

博士（工学）論文概要

再構成可能集積回路の 効果的な応用に関する研究

システム情報工学研究科 コンピュータ・サイエンス専攻

相部 範之

2005年3月

研究の背景と目的

本研究の目的は、再構成可能集積回路の1つである、FPGA (Field Programmable Gate Array) の「再構成可能」という特徴を生かせる応用(アプリケーション)を探索し、その有効性を実証することである。本論文では、4つの異なる側面からの適用例を用いて、FPGAの「再構成可能」という特徴を生かせるアプリケーションについて論じる。

FPGAなどの再構成可能集積回路は1980年代より一般に、LSI開発における、フォト・マスクを作成する前のロジック/タイミング検証(論理エミュレーション)を行うためのラピッド・プロトタイプリング・ツールとして、また、製造数が少ない場合におけるASIC (Application Specific Integrated Circuit) や ASSP (Application Specific Standard Product) などの代替として用いられてきた。そして、近年さらに、その「再構成可能」という特徴をハードウェア・アーキテクチャに取り込み、従来の逐次型MPU (Micro Processor Unit) では実現が困難な処理を効率良く処理しようとする試みが注目を集めている。これはリコンフィギャラブル・コンピューティングとよばれ、特に算術演算の高速化を中心に研究が進められており [1]~[10], FPGAよりさらに算術演算向きなアーキテクチャの再構成可能集積回路が次々に開発されている [7][9][10]。一方、本研究では従来の一般的な再構成可能集積回路であるFPGAを対象として、その再構成性を生かす応用について検討した。

FPGAは、実装対象となるアルゴリズムの並列度が非常に高い場合、現在注目されている算術演算向きの再構成可能集積回路よりも、高速な処理を実現できる可能性がある。また、I/Oインターフェースなど、小規模で、高速な処理を要求する制御回路などのランダム・ロジックは、FPGAでの実現に向いている。さらに、FPGAは既に市場に出してから20年以上経っており、ASIC/ASSPなどの代替として製品に組み込まれ、普及しているため、その応用であれば実用化もし易い。そこで、本研究ではFPGAへの並列度の高いアルゴリズムの実装、I/Oインターフェースへの応用などを行い、FPGAの「再構成可能」という特徴を生かせるアプリケーションの有効性を実証した。さらに、FPGAを用いた新しいハードウェア開発手法を提案し、IP (Intellectual Property) の流通への応用を行った。

再構成可能集積回路

再構成可能集積回路は、その名の通り回路構成を再構成できる集積回路で、回路の再構成によってさまざまなアプリケーションに容易に対応可能である。このため、従来より、フォト・マスク・ベースのLSIを設計する際の検証ツールとして利用されてきた。しかし、今日の半導体製造技術の発展により、再構成可能集積回路の集積密度、つまりその上に実現可能な回路の規模が大きくなることで、その製品開発におけるTAT (Turn Around Time) の速さから、フォト・マスク・ベースのLSIに代わり、製品にも実装されるようになった。製品の製造数によってはコスト面でもASICなどに勝る場合があり(図1)、メーカーは製品の種類や市場規模などによってフォト・マスクから作成する場合と、再構成可能集積回路を採用する場合とを使い分けるようになった。

再構成可能集積回路の基本構造は、基本要素となる回路が並列多重実装されており、その基本要素回路間の接続を変更可能とすることでさまざまな回路を実現できる。またその種類は、基本要素となる回

