

配線リソースを考慮した再構成可能 1bit プロセッサアレイ

中井伸郎 中西正樹 山下茂 渡邊勝正

奈良先端科学技術大学院大学情報科学研究科

〒 630-0192 奈良県生駒市高山町 8916-5

E-mail: {nobuo-n, m-naka, ger, watanabe}@is.naist.jp

あらまし 半導体メーカは、年々増加する回路規模と短い製品サイクルによるコストの増加を如何に抑えるかという課題を抱えている。その一解決法として再構成可能ハードウェアの利用が挙げられる。しかし再構成可能ハードウェアは様々な課題を抱えており、結果として ASIC やソフトウェアよりも高コストとなる場合が多い。本稿では低コストで効率の良い再構成可能ハードウェアアーキテクチャを提案する。本アーキテクチャでは、面積の大半を占める配線領域を抑えながらも柔軟に配線できるバス構造、プロセッサエレメント数を容易に拡大できるスケラビリティのある構成をとっている。提案するアーキテクチャのアプリケーションとして DCT 演算を行った結果について報告する。

キーワード リコンフィギャラブルコンピューティング, 粗粒度アーキテクチャ, ビットシリアルデータバス, 配線リソース

Reconfigurable 1-bit processor array with reduced wiring area

Nobuo NAKAI Masaki NAKANISHI Shigeru YAMASHITA Katsumasa WATANABE

Nara Institute of Science and Technology 8916-5 Takayama-chou, Ikoma-shi, Nara, 630-0192, Japan

E-mail: {nobuo-n, m-naka, ger, watanabe}@is.naist.jp

Abstract Semiconductor makers have a problem of how to reduce the production cost. Because of the increasing gates to implement and shortening production cycle, production cost is increasing. One of the way to solve this problem is to use of reconfigurable hardwares. Although reconfigurable hardwares seemed to be useful, they have some disadvantages. As a result, a system using software or ASIC costs lower than reconfigurable hardware in many cases. In this paper we propose an efficient architecture of reconfigurable hardware with low cost. The proposed architecture has the following features; It has high routability but wiring area is reduced, and number of processor elements can be increase easily. We mapped DCT circuit to proposed architecture and run. We also show some experimental results.

Keyword reconfigurable computing, coarse-grain architecture, bit-serial data path, wiring resource

1. はじめに

製品に組み込まれる LSI は年々高速化、高機能が望まれている。そのため実装する回路規模が増加し製品完成までに多大な工数と時間を要するようになった。しかしプロトコルやフォーマット、そしてユーザニーズの多様化により製品の寿命が短くなるため、製造コストも増加の一途を辿っている。そのため半導体メーカは如何にしてコストを下げるかという課題に悩まされている。

課題解決の一手法として再構成可能ハードウェアを利用することが挙げられる。再構成可能ハードウェアを使用した場合、チップ試作が不要なた

め開発コストの削減が可能である。また複数の製品が同一の再構成可能ハードウェアを用いる場合、製造工程を統一できるため製造コストの削減も可能である。実際に一部の機器では細粒度型アーキテクチャである FPGA が利用されていたり、ファイヤウォール装置に粗粒度型アーキテクチャである DAPDNA-2 が利用されている例もある [1]。

しかし再構成可能ハードウェアが一部でしか利用されない理由として、再構成可能ハードウェアが ASIC を利用するよりも高コストな場合が多いことが挙げられる。具体的には a) 細粒度型アーキテクチャではロジック密度が低く、動作速

度が遅い。b)粗粒度型アーキテクチャではアプリケーションとプロセッサエレメントのビット幅が合わないと無駄が生じる、細粒度型と比べてアプリケーションが限定されるからなどである。

本稿では低コストで効率の良い再構成可能アーキテクチャを提案する。具体的には面積効率が良く比較的多くのアプリケーションを処理できるものを目指している。提案するアーキテクチャ（以下、本アーキテクチャ）では、ロジック密度と動作速度の問題を改善するために粗粒度型アーキテクチャを採用している。またロジック密度に関しては、チップ面積の大半を占める配線リソースを軽減するため、シリアルバスを使用している。そして粗粒度型の問題であるビット幅の整合性に対しては、ビットシリアルにデータを処理する 1bit プロセッサエレメントを使用している。粗粒度型アーキテクチャのアプリケーションが限定される一因として、エレメント数が少なくその数が構造的に増やせないからということが挙げられる。本アーキテクチャはこの問題を解消するために、エレメント数を増やしやすいくスケラビリティのある構造をとっている。

