

パラメータ付けされた動的再構成可能システムとその応用

香西 伸治[†] 戸川 望^{††} 柳澤 政生[†] 大附 辰夫[†]

[†] 早稲田大学理工学部電子・情報通信学科
〒 169-8555 東京都新宿区大久保 3-4-1
E-mail: kozai@yanagi.comm.waseda.ac.jp
yanagi@yanagi.comm.waseda.ac.jp
to@ohtsuki.comm.waseda.ac.jp

^{††} 早稲田大学理工学総合研究センター
〒 169-8555 東京都新宿区大久保 3-4-1
E-mail: togawa@ohtsuki.comm.waseda.ac.jp

あらまし

近年、システムの動作中にシステムの一部の論理を書き換える動的再構成可能システムが研究されている。従来、動的再構成可能システムはそのハードウェア構成が固定されており、そのため、特定のアプリケーションに適用するにはそのシステム上の構成では機能が冗長、あるいは、不足するという問題があった。本研究では、これらの問題を解決するためにパラメータ付けされた動的再構成可能システムを提案する。提案する動的再構成可能システムは PCI インターフェース部、演算を実行する機能部、それらを制御する制御部から成る。機能部の処理速度や回路規模、再構成時間、ピン数などのデバイス構成やデバイス間の接続方式はアプリケーションに応じて可変である。デバイスのパラメータにはその性能に応じてコストが割り当てられておりユーザの指定した制約コスト内での最速処理をするデバイスのパラメータとそのデバイスから成るシステム構成の決定が可能である。本研究ではアプリケーションの適用例として画像符号化処理とパケット処理を挙げ、システムの有効性を評価する。

キーワード 動的再構成可能システム、再構成可能デバイス、FPGA、スケーラビリティ、画像符号化処理、パケット処理

A Parameterized Dynamic Reconfigurable System and Its Application

Shinji KOZAI[†] Nozomu TOGAWA^{††} Masao YANAGISAWA[†] Tatsuo OHTSUKI[†]

[†] Dept. of Electronics, Information and
Communication Engineering, Waseda University
3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan
E-mail: kozai@ohtsuki.comm.waseda.ac.jp
yanagi@yanagi.comm.waseda.ac.jp
to@ohtsuki.comm.waseda.ac.jp

^{††} Advanced Reserch Center for Science and
Engineering, Waseda University
3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan
E-mail: togawa@ohtsuki.comm.waseda.ac.jp

Abstract

Recently, there has been proposed a dynamically reconfigurable system where a part of the system can be reconfigured in-system. A conventional Hardware configuration in dynamically reconfigurable systems is fixed and cannot be changed. Therefore, there is a problem that functions of the system configuration can be redundant or insufficient. In this work, we propose a scalable dynamic reconfigurable system with configuration parameters which resolves the problem. The proposed dynamic reconfigurable system is composed of a PCI interface, Function Units which execute operations and a control unit which controls a PCI interface and Function Units. The device configuration parameters such as processing speed, size of the device, reconfiguration time and the number of pin as well as connections among devices in Function Units can be determined depending on applications. The device parameters have costs depending on its performance, and users can determine the system configuration and device parameters so that the application can run as fast as possible under a given cost constraint. In this work, we evaluate effectivity of this system by applying it to image processing and packet processing applications.

Key Words *dynamic reconfiguration, reconfigurable device, FPGA, scalability, image processing, packet processing*

1 まえがき

近年、システムの動作中にシステムの一部の論理を書き換える動的再構成可能システムが研究されている。動的再構成可能システムを実現するデバイスとしてFPGA (Field Programmable Gate Arrays) が挙げられる。FPGAは、予めチップに敷いてある論理ブロックの機能と配線パターンをユーザが手元でプログラミングすることで、任意の論理を構成可能にしたプログラマブルデバイスである。登場当初より、FPGAはゲート数の少なさがボトルネックとされ実用的ではなかったため、プログラマビリティや高いテスト性を生かし、システムの簡易プロトタイプ[2],[10],[14]、エミュレータ[1],[7],[9]、小規模のアプリケーション向けデバイス[15]として利用されてきた。

ある論理を割り当てられたFPGAに別の論理を割り当てることをFPGAの再構成と呼ぶ。FPGAの再構成機能を利用して、専用ハードウェアの高速性とソフトウェアの汎用性の両立を狙ったシステムを再構成可能システムという。FPGA内部の論理ブロックの機能や、論理ブロック間の配線パターンを論理構成データと呼ぶ。論理構成データを保持する素子にSRAMを用いたタイプのFPGAは、電源を投入したまま再構成できる。これは、アプリケーションを実行中にある論理を実行し終えたFPGAに別の論理を割り当てる動的再構成[3],[12]が可能であることを意味する。動的再構成可能システムのある瞬間における論理規模は、システムの物理的な論理規模によるが、動的再構成により時系列で仮想的に論理規模を増大できる。