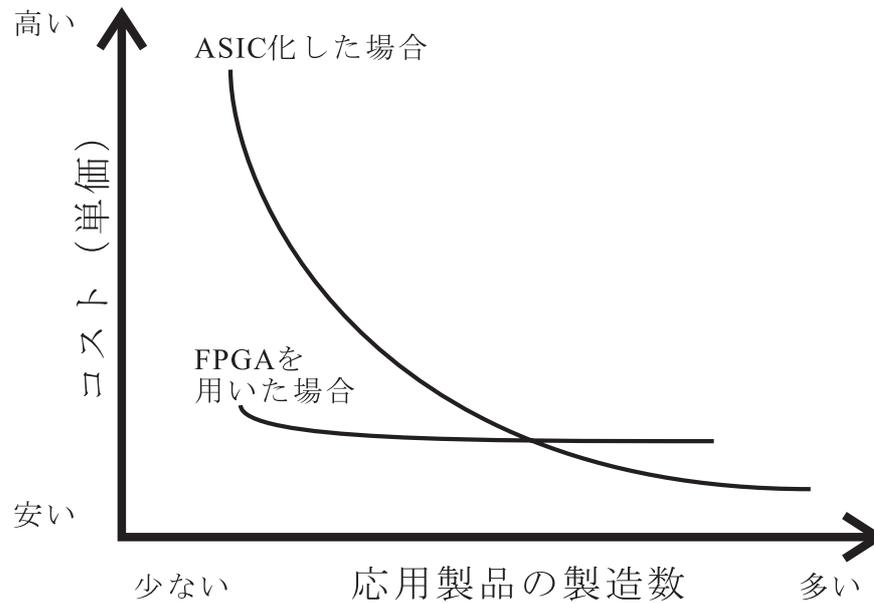


図1 ASIC化した場合とFPGAを採用した場合の応用製品の製造数とコスト

路の種類や、回路間の接続を変更可能にする方式の違いなどによりさまざまなものがある。現在の再構成可能集積回路は、大きく LUT (Look Up Table) アレイ型と ALU (Arithmetic Logic Unit) アレイ型に分類できる。LUT とは、予めデータを書き込んだ RAM のアドレス入力を、データ入力として扱うことで、結果的に書き込まれたデータに従った、任意の組み合わせ回路を実現するものである。従ってその実装形式は RAM そのものとなる。ALU は、従来から MPU に用いられている演算回路のことで、AND、OR、NOT などの論理演算に加え、加算、減算、シフト演算、或いは掛け算などの算術演算を行う回路を一つにまとめ、外部からの命令により指定の演算を行うものである。実装形式は通常、トランジスタ・レベルで直接製造される。LUT アレイ型と ALU アレイ型は、粒度という面から見ると LUT アレイ型が細粒度で、ALU アレイ型が粗粒度である。また、逐次処理型の MPU は、ALU アレイ型よりもさらに粒度が粗いと考えることが出来る(表1)。まず、LUT アレイ型は、基本要素とな

表1 実装方式の分類

	LUT アレイ型	ALU アレイ型	MPU
例	XILINX Virtex[23], Spartan[25], ALTERA Stratix[28], Cyclone[29]	広島大 FPAccA[7], NEC DRP[10] IP-Flex DAP/DNA[9]	Intel x86 系, IBM/Motorola PowerPC 系, Z80, H8, ARM
粒度	細かい	粗い	さらに粗い
適した設計レベル	RTL	演算命令レベル	Behavior Level

る回路を、LUT という NAND ゲートのように汎用性の高いものとし、これを通常のゲートアレイのように多数並べて、その接続先を多数のスイッチで切り換える構成（細粒度の構成）で、非常に汎用性の高い再構成可能集積回路である。しかし、この構成では、アプリケーションによっては要求される基本要素回路、スイッチの数が非常に多くなり、信号遅延が大きくなることで、全体の動作周波数を高くできない。XILINX 社や ALTERA 社製の FPGA や CPLD がこの構成にあたる。ALU アレイ型は、ALU という、ある程度大きな演算回路を基本回路としたもの（粗粒度の構成）で、必要なスイッチ数、基本要素回路数共に少なくなり、全体の動作周波数が高められる。しかし、粒度が粗いため、アプリケーションによっては実装が困難となる。このような構成としては、広島大学 越智の FPAccA (Field Programmable Accumulator Array)[7][8]、IP-Flex 社製の DAP/DNA[9] や NEC エレクトロニクス社製の DRP[10] などがある。また、従来の Intel x86 系、IBM/Motorola PowerPC 系、或いは組み込み機器向けの Z80、H8、ARM など、ノイマン型アーキテクチャの MPU は、一般に単一の ALU で構成される。このため、前述の再構成可能集積回路と異なり、複雑な切り替え機構などが無い分、信号遅延が小さく、最も高い周波数での動作が可能となる。しかし、最も演算器の粒度が粗い（演算器が 1 つしかない）ため、特に並列処理を基本とするアプリケーションは実装効率が悪い。

従って、再構成可能集積回路は単にラピッド・プロトタイピング・ツールとして機能するだけでなく、そのアーキテクチャは、従来のノイマン型アーキテクチャよりもアプリケーションによっては、その実装に適したアーキテクチャである可能性がある。次に、このように再構成可能集積回路のアーキテクチャを、演算器アーキテクチャとして捉えた、リコンフィギャラブル・コンピューティングについて述べる。

リコンフィギャラブル・コンピューティング

近年、FPGA などの再構成可能集積回路を用いて実現される、リコンフィギャラブル・コンピューティングの研究が盛んに行われている [1]~[12]。リコンフィギャラブル・コンピューティングとは、再構成可能集積回路の「再構成機構」をハードウェア・アーキテクチャとして取り込み、アプリケーションに適した計算方式を実現するものである。現在行われている研究の多くは、処理速度の面で MPU (Micro Processor Unit) を使うことが困難な演算を高速化するものに関する。

ある特定のアプリケーションを実現するために、速度、サイズ、消費電力の全ての性能面で最も高い性能を達成するハードウェアを製造する方法は、フル・カスタムのフォト・マスク・ベースで製造する方法である。これは一般的に特定用途向け集積回路 ASIC(Application Specific Integrated Circuit) と呼ばれる。しかし、ASIC はその都度フォト・マスクから設計、製造を行う必要があるため、非常に高度な設計・製造技術が必要となる他、開発・製造期間が長く、開発・製造コストも高い。フォト・マスク製造の期間を短縮し、また製造コストを削減する目的で、フル・カスタム方式の他にスタンダード・セル方式、ゲートアレイ方式、或いはストラクチャード ASIC などがある。しかし、予め製造しておけるのはフォト・マスク・レベルであり、工場で製造する必要があることに変わりない。一方、既存の MPU を用いて処理する場合は、予め MPU を搭載したシステムを作り上げた後で、任意のアプリケーションを実装することができる。従って、その実行性能が多少落ちたとしても、開発期間 (TAT: Turn Around Time) やコストを含めて評価したときに、MPU での実現が採用される場合が多い。従って、

