

1. アーキテクチャ基盤技術

7

再構成可能プロセッサ

天野 英晴

慶應義塾大学 情報工学科
hunga@am.ics.keio.ac.jp

はじめに

近年、コンフィギャラブルプロセッサ、リコンフィギャラブルプロセッサなど目的に応じて構成をカスタマイズすることが可能な「柔軟」なアーキテクチャが注目を集めている。このうち、コンフィギャラブルプロセッサは、応用に適した命令、構成、コプロセッサ機構を持つプロセッサを生成する手法で、チップ製造後に構成を変更することはできない。これに対してリコンフィギャラブルプロセッサは、ユーザプログラム可能なデバイスを用いて、チップ製造後に用途別に構成を作り込む方式である。本稿は、これらの柔軟性の持ったプロセッサアーキテクチャを、登場の背景、歴史的な経緯を含めて紹介する。なお、本稿の記述対象は、「1.6 タイルプロセッサ」、次号に掲載される「3.3 ユビキタス・コンピューティング時代の組み込みマイクロコンピュータ、SuperH と M32R」「3.4 動的再構成プロセッサ (DRP)」の記事と重複する部分があるが、「柔軟性の導入」という視点でこれらの記事との関連を明らかにしつつ紹介する。

Makimoto's Wave : 標準化とカスタマイズ

半導体チップは、標準化とカスタマイズの2つの方向性がある。標準部品は、さまざまな用途に用いることができるため、大量生産が可能であり、価格は下がる。しかし、用途が特定されてしまうと、標準部品は必ずしもその用途に最適化されてはいないため、無駄も大きいし、皆が同じものを使うと製品の競争力がなくなる。そこで、新しい応用分野がある程度広がると、その特定用途にカスタマイズしたチップが作られる。しかし、カスタマイズされたチップは設計コストが大きいため、製品数が少ないと元を取ることが難しい。このため、カスタマイズ

されたチップで生き残るものは少なく、生き残ったものはその分野での標準部品となっていく。図-1のように、標準化とカスタマイズの軸を揺れ動いて進んでいく、という考え方が Makimoto's Wave¹⁾ である。

ところが、一般的な半導体チップと異なり、計算機は本質的にプログラマブルであるため、機能的にカスタマイズする必要性が少なく、標準化が圧倒的に有利である。今まで、計算機がカスタマイズ方向に振れたのは、1970年代の終わりから1980年代のはじめまで、大型計算機からマイクロプロセッサの移行が始まりつつあったほんのしばらくの期間であった。この時代、容易にマイクロプロセッサを実現する方式としてマイクロプログラマブル方式が広く利用された。この方式は、計算機の制御部自体をマイクロ命令を実行する簡単小規模なマイクロ計算機で実現することで、制御される計算機の命令セットを自由にカスタマイズすることが可能であった。Smalltalk や Prolog などの高級言語に適した命令セットなど、目的用途別命令セットを入れ替え可能な計算機が登場し、実行中に自動的に命令を最適化する研究が行われた。しかし、この方式は RISC の台頭とともに姿を消し、計算機アーキテクチャの技術としては完全に失敗に

