

FPGAを用いたマルチメディア移動通信用プロセッサ

FPGA-Based Processor for Multimedia Mobile Communication

大澤 尚学, 張山 昌論, 亀山 充隆

Naotaka Ohsawa, Masanori Hariyama, Michitaka Kameyama

東北大学情報科学研究科

Graduate School of Information Sciences
Tohoku University

キーワード : ビットシリアルアーキテクチャ(bit-serial architecture), ハイレベルシンセシス (high-level synthesis), スケジューリング(scheduling), アロケーション(allocation)

連絡先 : 〒980-8579 仙台市青葉区荒巻字青葉05 東北大学大学院情報科学研究科亀山研究室
大澤尚学, Tel.: (022)217-7155, Fax.: (022)263-9167, E-mail: ohsawa@kameyama.ecei.tohoku.ac.jp

1. まえがき

マルチメディア移動体通信, 通信, ロボティクス, 制御などの高速応答性が要求される応用分野においては, 汎用マイクロプロセッサだけでは速度要求を満たすことが難しいため, 専用プロセッサの開発が望まれる. 従来, 専用プロセッサとしては, プロセッサ毎に専用のマスクを作成するASICが主流であった. しかしながら, ASICの開発は, 開発期間が長く, コストも高い. 近年, このような問題を解決するデバイスとして, フィールドプログラマブルゲートアレイ(FPGA)が注目を集めている. Fig. 1に現在の典型的なFPGAの構成を示す. FPGAは, プログラム可能な演算器(LUT)と, LUT間を接続するプログラム可能な配線(SB)から構成される. 各LUTは任意の n 入力1出力関数(通常 $n = 4$ 程度)を実現できる. LUTやSBをプログラムすることにより, 並列処理が行えるため, 従来の汎用プロセッサに比べ, 高性能化を達成できる. さらに, 開発期間

もASICの場合に比べ大幅に短縮できる. また, 1種類のFPGAデバイスを種々の専用プロセッサとして使用できるため, ASICに比べコストも減少できる. FPGAの問題としては, 以下のような原因により, ASICに比べ処理速度が大幅に低下するということが挙げられる.

- LUTはメモリを用いて実現されているため, 専用に演算器を組んだ場合に比べ, 演算器の速度が低下する.
- LUT間の配線がSBを多段に通過する場合には, 配線遅延により速度が低下する.

特に, 配線遅延に起因する性能劣化がFPGAにおいては深刻な問題となる. Fig. 2に現在FPGAとしてはトップレベルの性能を有するXilinx社のVirtexを用いて, デジタル通信において重要な処理である逆FFTを実現した場合の配線遅延を示す. LUT(4入力1出力関数)の遅延が1ns程度であるため, LUT

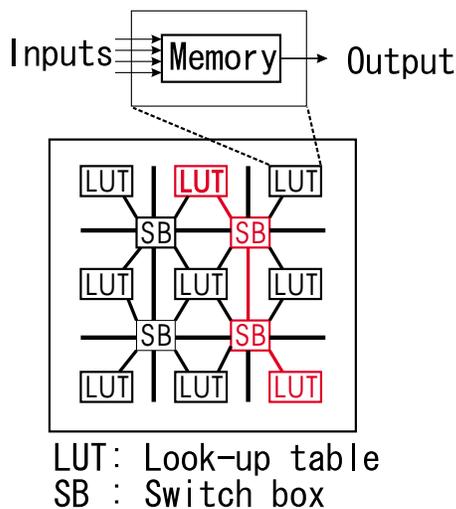


Fig. 1 従来のFPGAの構成.

の遅延よりも遅延時間が大きい配線が全体の84%以上を占めることがわかる. したがって,FPGAにおいては配線遅延に起因する性能劣化の解決が重要な課題となる.

以上のようなFPGAの問題を解決するために, 本稿では, 高性能リアルワールド応用FPGAの構成を提案する. 提案のFPGAでは, 演算器を高速化するために, 論理回路で構成したALUを内蔵した演算器(PE)を用いる. 演算器を論理回路で構成する場合に, 応用に依存して最適な語長が変化するという問題が生じる. そこで, 語長に依存せず演算器の稼働率を高くできるように, ビットシリアルアーキテクチャに基づくPEを提案する. また, 配線遅延に起因する性能劣化を解消するために, 通信の局所化を徹底したシフトレジスタ構造に基づく相互結合網を用いる. この構成では, 演算器間の通信の局所化・並列化が性能向上のポイントとなるため, 最適スケジューリング・アロケーションおよび,冗長性に基づく通信の局所化・並列化を提案する.