異なるアプリケーションにどの程度対応できるかという“汎用性”と、速度，サイズ，消費電力などの実行性能を考えると，図2のようになる．リコンフィギャラブル・コンピューティングが対象としている領域は，フォト・マスク・ベース LSI による特定用途向けの固定アーキテクチャと，MPU のノイマン型アーキテクチャの間にある．リコンフィギャラブル・アーキテクチャの優れた点は，フォト・マスク・ベース LSI を製造する場合よりも開発・製造期間が短く，低コストである上に，アプリケーションによっては，MPU で処理する場合よりも高性能な実行性能を得られることである．

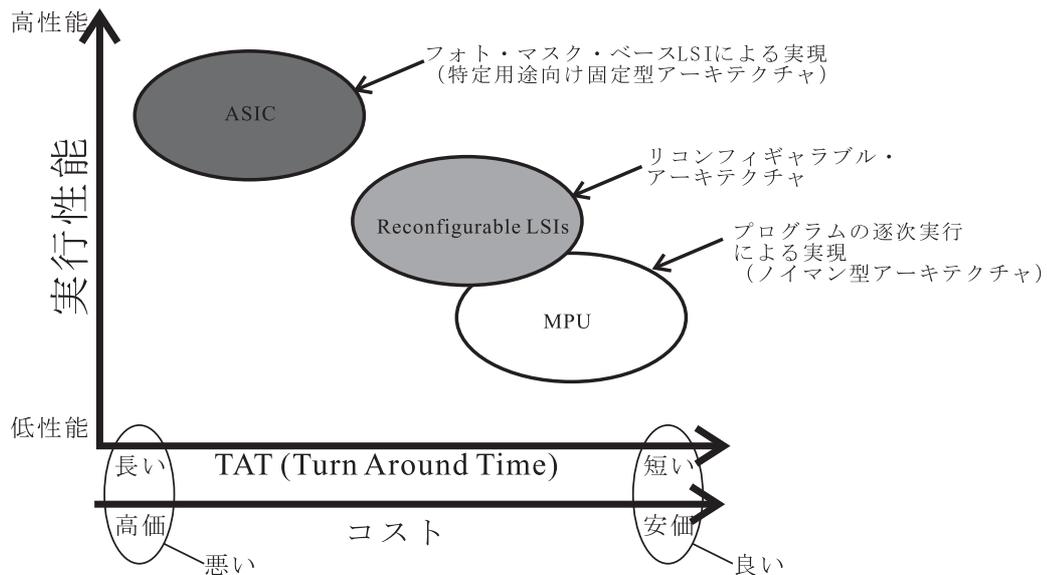


図2 リコンフィギャラブル・アーキテクチャの位置付け

現在，リコンフィギャラブル・コンピューティングの研究で注目されているのは，特に算術演算の高速化である．これは，従来の MPU の逐次処理型アーキテクチャに代わる新たな“汎用コンピュータ・アーキテクチャ”の一つ，という位置付けで進められている研究が多い．一方，本研究では算術演算の高速化アーキテクチャ以外の応用を対象とし，その効果的な応用を検討した．これは FPGA のような LUT アレイ型の再構成可能集積回路の方が，ALU アレイ型の再構成可能集積回路より，粒度が細かく，MPU の粒度とその差がより離れているため，MPU が苦手とする処理をより効率的に処理できる可能性があると考えたためである．

FPGA の効果的な応用

FPGA の効果的な応用（アプリケーション）は，従来の MPU で高い実行性能（速度，サイズ，消費電力）を得ることが困難で，かつ高い汎用性を要求するものである．これは，

1. 論理回路レベル (RTL) での高い並列性を活かせる，かつ，
2. 論理回路レベル (RTL) での後からの大きな変更が頻繁に要求される，

アプリケーションである．具体的には，パターン・マッチング，暗号処理，通信制御，I/O インターフェースなどが考えられる．

論理回路レベル (RTL) での高い並列性を活かせる (MPU より優れた点)

現在，ノイマン型アーキテクチャの MPU の処理速度は，逐次処理でありながら，数百 MHz 後半～数 GHz と非常に高い基本動作周波数で動作しているため，かなり高速である．このため，同一の製造技術で作られていながら，汎用性を高めたことで数十 MHz～数百 MHz 前半と基本動作周波数が下がってしまった FPGA では，MPU で行っていた処理をそのまま実装しても高速化は図れない．FPGA で高速化が図れるのは，並列性の高い処理のみである．MPU は演算の種類によって実行に必要なクロック・サイクル数が異なり，さらにパイプライン・ストールなどの影響を考慮する必要があるため，正確な性能比較は難しいが，例えば，単純に基本動作周波数 1GHz の MPU と同様の処理速度を，基本動作周波数 100MHz の FPGA で実現するためには 10 倍程度の並列度が必要となる．逆に，数十～数百倍以上の並列度が得られる処理であれば，FPGA により，MPU より高速な処理の実現が期待できる．