以下、アーキテクチャを特徴付けるビットシリアルデータパスについて、関連研究について、提案するアーキテクチャについて、チップに実装した環境と結果を、アプリケーションとして DCT 演算を行った結果を、そして最後にまとめと今後の課題について述べる。

2. ビットシリアルデータパス

通常の CPU は 16bit で入力されたデータを 16bit パラレルのまま一括で処理を行う。それに対して 16bit のデータを LSB から MSB の順（または逆順）に 1bit ずつ 16 クロックかけて処理を行う方式をビットシリアル方式という。このようなデータパスをビットシリアルデータパスといい、ビットパラレルに比べると、1)スループットが低い 2)割り算回路の実装が困難。という欠点がある。しかし 1)回路規模を削減できる 2)高速化しやすい 3)配線が単純という利点もある。同一エレメントを大量に敷き詰める構成の再構成可能ハードウェアでは、これらの利点は特に有用であると考えられる。図 1 のように 3 段カスケード接続されたブロックに 1 ワード 4bit のデータを入力し、4 つの加算を行う場合を考える。

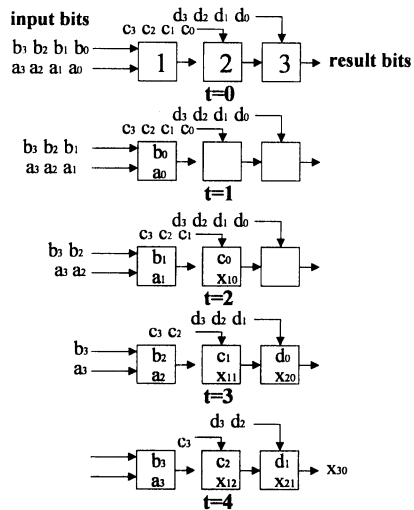


図 1: ビットシリアルデータパス

時刻 $t=1$ でブロック 1 が入力データの LSB を処理して結果 X_{10} を出力する。 $t=2$ ではブロック 1 は入力データ 2bit 目を、ブロック 2 は X_{10} ともう一方の入力 c_0 を処理する。同様な処理がブロック 3 でも行われ、最終的な結果が得られる。以降データを入力し続ければ、図の回路はパイプライン的に動作するため、4 クロックごとに 1 ワードの計算結果が得られる。

しかしビットシリアルで右シフト演算を行う場合、シフトする bit 数分だけデータを待つ必要があり、その分出力が遅れてしまう。そのような演算の出力を受け取って処理するブロックは、出力の遅延にあわせて同期を取る必要がある。提案するアーキテクチャでは、カウンタを用いた簡単な機構で同期を取れるよう設計している。

3. 関連研究

ビットシリアルデータパスに着目したアーキテクチャとして、FPVLSI [2] と Bit-Serial FPGA [3] が挙げられる。FPVLSI はビットシリアルにデータを処理するプロセッサエレメント (PE) を 2 次元に配置した構成を取る。そして PE 間の通信を 4 近傍に限定することで、FPGA で問題となる配線ボトルネックを解消している。

Bit-Serial FPGA は配線リソースは FPGA と同じであるが、ロジックブロックという 4 個の 4-ILUT で構成されるエレメントを利用している。ロジックブロックはビットシリアルデータパスを形成しやすい構造になっている。