動的再構成可能システムは、これまで、文献[4]-[6],[11],[13],[16]等の研究例が報告されている。動的再構成可能システムは、(a)既存のFPGAを用いたシステム、および(b)動的再構成に特化した独自のデバイスを用いたシステム、に分類される。文献[5],[6]は、(a)に分類される数少ない例である。既存のFPGAは再構成に数msから数10ms必要とされ、動的再構成可能システムに適用する場合にはシステム全体のボトルネックとなるが、文献[6]のシステムはJPEGエンコードで計算機に対して2倍の処理速度を達成している。また、文献[6]のシステムは、システムに搭載される各FPGAが固定接続されており拡張性に乏しい、論理構成データの保持にEPROMを使用しており論理規模がEPROMの容量に制限される、といった欠点を持つ。文献[5]のシステムは、それらの欠点を解決すべく演算ユニットを制御ユニットに並列にバスに接続し拡張性を確保し、ホスト計算機上から論理構成データをPCIバス経由で転送することにより論理構成データの容量がEPROMに制限されないという特長を併せ持つシステムである。文献[5]のシステムは、暗号化アルゴリズムで計算機に対して25倍の処理速度を達成している。文献[4],[11],[13],[16]は(b)に分類される。文献[11]は、仮想デバイスMPLDを用いた仮想ハードウェアである。文献[16]は、部分再構成可能なFPGAである。文献[4],[13]は、デバイス内部に論理構成データを保持し、高速に再構成可能なデバイスである。文献[4]は、4.9nsで再構成できる。文献[13]は、100nsで再構成できる。独自のデバイスを用いた動的再構成可能システムは、仮想デバイスであり、専用のCADを必要とする。以上の分類(a),(b)にて、(a)は既存のFPGAを用いているために実現性が高いが、近年のアプリケーションの高集積化、高速化を考慮すると再構成時間がボトルネックとなることは避けられない。(b)は高速な再構成時間を達成しているが現在では研究段階である。また、(a),(b)の分類に関わらず動的再構成可能システム[4]-[6],[11],[13],[16]等では、システム上の動的再構成可能デバイスやデバイス構成が固定されているために、特定のアプリケーション

に適用するには、そのシステム上のデバイスやデバイス構成では機能が冗長、あるいは不足するという問題がある。

以上の背景より、本稿では大規模なデジタル信号処理アプリケーションの高速実行を目的にパラメータ付けされたスケラビリティのある動的再構成可能システムを提案する。本システムは、ホストPCとのデータの授受を担うPCIインターフェース部、動的再構成可能な複数の演算部、システムを制御する制御部で構成される。演算部のデバイス構成やデバイス性能(最大周波数や再構成時間)、制御部と演算部のデバイス間の接続方式は適用するアプリケーションに応じて可変である。本システムはデバイスの最大周波数、回路規模、再構成時間、I/Oピン数をパラメータとしてそれぞれにコストが割り当てられており、ユーザの指定した制約コスト内での最速処理をするパラメータの出力が可能である。本システムのパラメータは既存のデバイスならびに現在は研究段階であるが将来的に実現性のあるデバイスも対象としている。そのため、本システムの分類は前述の(a),(b)の中庸であると言える。

本稿は以下のように構成される。2章では、動的再構成可能システムによって実現される仮想的な論理規模の増大を示すとともに動的再構成可能システムの有用性について考察する。3章では、パラメータ付けされた動的再構成可能システムを提案する。4章では3章で提案したシステムにて制約コスト内でアプリケーションを最速処理するデバイスとそのシステム構成を決定する手法を提案する。5章では、3章で設計したパラメータ付けされた動的再構成可能システムを、画像符号化処理JPEGおよびバケット処理に適用し、その結果を報告する。6章では、本稿を総括する。

2 動的再構成可能システム

動的再構成可能システムとはシステムの一部を動作中に再構成可能なシステムである。既にシステム上に実装されている回路の一部をシステムの動作中に別の回路に書き換えることにより仮想的に回路規模を増大できる。

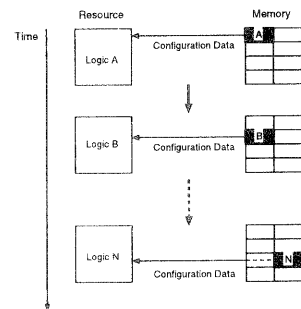


図1. 動的再構成による仮想的論理規模の増大。

例えば、図1に示すように、ある資源に論理Aが割り当てられているとする。論理Aを実行し終えた資源を再構成し、論理Bを割り当てる。ある瞬間における論理規模は資源の物理的な論理規模に等しいが、この手順を繰り返すことにより、資源はメモリの容量分だけの論理規模を実現可能となる。

以上のように動的再構成可能システムは、仮想的に論理規模の増大が可能である。しかし、(1)仮想的な論理規模の増大には再構成時間というオーバーヘッドが特殊な場合を除き必ず生じてしまう。また、(2)FPGAのような再構成可能デバイスの動作周波数は近年著しく高速化を達成してきているが、依然として専用

ハードウェアと比較すると遅い。以上の理由により動的再構成可能システムが専用ハードウェアより高速になることはない。また、動的再構成システムの資源と専用ハードウェアの資源では、チップの面積で比較すると専用ハードウェアの方が搭載可能なゲート数は多い。動的再構成可能システムで仮想的に回路規模の増大を用いても困難である。そのため動的再構成可能システムの有用性はソフトウェアよりも高速処理が可能であり、それをユーザの手元で容易に実現可能であることにある。本システムは提案手法によりユーザの設定するコスト内で最速処理をするシステムを決定する事が可能である。そして、CAD ツールを用いユーザの手元でシステムの実現が可能である。

3 パラメータ付けされた動的再構成可能システム

本章では、デジタル信号処理アルゴリズムの高速実行を目的としたパラメータ付けされたスケラビリティのある動的再構成可能システムを提案する。スケラビリティとは一般的には、コンピュータネットワーク、もしくは、コンピュータシステムの規模が拡大可能であるという意味で用いられる。本研究で用いられるスケラビリティとは本システムがアプリケーションの変更等に対し汎用性を確保するためにメモリ容量、システムバスのビット幅などが可変であり、また、アプリケーション変更時に冗長もしくは不足する機能が生じないためにシステム上のデバイス性能やデバイス間の接続方式も可変であることを指す。システム上のデバイス性能はその性能に応じてコストを設定しており、制約コスト内で最速な処理をするデバイスの決定が可能である。