また，通信機器の内部制御や I/O インターフェースなどで用いられる，多数のパルスのタイミングを並列に制御する回路などは，細粒度で RTL での設計に向く FPGA の方が，ALU アレイ型の再構成可能集積回路や MPU のような粗粒度なアーキテクチャに比べ，実装効率が高い．従って，粒度の細かい処理の場合は FPGA の方が高速，或いは小さい回路規模で実現できる可能性がある．

論理回路レベル (RTL) での後からの頻繁な大きな変更に対応可能 (ASIC より優れた点)

論理回路レベル (RTL) での構成が大きく変わらない場合には，複数の要求を満たす回路全てをワイヤード・ロジックで実装し，要求に応じて切り換えられるような回路構成で ASIC 化すれば，最も高性能なハードウェアが実現できる．しかし，要求に応じて回路の構成を大きく変える必要がある場合には，各回路の独立性が高くなる (共有化可能部分が減少する) ため，それぞれの回路をワイヤード・ロジックで実装すると，実装に必要な回路規模が大きくなってしまふ．さらに，ASIC 化の時点で想定されていなかった全く新しい機能を後から追加するようなことは，特定用途向けの固定アーキテクチャではまず不可能である．このような想定外のことにも対応できるような汎用性を持たせることは，即ち汎用向けのリコンフィギャラブル・アーキテクチャを採用することとなる．広義の意味においては，要求に応じて論理回路レベルでの構成が少しでも切り換えられるような構成は，全てリコンフィギャラブル・アーキテクチャであると言える．しかし，本論文では現在の再構成可能集積回路の LUT アレイ型アーキテクチャ，或いは ALU アレイ型アーキテクチャのように，汎用性の高い基本演算回路を多数並列実装し，後から想定外の全く異なる機能を実現できる，汎用アーキテクチャをリコンフィギャラブル・アーキテクチャと定義する．

また，ASIC 化した後から，前述のように回路構成の変更が大きく要求される場合でも，その頻度が数ヶ月～数年単位など非常に低く，かつ，コンシューマ向けの量産製品など，コスト面でも問題が無い場合には，ASIC でも，製造し直すことで対応できる．しかし，要求される回路変更の時間間隔が ASIC の

開発サイクルよりも短い場合，さらには数 ms サイクル以下で回路変更を行う動的再構成などは，FPGA などのもつ，リコンフィギャラブル・アーキテクチャでなければ実現できない．また，先に述べたように，ASIC の製造には多額のコストも必要となるため，量産製品などでなければ，ほとんどの場合において ASIC 化は困難である．

具体的な応用と研究手法

本論文では細粒度の再構成可能集積回路である LUT アレイ型の FPGA を用いて，上記のように MPU や ASIC などを実現が困難なアプリケーションの実現性を評価し，再構成可能集積回路の可能性について論じる．具体的には，FPGA に下記の 4 つのアプリケーションを実装し，評価することでその可能性を探った．

1. 確率的ニューラルネットワークを用いた画像認識装置
2. 性能可変 IP パケット・フィルタ装置
3. 汎用 I/O インターフェース
4. 新しいハードウェア開発手法

以下，各応用の詳細と，再構成性の応用点について述べる．

確率的ニューラルネットワークを用いた画像認識装置への適用

まず，並列性が非常に高いパターン認識のためのアルゴリズムである，確率的ニューラルネットワークと，それを用いた画像認識装置への適用を行った．確率的ニューラルネットワークは，アルゴリズムの 並列性が非常に高い ため，細粒度の FPGA への実装に非常に適したアルゴリズムである．さらに画像認識装置では，画像処理や特徴抽出などの前処理も必要となるが，この前処理は対象の種類などに応じて，大きくかつ頻繁 に回路の変更が要求されるため，このことから FPGA に適したアプリケーションである．本論文では，認識対象に合わせて，処理回路の再構成を行うことに加え，確率的ニューラルネットワークを FPGA と外部 RAM を用いて効率よく実装する，新たな手法を提案し，その実装結果と評価結果について述べる．

確率的ニューラルネットワーク (PNN: Probabilistic Neural Network)[13][14] はノンパラメトリックな統計的パターン認識手法の一つで，その認識精度の高さから，近年，顔認識 [15] や気象画像認識 [16][17] といった大規模な実用問題への適応が盛んに行われている．その理由は，構造が単純で並列性が高く，認識精度を決定するパラメータも 1 つのみで，扱いやすい上に，高い認識精度を得られるからである [18]．PNN のもつ認識パラメータは 1 つで，これが PNN の推定する確率密度分布の形状を決定し，認識精度を大きく左右する．従って，このパラメータは慎重に決定する必要があるが，その最適値は，認識対象や認識環境によって変化するため，これを決定論的に定める方法はない．このため，広範囲な探索空間を探索する必要があり，これには高速なプロセッサを用いても膨大な処理時間を要する．

そこで通常このパラメータは Try and Error で人手によって経験的に決定されるか，全探索することになる．経験的に手動で決定する場合は確実性，再現性が低く，また認識システムを構築するまでに時