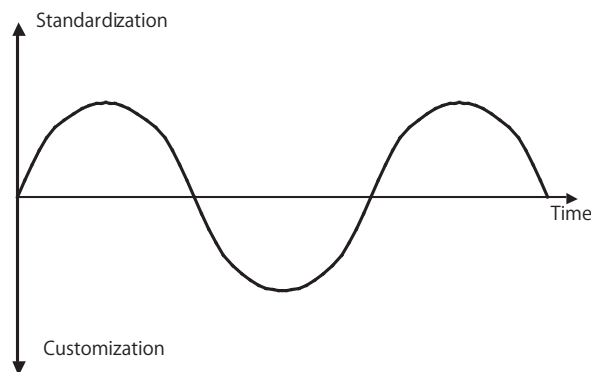


図-1 Makimoto's Wave

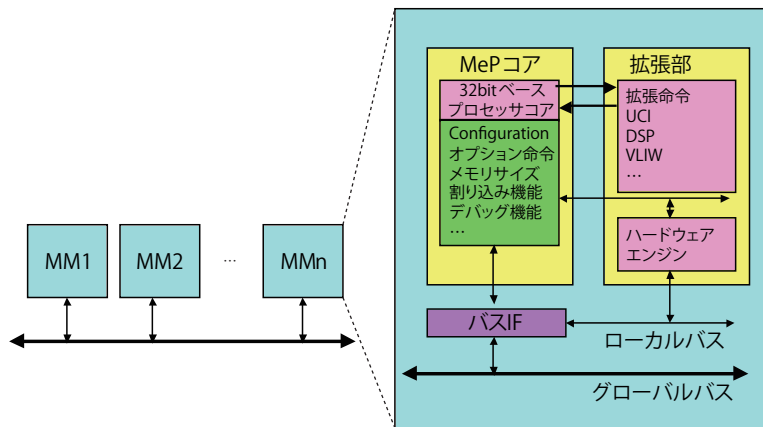


図-2 東芝MePの構成

終わった。この最も大きな原因は、マイクロコンピュータを用いて制御を行うために要するオーバーヘッドが、命令セットをカスタマイズすることによって受ける性能向上を上回ったためである。では、最近のコンフィギャラブル、リコンフィギャラブルプロセッサは、これらの可変命令セット型計算機とどのように違うのだろうか。

コンフィギャラブルプロセッサ

コンフィギャラブルプロセッサは、携帯電話、携帯端末、情報家電、ネットワーク制御等の組み込み用途が対象である。組み込み用途のプロセッサは、消費電力、コストの点で厳しい制約があるため、汎用の高性能マイクロプロセッサをそのまま利用することはできない。このため、次号の「3.3 ユビキタス・コンピューティング時代の組み込みマイクロコンピュータ、SuperHとM32R」で紹介されるように、主としてRISC型の命令セットを持った低電力かつ性能価格比に優れたプロセッサが開発され、利用されてきた。しかし、最近の組み込み機器の高性能化、高機能化により、画像、音声などの信号処理、認証などの暗号化複号化処理、通信用のコード化処理など、従来の組み込み用マイクロプロセッサの性能では追いつかない機能の実現が求められている。そこで、組み込みプロセッサに、特定機能のみを高速低電力で実行するハードウェア、あるいは特殊化された信号処理用プロセッサを組み合わせ、さらに、メモリ、標準I/Oを混載したSoC（System on a Chip）の開発が盛んになった。

ところが、単純にバスで接続して同一チップで組み込んだだけでは、組み込みプロセッサと専用ハードウェアの協調に要するロスが大きくなりやすい。この点を改善すべく、チップごとにデータ交換や同期の方法を考えると、チップ設計コストが増大してしまう。そこ

で、あらかじめ基盤となるプロセッサの大枠を決めておき、目的に合った専用命令を定義するなどのかたちで、専用ハードウェアをさまざまなレベルで組み合わせることを可能とするのがTensilica社のXtensaや東芝MePなどのコンフィギャラブルプロセッサのアプローチである。このアプローチでは、専用ハードウェアをCPUに組み合わせるといったよりは、専用目的化したプロセッサ自体を生成するという考え方をとる。基盤となるアーキテクチャを決めておき、これらをさまざまな形でカスタマイズするプラット

フォームアーキテクチャもこれと似た考え方と言える。図-2に東芝のMePを示す²⁾。MePでは、MePコアと呼ばれる拡張可能な32bitプロセッサコアをベースとし、専用ハードウェアやDSPや浮動小数演算器などの拡張部と組み合わせてMePモジュールを構成し、さらにこれらを複数接続してマルチプロセッサ構成をとるレベルまでをカバーする枠組みになっている。

このように、コンフィギャラブルプロセッサは、マイクロ命令による命令セットのカスタマイズと異なり、命令セットだけでなく、データバス、付加ハードウェアを含めたハードウェア構成自体がカスタマイズされる点が異なる。この点によって単純なプロセッサに比べてはるかに高い性能価格比、電力効率を実現することができる。また、このアプローチは、単純に標準化からカスタム化への波と考えられるわけではない。コンフィギャラブルプロセッサの枠組み、カスタマイズの手法自体は広くさまざまな目的に利用することができ、早期設計を可能として、設計コストを下げることに貢献する。すなわち、コンフィギャラブルプロセッサはカスタム化されたプロセッサを生成する枠組み自体を標準化した点で両方の方向性を持つアプローチといえる。

リコンフィギャラブルシステム

リコンフィギャラブルシステムは、書き換え可能なデバイス上に対象とするアルゴリズムを直接ハードウェア化して実行することにより、柔軟性と高速性の両立を狙ったシステムである。図-3にリコンフィギャラブルシステムの位置づけを示す³⁾。カスタマイズできるとはいえ、製造時に構成が決まってしまうコンフィギャラブルプロセッサと異なり、利用するたびに構成を変更することが可能で、デバイスによっては動作中に構成を変更してさまざまな処理を行うこともできる。ユーザが手