提案するアーキテクチャは、FPVLSI のように PE

を2次元に配置した構成を取る。相異点は離れたPEと通信できるよう、しかし配線がボトルネックとならないような配線アーキテクチャを採用している点である。

4. アーキテクチャ

4.1. 概要

本アーキテクチャの概略図を図2に示す。本アーキテクチャは、ホストプロセッサ(HP)の一つのデバイスとして利用されることを想定している。

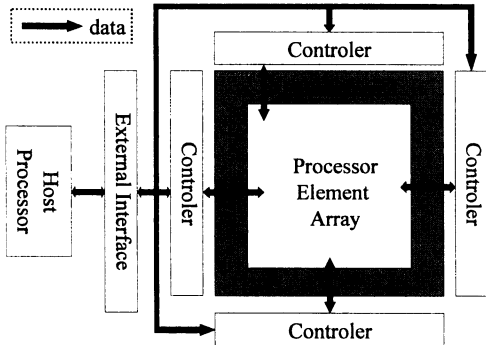


図2：アーキテクチャの概観

本アーキテクチャはコントローラ、プロセッサエレメント(PE)、そしてI/Oエレメント(IOE)で構成されている。コントローラはHPとのデータのやりとりやPE、IOEの制御を行う。PEは設定された命令に従って入力データを処理する。各PEに異なる命令を与えることで、PEアレイ全体としてあるアプリケーションを処理する回路を形成できる。IOEはPEとコントローラ間でデータを入力/出力するポートの役割を果たす。

PE、IOEは2次元状に配置されており、最外周にはIOE、それ以外はPEが配置されている。PE、IOEの元素数は、他の粗粒度型アーキテクチャとは異なり固定ではない。元素数にスケラビリティがある構造のため、面積制約を満たす限り、チップ実装時に設計者が自由に元素数を設定可能である。元素数にスケラビリティがあるのはコントローラやPEとのデータ通信をIOEに限定しているためである。また元素数以外にも4.2.で述べる配線のパラメータをチップ実装時に設定可能である。

本アーキテクチャはリセット後、HPからコントローラを通じて各PE、IOEにコンフィギュレーションデータが送られる。コンフィギュレーションデータには、各元素に実行させる命令、入力や出力セクタに与えるデータが含まれてい

る。コンフィギュレーション後、HPから処理すべきデータが入力されると、コントローラは入力ポートとして動作するIOEにデータを送る。IOEはPEにデータを送り、PEは与えられた命令に従ってデータを処理し、次のPEに結果を出力する。最終的にPEは出力ポートとなるIOEにデータを渡し、IOEは受け取ったデータをコントローラに送る。コントローラはそのデータを計算結果としてHPに出力する。

4.2. 配線ネットワーク

各PEとIOEをつなぐ配線ネットワークを図3に示す。

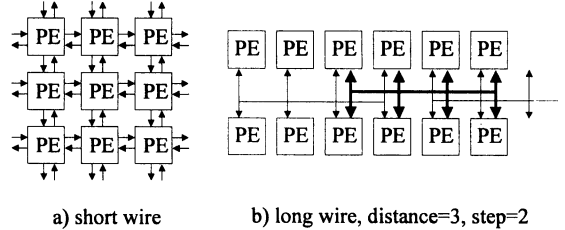


図3：配線ネットワーク

ネットワークはショートワイヤとロングワイヤの2種類の配線からなる。これらの配線はシリアル伝送を行うための2本のバスである。この2本の内一方を基本バス(master bus)、もう一方を反転バス(inverse bus)と呼ぶ。

元素間で送信/受信される信号の定義を図4に示す。クロックに同期して基本バスと反転バスの値が同じならハイインピーダンス、基本バスと反転バスの値が異なるなら基本バスの値をデータとして扱う。

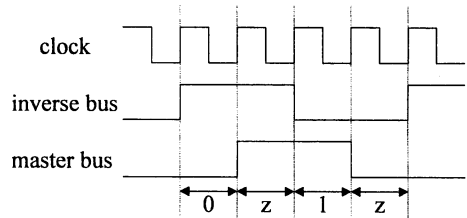


図4：データ信号の定義

ショートワイヤは隣り合う元素間にある単方向のワイヤである。各PEから4近傍に対して出力用ショートワイヤが張られているため、4近傍のPEとは自由に通信を行うことができる。IOEはPEとの間にもショートワイヤが張られている。なぜなら隣り合うIOE同士で通信する必要が無いためである。

ロングワイヤは元素の行/列間にあり、

ワイヤの両側に位置する複数のエレメントが共有する双方向バスである。あるエレメントがロングワイヤにデータを出力すると、そのロングワイヤを共有している複数のエレメントにデータがブロードキャストされる。どのエレメントがどのロングワイヤにデータを出力するか、あるいは出力しないかは再構成時に決定される。そして再構成時に、あるエレメントが出力に用いるロングワイヤは、一時的に他のエレメントがデータを出力することはない。