間が掛かる．一方，全探索の場合は確実で，再現性も高いが，やはり探索には時間が掛かってしまう．そこで本論文では，この認識パラメータを高速に全探索し，最適値を求めることのできる，従来にない新しいハードウェア・アーキテクチャ SPA(Sigma-Parallel Architecture) を提案する．SPA は，マルチ・プロセッサの実装の上で問題となる，メモリ・プロセッサ間のピン・ボトルネック問題を解決し，従来より，多くのプロセッサの実装を可能にする．このことから，探索の並列度を高くすることができ，高速な探索が可能となる．

本アーキテクチャで設計し，XILINX 社製の FPGA である XCV1000E-6HQ240C[24] に実装した回路は，Pentium 1GHz の PC 上に実現したソフトウェア・プログラムで探索する場合に比べ約 68 倍の速度性能が得られることを確認した．また，実際にこの学習エンジンを含むハードウェア認識システムを開発し，認識問題として，手形状画像を対象とした簡単な認識処理の評価実験を行った．

IP パケット・フィルタ装置への適用

次に，コンピュータ間通信ネットワークで用いられる，IP (Internet Protocol) パケット・フィルタ装置への適用を行った．IP パケット・フィルタの処理も，確率的ニューラルネットワークの場合と同様に，非常に高い並列性をもっているため細粒度向きの処理である．また，装置が設置されるネットワークの状況に応じて優先される処理性能が異なる．このため 状況に応じて頻繁に回路を変更する必要がある．ただし，この再構成性の応用例は，確率的ニューラルネットワークの場合と異なり，限られたハードウェア・リソースの中における，処理可能なデータ量と処理速度とのトレードオフという観点から研究を行った．本適用例も，市販の FPGA を用いて行った実験と，その評価結果について述べる．さらに，非常に変革の速い分野で，頻繁に処理アルゴリズムが更新されるため，この点からも再構成可能集積回路に向けたアプリケーションである．

IP パケット・フィルタ装置は，一般にルータなどのネットワーク機器の中に実装されているもので，普段我々がその存在を意識するものではないが，ネットワークのトラフィック制御やセキュリティ面で非常に重要な役割を果たしている [19] ~ [22]．また，ネットワークの上流になればなるほどパケットのトラフィック量が増すため，より高速な処理が必要となる．逆に下流ほどトラフィック量は少ないが，詳細なパケットの制御が要求される．このため本来は要求に応じて適切なアーキテクチャが異なるが，現在のルータ装置は上流向けの最大スループットが性能の指標であり，そのために処理回路の ASIC 化も行われている．しかし，固定されたアーキテクチャで構成されたルータ装置は，トラフィック量が少ない場合には回路の稼働率が低下するため無駄が多い．そこで，再構成可能集積回路を用いた，RPCA (Reconfigurable Parallel Comparison Architecture) と RPCA で構成された性能可変 IP パケット・フィルタ装置を提案する．RPCA は比較演算器とレジスタの構成の変更を可能にするものである．RPCA で構成された IP パケット・フィルタはトラフィック量が多い場合にはスループットを重視し，少ない場合にはパケットの制御精度を重視するなど，要求に応じてアーキテクチャを変更し，常に最大の回路稼働率を維持することができる．本研究は IP パケット・フィルタ処理の核となる比較演算に着目し，FPGA の再構成可能という特徴を生かして，用途に応じて常に最適な比較演算器を構成する

アーキテクチャを提案した。

実装結果はそれぞれ、速度優先で 128IP アドレス処理可能な場合の処理時間が 120ns、データ量優先で 2,048IP アドレス処理可能な場合の処理時間が 1,360ns であった。これはそれぞれ約 4.27Gbps、377Mbps のデータ伝送速度で On-The-Fly フィルタ処理を実現できることを示している。従って、演算の種類が同一であっても、優先する処理性能が用途によって異なるアプリケーションでは、FPGA の再構成性を十分生かすことができることがわかった。また、実装は XILINX 社製の FPGA である XCV300E-6PQ240C[24] を用いて完全同期設計を行い、ベースクロックは 100BASE-TX 対応の PHY チップである RealTek 社製の RTL8201BL を使用した都合で 25MHz とした。

汎用 I/O インターフェースの実現

さらに、上記 1、2 のような演算回路ではなく、I/O インターフェースにおいて、その「再構成可能」という特徴を生かすアプリケーションについて提案し、その実装結果と評価結果について述べる。提案するメタ・I/O インターフェースは FPGA を用いて実現される、汎用 I/O インターフェースで、PC や PDA (Personal Digital Assistant) などにおける USB (Universal Serial Bus) に代わるものである。インターフェース回路は高速で、かつ細かなタイミング制御が要求されるため、MPU での実現を含む、ALU を用いた粗粒度構成には不向きである。一方 FPGA は、もともと RTL (Register Transfer Level) での設計に適したアーキテクチャであり、回路規模の面からも効率よく実現できる。また、コンシューマ向けの周辺装置の I/O インターフェースは、エンドユーザにより頻繁に変更が要求されるため、FPGA での実現に非常に適したアプリケーションである。