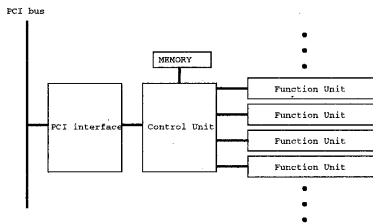


図 2. スケラビリティのある動的再構成可能システム

システムの概要を図2に示す。システムは、ホスト PC との通信を制御する PCI インターフェース、ホスト PC からのアプリケーションデータやコンフィギュレーションデータを保持するメモリ、Operation Unit や途中演算結果を保持するメモリから成る Function Unit、それらを制御する Control Unit で構成される。このうち、Operation Unit はアプリケーションの演算を行う。Operation Unit は演算用デバイスとして FPGA などの再構成可能なデバイスとローカルメモリ、ライブラリボードから成る。ローカルメモリとライブラリボードの有無は適用するアプリケーションにより可変である。本システムの演算用デバイスの再構成方法は A、B の 2 種類ある。A、B それぞれ再構成時間、再構成の自由度 (複数のデバイスを同時に再構成可能) に関し、A が再構成時間が早く自由度が低い、B は再構成時間は遅いが自由度が高い。

本システムは、以下の特徴を持つ。

システムのパラメータ化によるスケラビリティの実現 文献 [5] では、汎用性を確保するためにアプリケーション変更時のシステム拡張を可能にしており演算ユニット数、メモリ容量、システムバスの本数、各システムバスのビット幅が可変であった。これらに加え、本システムではシステム

上の再構成可能デバイスの最大周波数、回路規模、再構成時間、I/O ピン数やデバイス間の接続方式なども可変なパラメータとしアプリケーション変更に対し冗長、もしくは不足する機能が生じない。また、デバイスの性能に関する各パラメータ (最大周波数、回路規模、再構成時間、I/O ピン数) はそれぞれの性能に応じてコストを割り当て制約コスト内での最速処理のデバイスの決定が可能となっている。

2 種類の動的再構成による仮想的論理規模の増大 動的再構成により、システムに実装可能な論理規模を実論理規模以上に見せることが可能である。文献 [6] では、論理構成データを EPROM に保持しているため、実現可能な論理規模はシステム上の EPROM の容量に決定される。文献 [5] では、論理構成データはホスト PC から転送し SRAM に保持するため、実現可能な論理規模はシステム上のメモリ容量に左右されないが、論理構成データを一つの共有のメモリに保持するために複数の FPGA を同時に再構成を行うことが出来ない。本システムではアプリケーションにより上記 2 つの再構成法を選択することができ、再構成方法にも柔軟性を持つ。

既存 FPGA と研究段階のデバイスの採用 演算用デバイスに既存の FPGA のみを採用すればシステムの高い実現性が確保できるが、ユーザの満足できる結果を得るのは現実的には困難である。一方研究段階のデバイスでは満足のいく結果を得やすいがシステムの実現が困難になる。本システムはシステムに採用するデバイスとして既存の FPGA から研究段階のデバイスまでを用意しており上記に述べたようなトレードオフをユーザが選択できる。

ホスト計算機との親和性 本システムは、ホスト計算機の PCI バスに接続され PCI デバイスとして動作する。システムは、ホスト計算機と PCI バス経由で接続されるため、高いデータ転送能力を確保している。システムは、PCI デバイスとして動作するため、ホスト計算機上の OS に自動で認識され、GUI ソフトウェアによる対話的なシステム制御が可能である。

以下、Function Unit、Operation Unit、Control Unit、再構成可能デバイスについて説明する。

3.1 Function Unit

3.1.1 Function Unit の構成

Function Unit は以下のいずれかの構成である。

1. Function Unit は 1 つの Operation Unit で構成される。
2. Function Unit は M 個 (M は 2 以上の整数) の Operation Unit と N 個 (N は 0 または 1) のメモリがバスに並列接続されている構成で、それぞれの Operation Unit 内の再構成可能デバイスへ再構成の制御用の特殊ピンが Control Unit へダイレクトに接続されている。
3. Function Unit は M 個 (M は 2 以上の整数) の Operation Unit と N 個 (N は 0 または 1) のメモリがバスに並列接続されている構成で、再構成の制御用の特殊ピンもバスを介して Operation Unit 内の再構成可能デバイスへ接続されている。
4. Function Unit は 1 つのメインメモリで構成される。

図 3、図 4、図 5 に Function Unit として構成 1 が 3 つと構成 4 が 1 つ、構成 2 が 1 つ (M = 2, N = 1)、構成 3 が 1 つ (M = 2, N = 1) としたときのシステムの構成を示した。