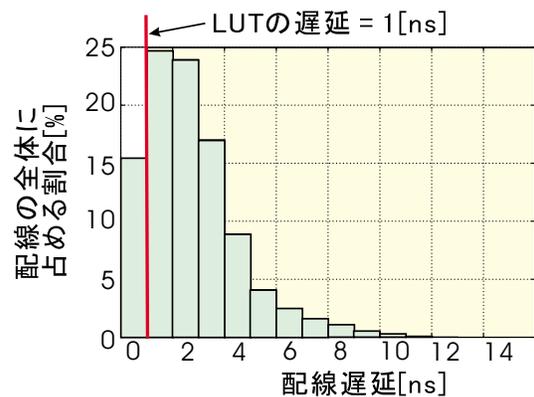
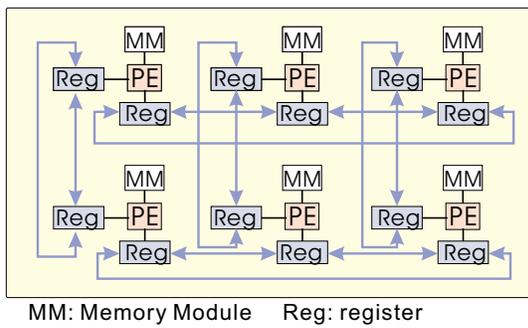


Fig. 2 FPGAを用いて16点逆FFTを実現した場合の配線遅延.

2. リアルワールド 応用FPGAの構成

2.1 全体の構成

Fig. 3に提案するFPGAの構成を示す. 1種類の演算器(PE)が複数個内蔵され, 各PE間では隣接したPE間でだけ配線を有する. そのため, これらの配線は専用化され高速な通信が可能となっている. 隣接したPE間では1ステップで通信が行える. また, PEに接続されたレジスタ間でデータをシフトすることにより通信を行うため, 隣接しないPE間での通信には, 通常PE間の距離に比例したステップ数がかかる. そのため, このアーキテクチャにおいては, 2.3節で述べるように通信時間をできるだけ短くするような工夫が重要となる. また, PEにおいては, 語長が異なる種々の処理を効率よく実行できることが望まれる. Fig. 4は, マルティメディア移動体通信の送信処理の一例である. 誤り訂正符号化のための畳み込み符号化においては, 1ビットの演算が行われる. また, OFDM変調においては16ビット以上の処理が要求される. このような場合には, PEのALUとして, 16ビット語長のものを用いたとすると, 1ビット処理を行う場合に, 15ビット分のハードウェアが使用されないことになる. そこで, 語長の異なる種々の処理に対して,



MM: Memory Module Reg: register

Fig. 3 提案するFPGAのブロック図.

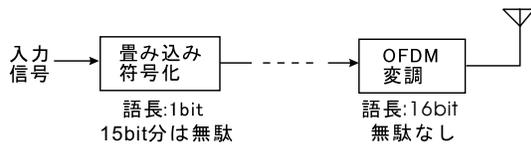


Fig. 4 語長が変化する処理の例.

ハードウェアの稼働率を高めるために、2.2節で述べるようにビットシリアルアーキテクチャに基づきPEを構成する。

2.2 ビットシリアルアーキテクチャに基づくPEの構成

PEでは、種々の演算に対応するために、以下の基本演算を行えるように設計する。

- 算術演算(加算, 減算, 除算, 乗算)
- 論理演算(論理否定, 論理和, 論理積)
- 比較演算(大小比較, 一致検出)

これらの機能を実現するための、PEの構成をFig. 5に示す。PEは、大きく分けて、算術演算を行うための部分と、論理演算を行う部分から構成される。算術演算は、頻繁に用いられる演算であるため、高速に実行できることが望まれる。算術演算は全て全加算の繰り返しにより行われるため、全加算器(FA)を専用に用意し、高速化を図っている。