現在、パーソナル・コンピュータや PDA など携帯端末とその周辺装置間の I/O インターフェースは“標準規格”化されることで、装置の種類、メーカーが異なっても相互に接続が可能となっている。しかし、この“標準規格”は必ずしも全ての装置間のインターフェースとして適しているものではなく、“標準規格”に合わせるために余分な回路を必要とし、装置のサイズやコストなどを上げる原因となっている。また、装置毎に異なる要求性能に対応するために、さまざまな“標準規格”が作られることで、PDA などの実装面積が限られた装置に多くのインターフェース回路を用意しなければならなかったり、種類が増えることがユーザを惑わす原因にもなっている。“標準規格”の中でも EIA-232 (RS-232C) や USB は、多くの種類の装置間の I/O インターフェースとして利用されており、特に USB 2.0 の伝送速度はより多くの装置間インターフェースの要求仕様を満たしており、今後このような“標準規格”の統一化が進んでいくことで後者の問題は解決されるであろう。しかし、前者の“標準規格”が要求仕様と一致しない問題は解決されない。

そこで FPGA などの再構成可能集積回路を I/O インターフェースに応用した、メタ・I/O インターフェースを提案する。メタ・I/O インターフェースは、1つの共有コネクタで“標準規格”と同様の相互接続性を保証しつつ、各装置間に最適な“非標準的”なプロトコルを用いたインターフェースの提供を可能とする。本論文では、キーボード入力装置のインターフェースを例に、XILINX 社製の FPGA であ

る XC2S50E[26] を用いた試作機による実装評価を行い，その有効性を確かめた．

新しいハードウェア開発手法の実現

4 つ目の FPGA の効果的なアプリケーションとして，ハードウェアの開発手法に関する新たな提案を行う．これは，これまで述べてきた回路アーキテクチャに関するものではなく，本来の FPGA の利用方法を発展させたものである．フォト・マスク・ベースの LSI は，設計データをもとに半導体工場で製造されるが，FPGA は設計データを直接ロードすることで目的の回路を実現できる．ここで用いられる FPGA は，予め半導体工場で製造済みの集積回路である．このため，回路設計者は，あたかも設計 = 製造として開発を行うことができ，これがラピッド・プロトタイピング・ツールとしての FPGA の特徴である．ところが，ここでさらに FPGA の設計データを，コンピュータ間通信ネットワークを通じて配信することを考えると，従来物理的なシリコン・チップであった集積回路を，あたかもネットワークを通じて伝送可能であるかのように扱うことができる．このことは単にメーカ内での開発効率の向上だけでなく，開発層をメーカ以外に広げる，新しいアウトソーシング形態の実現を可能にする．本論文では，FPGA とインターネットを用いて，特に教育機関を含めた，回路設計のアマチュア層へ広げる手法を提案し，その実施例と評価結果について述べる．

パーソナル・コンピュータが一般に普及して 20 年余り，コンピュータを媒体としたソフトウェアの世界では，Linux，BSD などの OS をはじめ，アプリケーション・ソフトウェアに至るまで，さまざまなオープンソースソフトウェア [31] ~ [33] が流通している．このオープンソースソフトウェアにはさまざまな特徴があげられるが，中でも取り分け重要なのが，その多くが個人によって開発されているということである．一方，ハードウェアであるコンピュータ本体はその性質上，ソフトウェアのように個人で開発することは困難であった．しかし FPGA などの再構成可能集積回路の登場により，コンピュータ自体をソフトウェアのように，個人レベルでも容易に開発することが可能となりつつある．そこで FPGA とインターネットを用いて実現可能な，新しい開発手法として，フリー・IP (Intellectual Property) による開発を提案する．フリー・IP とは，従来ワイヤード・ロジックで実現されていたハードウェアを，FPGA に置き換えることで，物理的な LUT と，LUT に書き込まれる回路構成情報に分離し，その回路情報をフリー（法律上，自由に利用，配布可能）とすることで，ハードウェアの一部のオープンソースソフトウェア化を可能にするものである．このようなフリー・IP は主に海外の Web サイトで既にいくつか配信されている [34][35]．しかし，これらフリー・IP を利用することは，企業などでは信頼性，保証などの面から難しく，また，回路設計のアマチュアなど，個人レベルで利用するには技術的に敷居が高いなどの理由から困難である．これらの問題を生じる要因は，多くの場合フリー・IP の質にあるのではなく，ドキュメントの不足やアプリケーション例の不足，入手の容易性の低さなどにあると思われる．そこでこれらの問題を解決した，個人，或いは企業でも利用できるようフリー・IP ライブラリを提供することで，ハードウェア開発において，オープンソースソフトウェアと同様なバザール型の開発手法 [30] と，新たな開発層の確立をめざす．

実際に XILINX 社製の FPGA 向けにインターフェース回路を中心とした IP コア・ライブラリを開発し，インターネット上の Web サイト (<http://www.susubox.org/>) 上で配信した．その結果，より普及

を速めるためにはドキュメント整備の他に、回路設計のアマチュア層に魅力的な IP コアのアプリケーション例が必要であることがわかった。また現在、FPGA の開発ソフトウェアは容易に入手可能となったものの、FPGA を実装した開発プラットフォーム（ボード）が高価で入手が困難であるため、これを安価で大量に普及させる仕組みが必要であることがわかった。