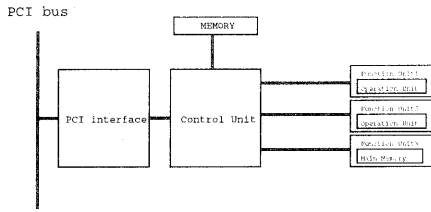


図 3. 構成 1 が 3 つと構成 4 が 1 つ

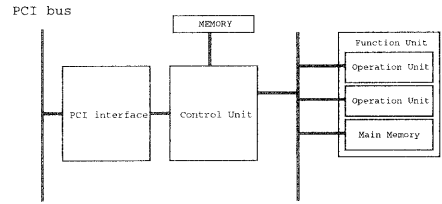


図 5. 構成 3 が 1 つ $M = 2, N = 1$

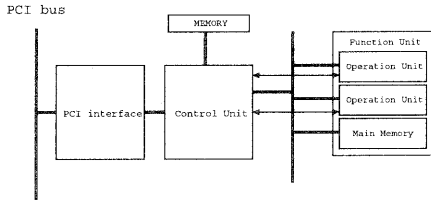


図 4. 構成 2 が 1 つ $M = 2, N = 1$

構成 1 では Operation Unit は、アプリケーションの実行に必要な十分な本数のピンとピン、および再構成に必要な特殊ピンが Control Unit へ直接接続されている。構成 2, 3 では Operation Unit と Control Unit はデータバス、アドレスバス、アウトプットイネーブル、ライト、およびチップセレクト線が共通のバスを介して接続されている。構成 2 では再構成の特殊ピンはバスとは別に両ユニット間で直接接続されている。構成 3 では再構成の特殊ピンも共通のバスを介して接続されている。データバス幅 Dat ビット、アドレスバス幅 Add ビットはアプリケーションに応じて決定される。また、チップセレクトのビット幅 CS ビットはバスに並列に接続されるデバイスの個数により決定される。デバイス数を N としたとき $CS = \log_2 N$ ビットとなる。

メインメモリの有無はアプリケーションに応じて決定される。

3.2 Operation Unit

Operation Unit の構成要素は演算用デバイスとローカルメモリ、ライブラリボードである。ローカルメモリはアプリケーションのスケジューリング結果に依存して、ライブラリボードは動的再構成の方法により不要となることがある。

3.2.1 演算用デバイス

本システムの演算用デバイスとしては動的な再構成が可能なデバイス (FPGA など) を用いる。演算用デバイスはスケジューラが割り当てた演算を行う。

3.2.2 ローカルメモリ

ローカルメモリは汎用 SRAM にて構成される。ローカルメモリは演算用デバイスのレジスタで保持しきれない演算の途中結果を保持する。

3.2.3 ライブラリボード

ライブラリボードには B コンフィギュレーションが選択された時に演算用デバイスのコンフィギュレーションデータを保持しており、Control Unit からの信号により選択されたライブラリ

ボード上の ROM が演算用デバイスに接続される。ライブラリボードの概要を図 6 に示す。

3.2.4 動的再構成

本システムの演算用デバイスの再構成方法は 2 種類ある。A コンフィギュレーションでは高速に書き換え可能であるがコンフィギュレーションデータを Control Unit に接続されている 1 つのメモリ (図 2 中の MEMORY) に保持するため、複数のデバイスの並列書き換え不能である。B コンフィギュレーションは低速な書き換え時間であるが、コンフィギュレーションデータを Operation Unit にあるライブラリボードに保持するため複数のデバイスの並列書き換え可能である。

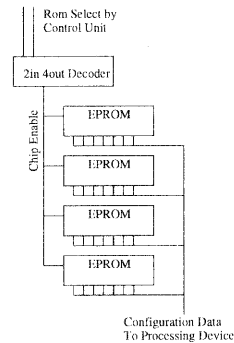


図 6. ライブラリボード

3.3 Control Unit

Control Unit の機能としてデータ転送の制御と動的再構成の制御が挙げられる。

3.3.1 データの制御

Control Unit は Operation Unit やメインメモリのデータ転送を制御する。Control Unit はデータバス、アドレスバス、アウトプットイネーブル、ライトの信号線を持ち、Operation Unit に複数のデバイスが接続されている場合にはそれらのデバイスを選択するチップセレクトを持つ。

Control Unit は事前に行われたスケジューリングによって得られたタイミングでデータ転送を行う。チップセレクトにより目的のデバイスを選択しアウトプットイネーブルを High にすれば所望のデバイスよりデータバスにデータが出力されライトを High にすればデータが入力される。

3.3.2 動的再構成の制御

Control Unit は Operation Unit の動的再構成を制御する。ここでは、再構成可能デバイスとして XILINX の XC4000 シリーズを用いた時の動的再構成に関する特殊ピンの制御について説明する。

XILINX の XC4000 シリーズでは、A,B コンフィギュレーションはそれぞれ A がスレープモード B がマスタモードに相当する。XC4000 シリーズでは動的再構成用の特殊ピンとしてマスタモードの際は PROGRAM, HDC, スレープモードの際は PROGRAM, CCLK, DIN, DONE がある。

Control Unit はデータ転送の制御と同様、事前のスケジューリングによって得られたタイミングで動的再構成を制御する。システムの再構成はマスタ、スレープいずれのモードの時でも PROGRAM ピンを 300ns 以上 High にすることにより開始される。そしてマスタモードではコンフィギュレーション中は HDC ピンが High を維持する為、HDC ピンをコントロールユニットが監視する。また、スレープモードでは DONE がコンフィギュレーションの終了を通知する。また、マスタモードではライブラリボード上のいずれかの ROM を選択するための ROM セレクト線が Operation Unit 内のライブラリボードに接続される。

3.4 演算用デバイスのパラメータとシステムのコスト

本システムに用いる演算用デバイスのパラメータは最大周波数 F_{lg} , 集積度 S_{ga} , 再構成時間 T_r , 入出力ピン数 N_p であり、それぞれの取り得る範囲を以下のようにする。

$$\begin{aligned} F_{lg} &: 40\text{M}\sim 200\text{M(Hz)} \\ S_{ga} &: 5\text{K}\sim 200\text{K(Gate)} \\ T_r &: 5\text{n}\sim 20\text{m (s)} \\ N_p &: 0.006S_{ga}\sim 0.015S_{ga} \end{aligned}$$