論理演算は、全加算に比べると高速に行えるため、始めに、論理否定、論理和、論理積の出力を選択し、その出力と算術演算の出力を選択することに

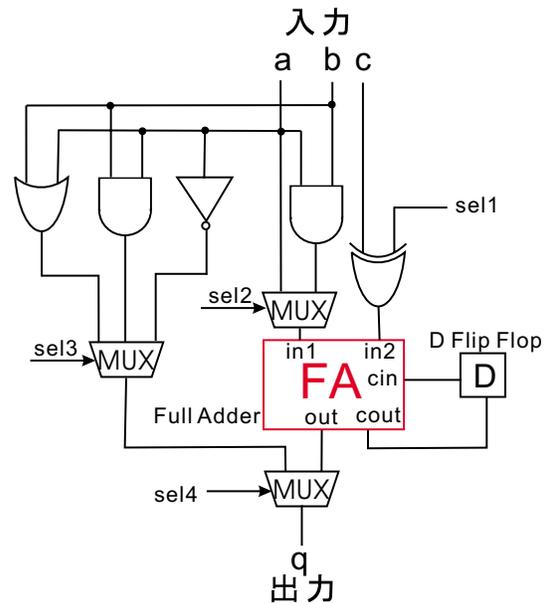


Fig. 5 PEの構成.

より、論理否定、論理和、論理積、算術演算を4-to-1MUXで選択する場合に比べ、算術演算の出力の遅延ができるだけ少なくなるように設計している。

2.3 メモリの構成

ビットシリアル処理を効率よく行うためには、ビットシリアルにPEにデータを供給する機構が重要となる。通常のメモリでは、Fig. 6(a)に示すように、読み出すべきデータの先頭アドレスを指定し、アドレスカウンタをインクリメントすることで、ビットシリアルなデータ転送を実現する。この方法では、アドレスカウンタが通常多ビットになるため、アドレスカウンタの遅延がPEの遅延よりも大きくなり、性能劣化を生じる。そこで、Fig. 6(b)に示すように、シフトレジスタを用いたアドレッシング方式を提案する。各ワード毎に、ワードラインにレジスタが接続され、これらのレジスタはワード方向にシフトできる構成となっている。読み出すべきデータの先頭アドレスに対応したワードのレジスタに1がセットされる。その後は、クロックに同期して、1がシフトされ、レジスタの値が1である

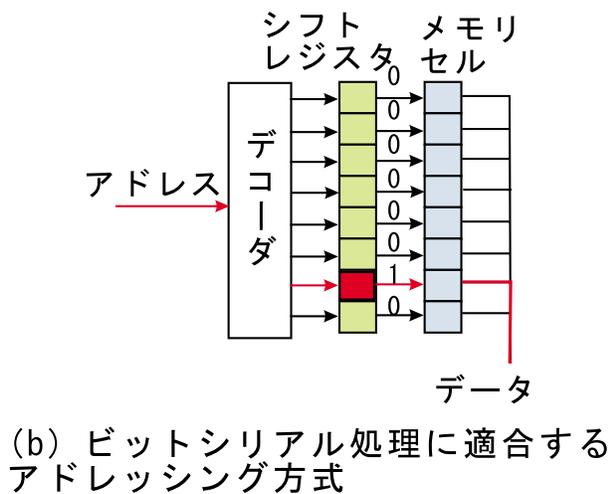
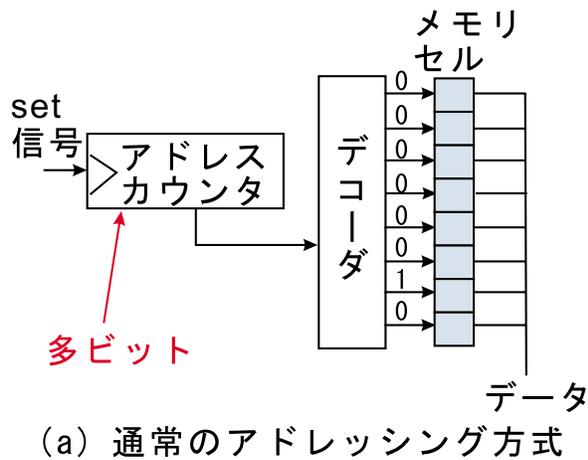


Fig. 6 メモリの構成.