結論

本論文では、再構成集積回路の中でも、LUT アレイ型のアーキテクチャを持つ FPGA を用いて、その「再構成可能」という特徴を生かせる、効果的なアプリケーションについて述べた。また、実際に 4 つのアプリケーションに適用し、その有効性を示した。

再構成可能集積回路の「再構成性」を生かせるアプリケーションは、従来の MPU で実現が困難な高速処理、サイズ、省電力性能を要求し、かつ ASIC などのフォト・マスク・ベース LSI で実現が困難な汎用性、TAT、コストが求められるものである。これは、

条件 1.) MPU での実現が困難なものとして

1. 高速処理を要求し、アルゴリズムの並列性が高い応用
2. 小さい回路規模、省電力性能を要求する応用

のいずれかを満たし、かつ

条件 2.) ASIC 化が困難なものとして

- a. 適切な処理アルゴリズム / プロトコルが未確定な分野での応用、
- b. 優先する性能が変化する応用、
- c. 回路構成情報のソフトウェアとしての利点を生かせる応用

のいずれかを満たすものである。

また、下記に示す具体的な適用例にて実証した。

満足条件	適用例
1-a.	確率的ニューラルネットワークを用いた画像認識装置
1-b, (1-a).	性能可変 IP パケット・フィルタ装置
1-a, 2-a, 1-c, 2-c.	汎用 I/O インターフェース
1-c, 2-c.	新しいハードウェア開発手法

確率的ニューラルネットワークはアルゴリズムの並列性が非常に高く、これを FPGA に実装して構築した画像認識装置は、認識対象によってはビデオレート時間で 98.3% 以上の認識精度を達成するなど、高い性能を実現することができた。また、このような高い性能を得るためには、認識対象に応じて前処理手法などを変更する必要がある。これはコスト面や TAT から考えると、ASIC 化するよりも FPGA

で実現した方が適したアプリケーションであるといえる。さらに、本研究で開発したシステムは、高速に PNN 計算が行えるため、特徴抽出が困難な認識処理において特に有用となる見通しを得た。

また、IP パケット・フィルタも非常に並列性の高いアルゴリズムを持ったアプリケーションであり、FPGA によって実現される IP パケット・フィルタ装置は、特に基幹ネットワーク～エンドユーザ・ネットワークを結ぶ、中流ネットワークにおいて、性能、コスト面から有用であることがわかった。さらにネットワークの状況に応じた処理速度と処理可能な IP アドレス数の優先順位の変更が可能な IP パケット・フィルタ装置が実現できる見通しを得た。

本論文で提案した FPGA によって実現される汎用 I/O インターフェースは、PC や PDA と周辺装置間の I/O インターフェースなど、従来 ASIC/ASSP 化されていたインターフェース回路に適用することで、そのプロトコルを標準規格でない、各メーカ独自のものを使用することを可能とした。これにより、周辺装置毎に最適なプロトコルを実現可能となり、周辺装置の高性能化が期待できる他、開発コストの低減や、周辺装置のユーザビリティの向上が可能となる見通しを得た。

4 つ目の適用例は他の適用例と異なり、回路の性能向上などに関するものではなく、ハードウェア開発における、開発効率の向上の可能性を示した。FPGA 上に構築する回路を、オープンソース・IP (オープンソース・ソフトウェア) としてインターネットを通じて配信し、新しいアウトソーシングの形態を構築することで、メーカにおけるハードウェア開発の効率を向上させるだけでなく、学生などを含む、回路設計のアマチュア層へ製品開発を広げることが可能であることを示した。

参考文献

- [1] 末吉 敏則, 稲吉 宏明, 飯田 全広, 田丸 啓吉, 今井 正治, 樋口 哲也, 特集: やわらかいハードウェア, 情報処理学会誌, Vol.40, No.8, pp.777-800, Aug. 1999.
- [2] K. Compton and S. Hauck, *Reconfigurable Computing: A Survey of Systems and Software*, ACM Computing Surveys, Vol.34, No.2, pp.171-210, June 2002.
- [3] Waugh T.C., *Field Programmable Gate Array Key to Reconfigurable Array Outperforming Supercomputers*, Proc. IEEE CICC, pp.6.6.1-6.6.4, 1991.
- [4] Brown S.D., Francis R.J., Rose J. and Vranesic Z.G., *Field-Programmable Gate Array*, Kluwer Acad. Pub., pp.13-43, 1992.
- [5] Brown S.D., *Field-Programmable Devices*, Stan Baker Associates, 1995.
- [6] Andre DeHon, *The Density Advantage of Configurable Computing*, COMPUTER, IEEE Computer Society, pp.41-49, April 2000.
- [7] 越智 裕之, *FPAA: フィールドプログラマブルアキュムレータアレイ*, 計算機アーキテクチャ, pp.97-102, Oct. 1997.
- [8] 越智 裕之, *FPAccA: フィールドプログラマブルアキュムレータアレイ-FPAccA model 1.0 チップの設計と評価*, 情報処理学会論文誌, Vol.40, No.4, pp.1717-1725, Apr. 1999.
- [9] アイピーフレックス株式会社, *DAPDNA-2 ダイナミック・リコンフィギュラブル・プロセッサ データシート*, <http://www.ipflex.com/jp/>, 2004.
- [10] NEC エレクトロニクス株式会社, *動的再構成プロセッサ*, <http://www.necel.com/ja/techhighlights/drp/>, 2004.
- [11] Duncan A. Buell, Jeffrey M. Arnold, and Walter J. Kleinfelder, *Splash2 : FPGAs in a Custom Computing Machine*, IEEE Computer Society Press, 1996.