これらの値に表 1 のようなコストを与える。そして、表 1 中の各パラメータのコストを $C(F_{lg})$, $C(S_{ga})$, $C(T_r)$, $C(N_p)$ とし、デバイスのコスト C を

$$C = \alpha C(F_{lg}) + \beta C(S_{ga}) + \gamma C(T_r) + \omega C(N_p) \quad (1)$$

$$\alpha + \beta + \gamma + \omega = 1 \quad (2)$$

で与える。 α , β , γ , ω は適用する再構成可能デバイスの特性にあわせて与える。

システムのコストは各デバイスのコストの総和である。

4 システム構成の決定

本章では、提案したシステムにユーザが制約コストを設け制約コスト内での最速処理をするシステム構成を決定する手法を提案する。この手法はアプリケーションと制約コストを入力とし、デバイスの各パラメータとシステム構成が出力される。

デバイスの各パラメータとシステム構成の決定は以下の手順で行う。

Step 1 スケジューリング

Step 2 Operation Unit の構成の決定

Step 3 Function Unit の構成の決定

4.1 スケジューリング (Step1)

アプリケーションのスケジューリングをすることによりデバイスのパラメータが決定される。

スケジューリング手法は本研究の共同研究である文献 [8] の手法を用いる¹。文献 [8] のスケジューリング手法は、アプリケーションの各タスクの処理時間を入力としてアプリケーションのスケジューリング結果を出力する。本システムでは、デバイスのパラメータとしてデバイスの最大周波数が与えられている。スケジューラの入力となる各タスクの処理時間は、デバイスの最大周波数から次のように算出できる。

最大周波数と処理時間

デバイスの最大周波数と処理時間の関係を導出する。

サンプル回路として 8 ビットカウンタと離散コサインモジュールを異なる最大周波数のデバイス上にて配置配線した結果を表 2 に示す。配置・配線ツールは XILINX の XACT、ターゲットデバイスは XC4000 シリーズである。

表 2. 最大周波数とクロックサイクル

最大周波数 (MHz)	111	125	166
カウンタ (ns)	21.9	16.9	11.1
DCT (ns)	214.7	164.9	119.1

表 2 より、8 ビットカウンタの最大周波数とクロックサイクルの関係性を、最大周波数を F_{lg} とクロックサイクルを T_{cycle} とし

$$T_{cycle} = a / (F_{lg})^n + b \quad (3)$$

と仮定する。

ここで、 $n = 3/2$ で、 $a = 27420$, $b = -1.72$ となる。表 2 より、DCT のクロックサイクルはカウンタ回路のおよそ 10 倍であり、他のモジュールでも定数倍になっていると考えられる。したがって最大周波数 F_{lg} とクロックサイクル T_{cycle} の関係は、比例定数を α とし、

$$T_{cycle} = \alpha (27420 / (F_{lg})^{3/2} - 1.72) \quad (4)$$

となる。

最大周波数とアプリケーションのタスクの処理時間は、あるタスクを最大周波数 $F_{c_{lg}}$ のデバイスで配置・配線を行いそのタスクの処理時間が $T_{F_{c_{lg}}}$ で与えられた時、任意の最大周波数 F_{lg} でそのタスクの処理時間 $T_{F_{lg}}$ は、次式で与えられる。

$$T_{F_{lg}} = T_{F_{c_{lg}}} \times \frac{T_{cycle}}{T_{cycle_{F_{lg}=F_{c_{lg}}}}} \quad (5)$$

$T_{F_{lg}}$ を用いてスケジューリングをする。

スケジューラによりユーザの設定したコスト内での最速処理するデバイスの各パラメータ最大周波数 F_{lg} , 集積度 S_{ga} , 再構成時間 T_r , 入出力ピン数 N_p と演算用デバイスの個数 N_a , 再構成方法 r (0: A コンフィギュレーション, 1: B コンフィギュレーション) を決定する。

4.2 Operation Unit の構成の決定 (Step2)

Operation Unit 内のローカルメモリ、ライブラリボードの有無を決定する。ローカルメモリはアプリケーションの途中演算結果を演算用デバイスのレジスタのみで保持しきれない、もしくは、スケジューリングにて演算用デバイスが動的再構成するようなスケジューリング結果のときに必要である。また、ライブラリボードは再構成方法が B コンフィギュレーションのとき必要である。

¹文献 [8] では、デバイスのパラメータやコストを含んだスケジューリングではないが今後本システム向けに拡張予定である。

表 1. パラメータとコスト

コスト	10	20	30	40	50	60	70	80	90	100
F_{iq} (MHz)	40	45	50	60	70	80	100	120	150	200
S_{ga} (Kgate)	5	10	15	20	30	50	80	120	160	200
T_p (s)	20 μ	5m	600 μ	150 μ	20 μ	5 μ	200n	80n	20n	5n
N_p	0.006 S_{ga}	0.007 S_{ga}	0.008 S_{ga}	0.009 S_{ga}	0.010 S_{ga}	0.011 S_{ga}	0.012 S_{ga}	0.013 S_{ga}	0.014 S_{ga}	0.015 S_{ga}

4.3 Function Unit の構成の決定 (Step3)

スケジューリングにより Operation Unit の個数が N_a と出力された場合、FunctionUnitは、(1) 図3のような構成1が N_a 個ある構成となる、(2) 図4のような構成2となる、(3) 図5のような構成3となる、のいずれかである。

(1), (2), (3) の各構成において (1) は、Operation Unit がそれぞれ独立に Contorol Unit に接続されている一方で (2), (3) ではバスを介して接続されている。そのため (1) の Contol Unit は (2), (3) の Control Unit に必要なバスコントローラの実装が不要である。また、(1) は複数の Operation Unit に同時にアクセス可能である。以上実装の容易さとアクセスの自由度の高さの点で、Function Unit の構成は、(1) 図3のような構成1が N_a 個ある構成が優先される。しかし、(1) は Operation Unit がそれぞれ独立に Contorol Unit に接続されているため、Operation Unit で多ビット演算をするには、Contorol Unit のピン数が不足してしまう。その時に、(2) 図4のような構成2となる。このとき、Function Unit 内の OperationUnit の個数 M が N_a となる。また、(2) は、再構成用の特殊ピンはバスを介さずに直接接続されているが特殊ピンは複数本あり、Operation Unit が多数になると Contorol Unit のピン数が不足することが考えられる。その時は、(3) 図5のような構成3となる。このとき、Function Unit 内の OperationUnit の個数 M は N_a である。