ロングワイヤは距離とステップ幅の2つのパラメータにより、バスの張り方が決定される。距離はあるエレメントを起点とし、行/列方向に何エレメント離れた所まで通信できるかを表す。ステップ幅はロングワイヤを行/列方向に何エレメントごとに張るかを表す。

図5のように、ある行/列間に存在するバスは最高でも(距離+1)本なので、他のアーキテクチャに比べて配線を抑えられている。また配線リソースにはFPGAのようにスイッチマトリクスが存在しないため、配線領域はそれほど大きくならない。

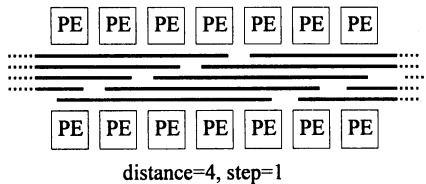


図5：行/列間に存在するロングワイヤ

ロングワイヤは行/列間にあるため、図6のように隣の行/列にあるエレメントと通信が行える。よってPEへの命令の割当や配線を柔軟に行うことが可能である。ロングワイヤを用いても通信できないPEと通信するには、途中に存在するPEを配線用に再構成し、そのPEを経由させる事で可能となる。ただしPEを経由する際に1クロックかかるため、注意が必要である。

2つのパラメータの取り方により、マッピング可能なアプリケーションや実装に要する面積が異なってくる。複雑な配線が必要なアプリケーションを主な対象とする時は、チップ実装時に距離を大きくステップ幅を小さく設定する。逆に単純な配線しか必要としないアプリケーションを対象とする時は、距離を小さくステップ幅を大きく設定する。前者だと遠くのPEと通信可能な共有バスが多数存在するため、柔軟にルーティングが行える。後者の場合ルーティングは限定されるが、入

力/出力セクタが小さくなり、1エレメントあたりの面積を小さくすることができる。

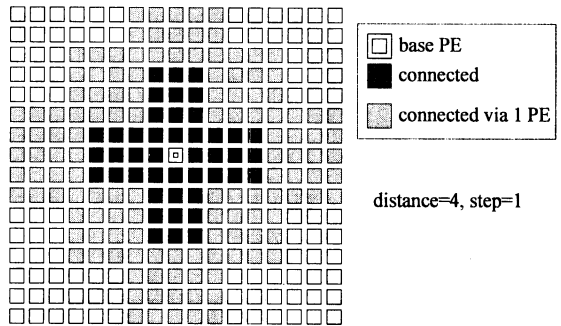


図6：通信可能なエレメント

4.3. プロセッサエレメント

PEは他のエレメントから入力されるデータを選択し、演算を行い他のエレメントに結果を出力する。他の粗粒度型アーキテクチャと異なり、PEは1bit幅のプロセッサである。そのため、シリアルで入力されたデータをシリアルのまま演算し、結果もシリアルで他のエレメントに出力する。演算ユニットがシリアルであるため、ALUの回路規模を削減できる。また加算などのキャリー伝搬が生じる演算がボトルネックにならないため高速化が容易となる。

PEが備えている命令は論理演算、除算以外の算術演算、シフト演算の他に任意の論理を実行できるLUT命令がある。またビットシリアルなプロセッサであるためデータ幅を変更する命令、右シフトのように出力に遅延が生じるデータと同期をとるための遅延命令を持っている。

PEの構成を図7に示す。

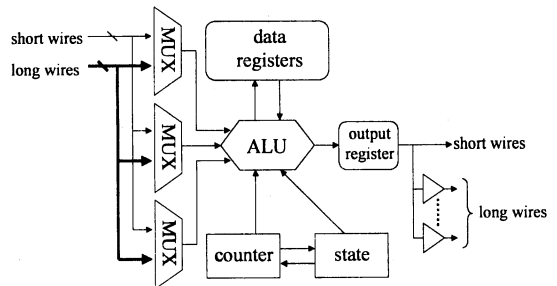


図7：PEの構成

PEはALUの他に入力セクタ、出力セクタ、データレジスタ、ステートレジスタ、カウンタからなる。データレジスタは3つの16bitレジスタであり、シフト演算や定数の記憶、乗算命令やLUT命令に利用される。ステートレジスタはPEの状態を保持するレジスタである。PEは動作/停止