- [12] Ernesto Novillo and Paul Lu, *A Case Study of Selected SPLASH-2 Applications and SBT Debugging Tool*, IEEE International Parallel and Distributed Processing Symposium (IPDPS'03), Apr.22-26, 2003.
- [13] Specht, D.: Probabilistic Neural Networks, *J.Neural Networks*, Vol. 3, pp. 109–118 (1990).
- [14] Specht, D.: Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification, *IEEE Trans. Neural Networks*, Vol. 1, No. 1, pp. 110–111 (1990).
- [15] Mao, K.Z., Tan, K.C. and Ser, W.: Probabilistic Neural-Network Structure Determination for Pattern Classification, *IEEE Trans. Neural Networks*, Vol. 11, No. 4, pp. 1009–1016 (2000).
- [16] Tian, B., R.Azimi-Sadjadi, M. *Senior Member*, IEEE. H.Vonder Harr, T. and Reinke, D.: Temporal Updating Scheme for Probabilistic Neural Network with Application on Satellite Cloud Classification, *IEEE Trans. Neural Networks*, Vol. 11, No. 4, pp. 903–920 (2000).
- [17] Tian, B., R.Azimi-Sadjadi, M.: Comparison of Two Different PNN Training Approaches for Satellite Cloud Data Classification, *IEEE Trans. Neural Networks*, Vol. 12, No. 1, pp. 164–168 (2001).
- [18] Minchin, G. and Zaknich, A.: A Design for FPGA Implementation of the Probabilistic Neural Network.
- [19] 山本 幹, “アクティブネットワーク”, *電子情報通信学会誌 TS*, Vol.86, No.7, pp.489–492, July, 2003.
- [20] A. G. Fragkiadakis, N. G. Bartzoudis D. J. Parish, and M. Sandford, “Active networking using programable hardware”, PGNet2003, June, 2003.
- [21] S. Li, J. Torresen, O. Soraasen, “Exploiting reconfigurable hardware for network security”, FCCM2003, 2003.
- [22] Y. H. Cho, S. Navab, and W. H. Mangione-Smith, “Specialized hardware for deep network packet filtering”, FPL2002, pp452-461, 2003.
- [23] XILINX Inc., “Virtex 2.5V Field Programmable Gate Arrays Data Sheet (DS003-1 v2.5)”, <http://www.xilinx.co.jp/bvdocs/publications/ds003.pdf>, April 2001.
- [24] XILINX Inc., “Virtex-E 1.8V Field Programmable Gate Arrays Data Sheet (DS022-1 v2.3)”, <http://www.xilinx.co.jp/bvdocs/publications/ds022.pdf>, July 2002.
- [25] XILINX Inc., *Spartan and Spartan-XL Families Field Programmable Gate Arrays (DS060 v1.7)*, <http://direct.xilinx.com/bvdocs/publications/ds060.pdf>, June 2002.
- [26] XILINX Inc., *Spartan-II 1.8V FPGA Family: Complete Data Sheet (DS077)*, <http://direct.xilinx.com/bvdocs/publications/ds077.pdf>, July 2004.
- [27] XILINX Inc., *Virtex-II Pro and Virtex-II Pro X FPGAs: Complete Data Sheet (DS083 v4.1)*, <http://direct.xilinx.com/bvdocs/publications/ds083.pdf>, November 2004.
- [28] ALTERA Corp., *Stratix Device Handbook, Volume 1 and 2 (S5V1-3.1, S5V2-3.1)*, http://www.altera.co.jp/literature/hb/stx/stratix_handbook.pdf, 2004.
- [29] ALTERA Corp., *Cyclone Device Handbook, Volume 1 and 2 (CSV1-1.4, CSV2-1.0)*, http://www.altera.co.jp/literature/hb/cyc/cyclone_device_handbook.pdf, 2003.
- [30] Eric S. Raymond, “The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary”, O'Reilly & Associates Inc., Oct., 1999.
- [31] 青山 幹雄, “オープンソースソフトウェアの現状”, *情報処理学会会誌*, Vol.43, No.12, pp.1319-1324, Dec. 2002.
- [32] 河原 正治, 大瀧 保広, “オープンソースから見た超流通技術”, *情報処理学会研究報告*, 電子化知的財産・社会基盤, Vol.2000, No.056, Jun. 2000.
- [33] Tim O'Reilly, “Lessons From Open-Source Software Development”, *Communications of the ACM*, Vol.42, No.4, pp.33-37, Apr. 1999.
- [34] “OPENCORES”, <http://www.opencores.org/>, 2004.
- [35] Jeung Joon Lee, “CMOSexod”, <http://www.cmosexod.com/>, 2004.