5 システム応用

本章では、提案システムを JPEG エンコーダとパケット処理に適用した結果を報告する。JPEG エンコーダとパケット処理ともにスケジューリングは手動で行った。

5.1 JPEG エンコーダ

JPEG はカラー静止画像符号化標準を目指して作られた画像方式である。JPEG は複数の動作モードを持つが、本稿では JPEG の最も一般的な構成である DCT (Discrete Cosine Transform: 離散コサイン変換) とハフマン符号化を用いた方式により JPEG エンコーダを実現する。設計した JPEG エンコーダは RGB 画像データを入力とし、JPEG 符号化された圧縮画像データを出力とする。以下の流れによって符号化される。

1. 前処理 (RGB \rightarrow YUV 変換, 色差間引き)
入力された RGB 信号を輝度と色差成分 (YUV 信号) に変換する。変換された画素値はそれぞれ 8×8 毎のブロック集合に分けられ DCT に送られる。
2. DCT
2D-DCT (2次元離散コサイン変換) により空間周波数に変換する。
3. 量子化
2D-DCT からの出力を量子化する。量子化テーブルを用い係数位置ごとに異なったステップサイズで線形量子化する。
4. ハフマン符号化
ハフマン符号化を行い、JPEG 圧縮されたビットストリームを 8bit 毎に出力する。

5.2 JPEG エンコーダの適用

表3に JPEG エンコーダの各タスクのゲート数と処理時間を示した。表3は、XILINX の配置・配線ツールである XACT を用いて XC4020E-1 に配置・配線した結果である。表3より、JPEG

表 3. JPEG エンコーダの各処理の値

	RGB2YUV	2D-DCT	Quantum	Huffman
ゲート数 (Gate)	13400	17600	600	1600
8×8 画素の処理時間 (μ s)	10	133	20	26
320×240 画素の処理時間 (ms)	12	160	24	31

エンコーダの各タスクの最大ゲート数は 2 次元 DCT の 17600 ゲートである。17600 ゲートのタスクが搭載可能な最も小さいデバイスは表1より 2 万ゲートのデバイスであるため、最低でも 2 万ゲートのデバイスは 1 つは必要となる。また、すべてのタスクの合計ゲート数は $(13400+17600+600+1600 =)$ 33200 ゲートである。33200 ゲートのタスクを搭載するには (1) 5 万ゲートのデバイスを 1 つ用いる、(2) 2 万ゲートのデバイスを 2 つ用いる、(3) 2 万ゲートのデバイスを 1 つ用いて動的再構成をすることにより仮想的に論理規模を増大させる、の 3 通りがある。(1),(2) は動的再構成の適用なし、(3) は動的再構成の適用ありである。(1),(2) 動的再構成を適用しない場合

動的再構成を適用しない場合、パイプライン処理の適用が可能である。これは動的再構成は不要になった資源を新たな資源に書き換えるのに対しパイプライン処理では連続的に全ての資源を必要とするためである。したがって、JPEG エンコーダの全処理時間は処理時間の最も長い 2 次元 DCT の処理時間に依存する。XC4020E-1 の最大周波数は文献 [17] より 166MHz で与えられる。よってパイプライン処理をした際の処理時間は

$$T_{all} = 160 \times \frac{T_{cycle}}{T_{cycle} F_{iq} = 166} \quad (6)$$

であたえられる。

また、用いるデバイスの個数は各タスクの合計ゲート数が 33200 ゲートであることから、(1) 5 万ゲートのデバイスを 1 つ用いる方法と (2) 2 万ゲートのデバイスを 2 つ用いることが考えられるが表1より (1) の方がコストが低いので 5 万ゲートのデバイスを 1 つ用いる。そのときのコスト C は

$$C = \alpha C(F_{iq}) + \beta \cdot 60 \quad (7)$$

となる。

(3) 動的再構成を適用する場合

動的再構成を適用する場合には、JPEG エンコーダの全てのタスクにまたがるパイプライン処理はできないが、動的再構成の前後ではパイプライン処理が可能である。RGB2YUV のゲート数が 13400 ゲート、それ以降の処理の合計が 19800 ゲートであるので、JPEG エンコーダのスケジューリングは 2 万ゲートのデバイスを 1 つ用いて RGB2YUV の後に動的再構成を行うものとし、その再構成の後はパイプライン処理で 2 次元 DCT、量子化、ハフマン符号化を行うものとする。その場合の全体処理時間 T_{all} は

$$T_{all} = (12 + 160) \times \frac{T_{cycle}}{T_{cycle} F_{iq} = 166} + T_r \quad (8)$$

であたえられる。その時のコスト C は

$$C = \alpha C(F_{lg}) + \beta \cdot 40 + \gamma C(T_r) \quad (9)$$

となる。

ここで、 $\alpha = 0.3$, $\beta = 0.3$, $\gamma = 0.3$, $\omega = 0.1$ としたときの全体処理時間 T_{all} と全コスト C を表 4, 表 5, 表 6 に示した。

表 4. 動的再構成をしない時の全処理時間 T_{all} とコスト C

F_{lg}	40	45	50	60	70	80	100	120	150	200
$C(F_{lg})$	10	20	30	40	50	60	70	80	90	100
$T_{all}(ms)$	1537	1284	1093	826	650	527	370	275	190	114
C	21	24	27	30	33	36	39	42	45	48