ワードの内容が読み出される。この方式では、アドレス指定が、データシフトだけで行えるため、高速なアドレッシングが可能である。

2.4 PEの評価

Fig. 7に、0.6 μm CMOS 2層配線プロセスを用いたPEのレイアウトを示す。また、表1にチップ諸元を示す。ビットシリアルアーキテクチャを用いているため、コンパクトな構成になっていることが分かる。

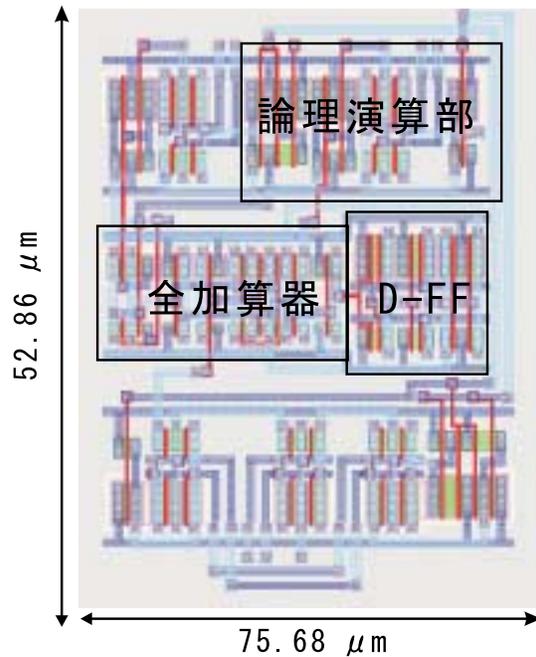


Fig. 7 PEレイアウト.

3. シフトレジスタ構造相互結合網のための通信の局所化・並列化

3.1 最適スケジューリング・アロケーション

シフトレジスタ構造に基づきデータ通信を行うアーキテクチャにおいては、スケジューリングおよびアロケーションが高性能化の重要なポイントとなる。Fig. 8にスケジューリングおよびアロケーションの例を示す。簡単のために、1次元のシフトレジスタ構造を考える。Fig. 8(a)に示すようにスケジューリングおよびアロケーションを行った場

Table 1 PE諸元

設計ルール	0.5- μm CMOS 2層配線
面積	52.86 × 75.68 μm^2
電源電圧	3.3V
処理時間	3.8 ns/bit
トランジスタ数	76

場合には、演算 O_1 の結果を O_2 で使用するために、 PE_1 から PE_3 へのデータ転送が必要となる。また、演算 O_3 の結果を O_4 で使用するために、 PE_2 から PE_4 へのデータ転送が必要となる。この場合、 PE_2 と PE_3 の間の配線において、通信の競合を生じないようにするために、Step 2において、 PE_2 から PE_4 へのデータ転送を一時中断している。一方、Fig. 8(b)に示すスケジューリング・アロケーションでは、PEでの通信が局所化されているため、演算 O_1 の結果を O_2 で使用するための通信 PE_1 - PE_3 、演算 O_3 の結果を O_4 で使用するための通信 PE_2 - PE_4 が並列に行える。このように、スケジューリングおよびアロケーションにより、処理時間が異なるため、最適なスケジューリング・アロケーションを求めることが重要となる。

本アーキテクチャでは、アロケーションにより通信時間が異なるために、スケジューリングにおいてもアロケーションの情報を考慮する必要がある。そのための手法として、スケジューリング・アロケーションの統合に基づく手法^{1),2)}を用いる。この方法では、各演算ノードに対するスケジューリング・アロケーションをFig. 9に示すように木探索で表現している。さらに、総当たり探索では探索空間が膨大となるために、クリティカルパスに着目した処理時間の下限に基づく分枝限定法により探索空間を効率よく限定している。

本稿では、FPGAアーキテクチャの性質に着目し、探索空間をさらに限定する。従来の問題設定¹⁾では、各PEは異なる機能を有する場合を考慮した一般的な場合を取り扱っていた。一方、本稿では、各PEは全て同じ構成であるために、明らかに性能が同じアロケーションが存在する。そこで、そのようなアロケーションの探索を省略することにより、探索空間を限定できる。

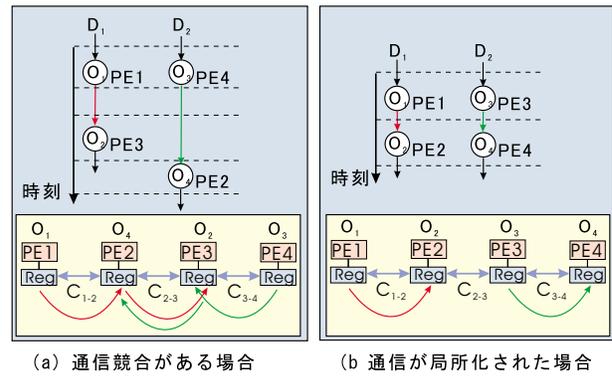


Fig. 8 最適アロケーション・スケジューリングの必要性.