の二つの状態を持ち、状態によって他のエレメントへの出力を、ハイインピーダンスにするかデータにするかを決定する。カウンタは、PEが動作状態では何bit目の処理を行っているか、停止状態では何クロックウェイトを置いたかをカウントするユニットである。カウンタがあることで可変長bitの処理や、ウェイトを自由に設定できる。

4.4. I/O エレメント

IOEはコントローラとPEとの間でデータをやりとりするための特殊なPEである。コントローラとのデータのやりとりをIOEに限定しているのは、配線が複雑になるのを防ぎPEアレイの数をスケラブルにするためである。コントローラと全てのPEが通信できる場合、膨大な配線が必要となる。またPEアレイの数が増えると配線遅延の問題も発生する。多くの粗粒度型アーキテクチャはこれらの問題があるため、エレメント数のスケラビリティがない、もしくは柔軟にルーティングが行えない配線リソースになっている。

IOEはPEからALUを取り除いて、データレジスタを4つの16bitレジスタに変更した構成である。またPEとはショート/ロングのシリアルバスで通信するが、コントローラとは16bitの平行バスで通信を行う。4つのデータレジスタは入力/出力バッファの役割を持ち、循環配列のような動作を行う。

5. 実装

4で述べたアーキテクチャを持つ再構成可能ハードウェアを表1の環境で設計した。

表1: 開発環境

Tool	name(vender)	version
HDL	Verilog-HDL	---
simulator	Verilog-XL(Cadence)	2.13
synthesis	Design Compiler(synopsys)	2000.11-sp4
layout	ApolloII(avant!)	2000.2.3.4.0.2
process	ROHM 0.35um 3metal-layer	---

外部環境とのインタフェースが決定していないため、今回はコントローラを実装していない。配線のパラメータとして配線距離を6、ステップ幅を1、動作クロックを133MHzに設定して論理合成、レイアウトを行った。結果、1個のPEの面積が0.1681 m²、IOEの面積は0.1444 m²であった。また3mm x 3mmチップ上に7 x 7個のエレメントを実装できることを確認した。

6. アプリケーション

本アーキテクチャにDCT演算を行う回路をマッピングした。DCT演算はJPEGやMPEGをはじめとする画像の不可逆圧縮によく利用される演算であり、処理全体の大半を占めている。JPEGやMPEGでは画像を8x8のマクロブロック単位に分割し、それぞれに2次元DCTをかけている。

6.1. 比較対象

本アーキテクチャとの比較対象としてソフトウェア(CPU)、細粒度型アーキテクチャであるFPGA[4]、そして粗粒度型アーキテクチャとしてMorphoSys[5]での結果を参考にしている。ソフトウェアはC言語で記述したものであり、その実行・開発環境を表2に示す。

表2: ソフトウェアの実行環境

OS	Windows 2000
CPU	Pentium II 350MHz
Memory	PC100 128MB
Compiler	Borland C++ 5.5.1

6.2. DCT演算器の構成とマッピング結果

DCT演算は図8のように、2つの1次元DCT演算器と行列変換器で構成している。DCT演算器をマッピングするため14 x 23(322)個のエレメントが必要であった。

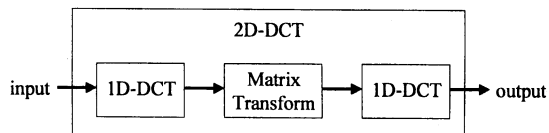


図8: DCT演算器の構成

マッピングしたDCT演算器にデータを入力しデータバスにデータが飽和すると、30クロックごとにマクロブロックの各列の結果が15bit幅で出力される。これは回路内部で小数部込み30bit幅の中間結果が出力されるが、最後に小数部を切り捨てるため、遅延が生じるデータ幅縮小命令を行うためである。よって1マクロブロック分(8列)の結果を得るには240クロック必要となる。

6.3. ソフトウェア、FPGAとの比較

MPEG2のMP@ML(720 x 576 pixel /frame, 30 frame /sec, Y:Cr:Cb=4:1:1)と同じマクロブロック数分のDCT演算を本アーキテクチャとそれぞれの比較対象で処理した結果を表3に示す。