表 5. 動的再構成をした時の全処理時間 $T_{all}(ms)$

T_r	最大周波数 F_{lg}									
	40	45	50	60	70	80	100	120	150	200
20	1672	1400	1195	907	718	587	418	316	224	143
5	1657	1385	1180	892	703	572	403	301	209	128
0.6	1653	1381	1175	888	699	567	398	297	205	124

表 6. 動的再構成をした時のコスト C

T_r	最大周波数 F_{lg}									
	40	45	50	60	70	80	100	120	150	200
20	18	21	24	27	30	33	36	39	42	45
5	21	24	27	30	33	36	39	42	45	48
0.6	24	27	30	33	36	39	42	45	48	51

ここで、仮に制約コストが 40 と与えられているとする。制約コスト内で最も高速なスケジューリング結果は、表 4, 表 5, 表 6 より動的再構成ありで、 $F_{lg} = 120\text{MHz}$, $T_r = 20\text{ms}$ のときであり、最速処理時間として 316ms が選択される ($C = 39$)。

次に $C(N_p) = 10$ とすると $N_p = 300$, $C = 40$ となり、制約コストを満たす。再構成は高速の A コンフィギュレーションとなる。Opertion Unit の構成は再構成可能デバイスに途中演算結果保持用のローカルメモリが接続されたものである。また、A コンフィギュレーションのためライブラリボードは不要である。 $N_p = 300$ なので Function Unit の構成は構成 3 の Operation Unit が 1 つ接続されているものとなる。

このときのシステム構成を図 7 に示す。

以上、JPEG エンコーダをシステムに適用した。

5.3 バケット処理

バケット処理のアプリケーションとして図 8 のプロトコルプスタを用いたネットワークを考える。プロトコルプスタはネットワーク上のノードに機能を付加し、必要に応じてバケットに対して処理を施すプログラムであり、ネットワークの経路上のプスタとデ・プスタによって実現される。通常、プスタは、プロトコルプスタ・マルチプレクサによってプスタが選択されるが、プスタとデプスタの機能を動的再構成による仮想的回路規模の増大により実現することも可能である。

5.4 バケット処理の適用

プスタを暗号化アルゴリズム、デプスタを復号化アルゴリズムとしてシステムに適用する。データ暗号化標準 (DES) の暗号化アルゴリズムのゲート数と処理時間を表 7 に示す。復号アルゴリズムのゲート数と処理時間は表 7 の暗号アルゴリズムの値を用いる。表 7 は、XILINX の配置・配線ツールである XACT を用いて XC4020E-1 に配置配線した結果である²。

²XC4020 は 2 万ゲート相当であるが暗号化 3 段分が 19200 ゲートであったことから表 7 を算出した

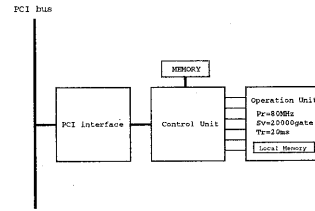


図 7. JPEG をシステムに適用した図

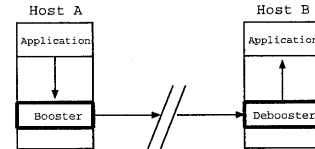


図 8. プロトコルプスタを用いたネットワーク

暗号化・復号化アルゴリズムの各タスクが 106000 ゲートであることから (1) 動的再構成を用いずに 12 万ゲートのデバイスを 2 つ用いる、(2) 動的再構成を適用して 12 万ゲートのデバイスを 1 つ用いる、の 2 通りが考えられる。プロトコルプスタを用いたアプリケーションでは次にプスタするのかデプスタとするのかは判らないが、ここでは、暗号化と復号化を交互に 1 対 1 の割合で行うものとし、暗号化と復号化の処理時間の合計で評価をする。

(1) 動的再構成を適用しない場合

動的再構成をしない場合の暗号化と復号化の処理時間の合計は、

$$T_{all} = (900 \times 2) \times \frac{T_{cycle}}{T_{cycle} F_{lg=166}} \quad (10)$$

デバイスのコスト C は

$$C = \alpha C(F_{lg}) + \beta \cdot 80 \quad (11)$$

となる。

(2) 動的再構成を適用する場合

動的再構成をする場合の暗号化と復号化の処理時間の合計は、

$$T_{all} = (900 \times 2) \times \frac{T_{cycle}}{T_{cycle} F_{lg=166}} + 2 \cdot T_r \quad (12)$$

であたえられる。

デバイスのコスト C は

$$C = \alpha C(F_{lg}) + \beta \cdot 80 + \gamma C(T_r) \quad (13)$$

となる。

ここで、 $\alpha = 0.3$, $\beta = 0.3$, $\gamma = 0.3$, $\omega = 0.1$ としたときの全体処理時間 T_{all} と全コスト C を表 8, 表 9, 表 10 に示した。

ここで、仮に制約コストが 64 と与えられているとする。制約コスト内で最も高速なスケジューリング結果は、表 8, 表 9, 表 10 より、動的再構成ありで、 $F_{lg} = 80\text{MHz}$, $T_r = 200\text{ms}$ のときであり、最速処理時間として $6.3\mu\text{s}$ が選択される ($C = 63$)。以下は JPEG の時と同様にシステムが決定する。

以上、プロトコルプスタを用いたバケット処理をシステムに適用した。

表 7. 暗号化・復号化アルゴリズムのゲート数と処理時間

DES 暗号化アルゴリズム	
ゲート数 (Gate)	106000
処理時間 (ns)	900

表 8. 動的再構成をしない時の全処理時間 T_{all} とコスト C

F_{Lg}	45	50	60	70	80	100	120	150	200
$C(F_{Lg})$	20	30	40	50	60	70	80	90	100
$T_{all}(\mu s)$	14.4	12.2	9.2	7.3	5.9	4.1	3.1	2.1	1.2
C	54	57	60	63	66	69	72	75	78

