転送機能を付け PowerPC を介さないデータ転送を行い、その処理時間を計測する必要がある。また、今回は JPEG デコーダによる実装であったが他のアプリケーションについても調査を行う予定である。

文 献

- [1] 坂本伊左雄, 須崎貴憲, 柴村英智, 飯田全広, 久我守弘, 末吉敏則: “プロセッサ混載 FPGA における部分再構成制御機構の開発,” 信学技報, vol.105, no.287, pp.43-48, 2005.
- [2] 清田享伸, 八並森一郎, 木佐貞健, 吉廣秀章, 久我守弘, 末吉敏則: “動的再構成システムにおける FPGA 部分再構成技術の適用と評価,” 信学技報, vol.106, no.454, pp.55-60, 2007.
- [3] Xilinx Inc.: “Virtex-4 User Guide,” http://www.xilinx.com/support/documentation/user_guides/ug070.pdf, 2007
- [4] 柴村英智, 飯田全広, 久我守弘, 末吉敏則: “EXPRESS-1: プロセッサ混載 FPGA を用いた動的セルフリコンフィギュラブルシステム,” 信学論 (D), vol. J89-D, no.6, pp.1120-1129, 2006.
- [5] 牛嶋和行, 永田和生, 原田英雄, 久我守弘, 末吉敏則: “FPGA 搭載機器のための自己更新機構の開発,” 信学技報, Vol.106-247, pp.61-66, 2006.