結果より本アーキテクチャはソフトウェアやFPGAよりも高速な処理が可能であることがわかった。面積の評価として、我々が以前作成したア

イランド型の FPGA [6] と比較した。この FPGA は今回提案するアーキテクチャの実装と同じプロセスを用いている。エレメントあたりの面積は、本アーキテクチャは FPGA の 3 倍程度であった。しかし必要なエレメント数は 1/15 程度であるため、省面積で同じアプリケーションをマッピング可能である。

表 3 : 比較結果

	CPU	FPGA	Morpho Sys	Ours
frequency[MHz]	350	17.45	100	133
clks/MacroBlock	2.5M	64	37	240
time[ms]	2163	1069	107	525
elements	1	4386	64	322
Area[sq-mm]	118	267	180	60

6.4. MoryphoSys との比較

MoryphoSys と比較すると、処理速度の面では本アーキテクチャは低速である。しかしエレメント 1 個あたりの面積では MoryphoSys は 2.8mm^2 であった。本アーキテクチャは MorphoSys の 5 倍のエレメントが必要であるが、エレメント 1 個あたりの面積は 1/15 程度であった。よって本アーキテクチャは省面積で同じアプリケーションを処理可能である。また DCT 演算器だけでなく MPEG エンコーダを実装する場合、高負荷な処理として DCT, IDCT, Motion Estimation を行う必要がある。MorphoSys に MPEG エンコーダを実装する場合、コンテキスト切り替えを行わなければならないオーバーヘッドが生じる。これは MoryphoSys がアーキテクチャ的に PE の数を増やせない構成だからである。本アーキテクチャは PE の数を増やすため、全ての回路がマッピング可能な位の十分な面積があると、オーバーヘッドなく処理を行うことが可能である。

7. まとめ

本稿では低コストで効率の良い再構成可能アーキテクチャを提案した。具体的には面積の大半を占める配線領域を抑え、かつ柔軟に配線するためにシリアルバスのアーキテクチャを提案した。そしてバスアーキテクチャと親和性が高い粗粒度型のアーキテクチャを採用した。また粗粒度型の問題であるアプリケーションと PE のビット幅整合性問題を解決するため、1bit 幅の PE を提案した。この PE はビットシリアルデータバス向けであり、

高速化が容易である。アプリケーションが限定されるという粗粒度型のもう一つの問題については、PE の数を増やしやすいスケラビリティのあるバスアーキテクチャによって解決している。

本アーキテクチャに DCT 演算器をマッピングし処理させた結果、FPGA よりも少ない面積で高速に処理が行えた。MorphoSys とは処理速度は及ばなかったものの、少ない面積で同じアプリケーションを処理することが可能であった。

今後の課題としては他のアプリケーションをマッピングした場合について評価をすることが挙げられる。また配線距離とステップ幅を変化させたときに、特性がどう変化するかについて調べることも挙げられる。

文 献

- [1] 片山勝, 甲斐英則, 吉田順一, 山田博希, 塩本公平, 山中直明, "リコンフィギャラブルプロセッサを用いた 10Gb/s ファイアウォール装置の実現," 第 4 回リコンフィギャラブルシステム研究会, pp.67-72, Sept.2004.
- [2] 大澤尚学, 坂本修, 張山昌論, 亀山充隆, "ビットシリアルパイプラインアーキテクチャに基づくフィールドプログラマブル VLSI プロセッサの設計," 研究報告 SLDM vol.2003 NO.105, pp.145-149, October.2003.
- [3] Akihisa OHTA, Tsuyoshi ISSHIKI, Hiroaki KUNIEDA, "A New FPGA Architecture for High Performance Bit-Serial Pipeline Datapath," IEICE TRANS. Fundamentals VOL. E83-A NO. 8, pp.1663-1672, August.2000.
- [4] Discrete Cosine Transform MegaFunction Data Sheet, <http://166.111.64.217/datasheet/altera/sb/sb009.pdf>
- [5] Hartej Singh, Ming-Hau Lee, Guangming Lu, Fadi J. Kurdahi, Nader Bagherzadeh, "MorphoSys: An Integrated Reconfigurable System for Data-Parallel Computation-Intensive Applications", Computers, IEEE Trans, pp.465-481, May.2000.
- [6] S. Kimura, T. Horiyama, M. Nakanishi and H. Kajihara, "Folding of Logic Functions and Its Application to Look Up Table Compaction", Proc. on ICCAD 2003, pp.694-697, November.2002.