表 9. 動的再構成をした時の全処理時間 $T_{all}(\mu s)$

T_r	最大周波数 F_{Lg}								
	45	50	60	70	80	100	120	150	200
5000	24.4	22.2	19.2	17.3	15.9	14.1	13.1	12.1	11.2
200	11.4	12.6	9.6	7.7	6.3	4.5	3.5	2.5	1.6
80	14.6	12.4	9.4	7.4	6.0	4.3	3.2	2.3	1.4
20	14.4	12.3	9.3	7.3	5.9	4.2	3.1	2.1	1.3

6 むすび

本稿では、大規模なデジタル信号処理アプリケーションの高速実行を目的としたパラメータ付けされたスケーラビリティのある動的再構成可能システムを提案した。また、制約コストを設定し制約コスト内で最速に処理するデバイスを決定する手法を説明し、そのアプリケーションとして JPEG とパケット処理を適用した例を述べた。本システムは適用するアプリケーションに応じシステムの構成、デバイスの構成が変化するものが最大の特長である。なお、パラメータの設定により文献 [5], [6] の動的再構成可能システムと同様の構成が可能である。

現在、アプリケーションのスケジューリングは手動で行っているが文献 [8] のスケジューリングアルゴリズムを拡張することによりスケジューリングの自動化を目標としている。また、パラメータとコストの関係が変わることにより結果が変わり得るので現実のデバイスでパラメータとコストの相関関係を導き出すのが今後の課題である。

謝辞

本研究に関し、有用な議論、討論を頂いた本学修士課程 2 年石飛貴志氏、修士課程 1 年陳曉梅氏、に感謝致します。本研究の一部は、文部省科学研究費補助金（基盤研究 C(2)、課題番号 10650345 および奨励研究 (A)、課題番号 10750369）の援助を受けた。

参考文献

- [1] M. Butts, "Future directions of dynamically reprogrammable systems," in *Proc. IEEE 1995 CICC*, pp. 487-494, 1995.
- [2] A. Chowdhary and J. P. Hayes, "Technology mapping for field-programmable gate arrays using integer programming," in *Proc. IEEE 1995 ICCAD*, pp. 346-352, 1995.
- [3] J. G. Eldredge, and B. L. Hutchings, "Density enhancement of a neural network using FPGAs and run-time reconfiguration," in *Proc. IEEE Workshop on FPGAs for Custom Computing Machines*, pp. 180-188, 1994.
- [4] T. Fujii, K. Furuta, M. Motomura, M. Nomura, M. Mizuno, K. Anjo, K. Wakabayashi, Y. Hirota, Y. Nakazawa, H. Ito and M. Yamashita, "A dynamically reconfigurable logic engine with a multi-context/multi-mode unified-cell architecture," in *ISSCC Digest of Technical Papers*, 21.3, 1999.
- [5] 羽切 崇, 戸川望, 柳澤政生, 大附辰夫, "FPGA を用いた動的再構成可能システムと暗号化アルゴリズムへの応用," 信学技報, VLD99-109, pp. 7-14. 2000.
- [6] 長谷川洋平, 戸川望, 柳澤政生, 大附辰夫, "FPGA を用いた動的再構成可能システムとその応用," 信学技報, VLD98-143, pp. 17-24,

表 10. 動的再構成をした時のコスト C (パケット処理)

T_r	最大周波数 F_{Lg}								
	45	50	60	70	80	100	120	150	200
5000	48	51	54	57	60	63	66	69	72
200	51	54	57	60	63	66	69	72	75
80	54	57	60	63	66	69	72	75	78
80	57	60	63	66	69	72	75	78	81

1999.

- [7] D. L. How, "A self clocked FPGA for general purpose logic emulation," in *Proc. IEEE 1996 CICC*, pp. 148-151, 1996.
- [8] 石飛 貴志, 戸川望, 柳澤政生, 大附辰夫, "FPGA を用いた動的再構成可能システムを対象とするスケジューリング手法," 信学技報, VLD, 2001.
- [9] D. Jones and D. M. Lewis, "A time-multiplexed FPGA architecture for logic emulation," in *Proc. IEEE 1995 CICC*, pp. 495-498, 1995.
- [10] Y. S. Lee and A. C. H. Wu, "A performance and routability driven router for FPGAs considering path delays," in *Proc. 32nd DAC*, pp. 557- 561, 1995.
- [11] X. P. Ling and H. Amano, "WASMII: A data driven computer on a virtual hardware," in *Proc. of the IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 33-42, 1993.
- [12] W. Luk, N. Shirazi, and P. Y. K. Cheung, "Modeling and optimizing run-time reconfiguration systems," in *Proc. of IEEE Workshop on FPGAs for Custom Computing Machine*, pp. 167-176, 1996.
- [13] K. Nagami, K. Oguri, T. Shiozawa, H. Ito and R. Konishi, "Plastic cell architecture: Towards reconfigurable computing for general-purpose," in *FCCM'98*, pp. 68-77, 1998.
- [14] J. H. Novak and E. Brunvand, "Using FPGAs to prototype a self-timed floating point co-processor," in *Proc. IEEE 1994 CICC*, pp. 85-88, 1994.
- [15] P. Sawkar and D. Thomas, "Performance directed technology mapping for look-up table based FPGAs," in *Proc. 30th DAC*, pp. 208-212, 1993.
- [16] Xilinx, *XC6200 Field Programmable Gate Arrays*, 1997.
- [17] Xilinx, *プログラマブル・ロジック・データブック*, 1997.