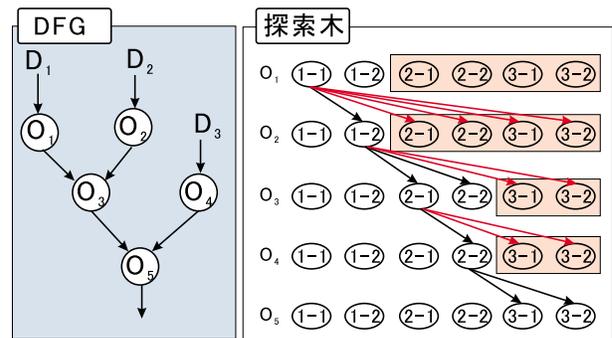


Fig. 9 木探索によるスケジューリングとアロケーションの統合.

3.2 冗長性に基づく通信の並列化・局所化

通信の並列化・局所化を実現するための、別のアプローチとして冗長性に着目した通信の通常、PEの面積が大きい場合には、PEの稼働率を高めることによりチップ面積制約下での性能を向上できる。一方、本アーキテクチャでは、ビットシリアルアーキテクチャに基づきPEを構成しているために、PEの面積は小さい。このような場合には、演算器を冗長に用いたとしても、通信の局所化・並列化により、PE間通信に起因する性能劣化を解消できれば、全体の高性能化を達成できる可能性がある。

このような冗長性に基づく通信の並列化・局所化の例として、アルゴリズムレベルでの例を示す。逆FFTはデジタル通信や信号処理において、よく用いられる処理である。Fig. 10(a)に4点FFTの

- 2) 工藤隆男, 亀山充隆, “転送ボトルネックフリーVLSIシステムのハイレベルシソセス,” 計測自動制御学会東北支部研究集会, 188-6(2000).

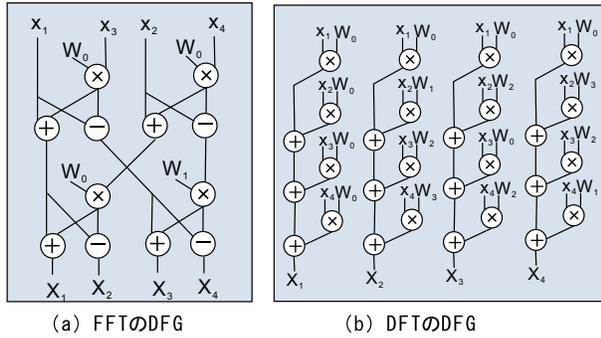


Fig. 10 4点FFTと4点DFTのDFG.

データフローグラフ(DFG)を示す. 逆FFTのDFGでは, ある演算ノードの出力が複数の演算ノードの入力として使用されるため, 並列処理を行う場合には, グローバルな通信が必要となる可能性が高い. 一方, Fig. 10(b)に示すように4点DFTでは, 計算量は増加するが, 各演算ノード出力が1個の演算ノードの入力としてだけ使用されるため, 通信の並列化・局所化を実現できる.

4. むすび

本稿では, 通信の局所化を指向したシフトレジスタ構造に基づくFPGAアーキテクチャを提案した. また, 種々の応用における語長の変化に対応するために, ビットシリアルアーキテクチャに基づくPEおよびメモリアーキテクチャを提案した.

また, 任意のDFGが与えられた場合に, 本アーキテクチャにマッピングを行うスケジューリング・アロケーション手法の検討が重要である. 現状では, 実用的な時間でマッピングできるのは, ごく小規模の問題だけである. そこで, より大規模かつ実用的な問題を, 本アーキテクチャにマッピングするための手法の検討が重要となる.

参考文献

- 1) 工藤隆男, 亀山充隆, “シフトレジスタ構造に基づくロジックインメモリVLSIプロセッサとその応用,”