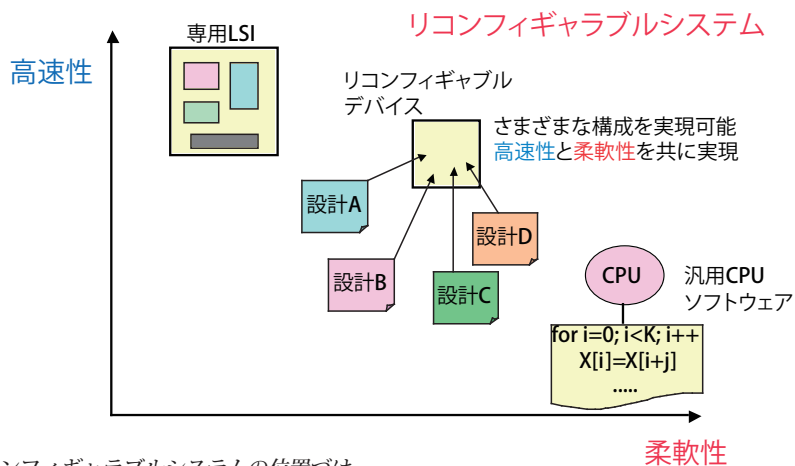


図-3 リンフィギャブルシステムの位置づけ

元で書き換え可能な大規模 PLD (Programmable Logic Device) である SRAM 型 FPGA (Field Programmable Gate Array) は 1990 年代はじめに登場し、以降 15 年間で驚異的な発展を遂げた。1991 年を 1 として 2004 年までに集積度は 200 倍、動作速度は 40 倍、消費電力は 1/40、コストは 1/500 になっている⁴⁾。これらのデバイスは 4 入力程度の小規模な LUT (Look Up Table) とフリップフロップから構成される基本論理素子をアレイ状に配置し、これらをプログラマブルスイッチにより接続した基本構成を持つ。LUT の内容とプログラマブルスイッチの接続情報は、構成データとして SRAM 上に格納され、この内容を入れ替えることにより、ユーザはさまざまな論理回路を同一チップ上に実現することができる。最近の FPGA は、基本論理素子のアレイ中に一定の間隔でデータ格納用の RAM や、乗算器、クロックコントローラ、場合によってはマイクロプロセッサなどを内蔵し、これらを組み合わせることによって、大規模なシステム全体を格納可能にしている。

リコンフィギャブルシステムは、この FPGA 複数個 (場合によっては 1 個)、外部メモリ、I/Oなどを互いに接続した構成を持ち、多くの場合はホストとなる PC や組み込み用のマイクロプロセッサと組み合わせて利用する。FPGA は、柔軟性を実現する構造上、その動作速度は制限されるため、単純に対象アルゴリズムをハードウェア化しただけではホスト上のソフトウェアに比べて大きな高速処理は困難である。そこで、パイプライン処理、SIMD、MIMD、シストリックアルゴリズム、データ駆動処理等さまざまな並列化技法を用いて高速化を行う必要がある。リコンフィギャブルシステムの柔軟性を活かすためには、対象とする問題ごとにその構造を最適化することが望ましい。しかし一方で、リコンフィギャブルシステム上で動作するハードウェア構造をユーザがハードウェア記述言語等で設計し直すのは負担が大きい上、変更した記述を FPGA の構成データに変換するには時間がかかる。このようなリコンフィギャ



図-4 ReCSiPボード

ブルシステムの利用上の問題点は、ある程度問題の範囲を定めれば、解決することができる。我々が北野共生システム、長崎大学と共同開発しているバイオインフォマティクス用アクセラレータ ReCSiP⁵⁾を用いた生化学シミュレータを例に解説する。

図-4に写真を示す ReCSiPは、図-5に内部構成を示すように Virtex-II Pro⁶⁾、多数のメモリ、PCI インタフェースを搭載したアクセラレータボード上で、細胞の解析などに用いる生化学シミュレーションを高速に実行するリコンフィギャブルシステムである。まず、ユーザは、SBML (System Biology Markup Language) という XML ライクの標準記述言語で、生化学シミュレーションの反応モデルを記述する。この記述を元にホスト PC 上のフロントエンドは、FPGA 上で動くハードウェアソルバーを自動的に生成する。このソルバーは、反応モデルに対応する連立常微分方程式を並列数値演算するハードウェアで構成され、Verilog-HDL 記述の形で自動的に生成されて、ボード上の Virtex-II Pro 上で動作する構成データに変換される。最後のステップでは、Xilinx 社の CAD 上で配置配線を行う必要があって相当の時間を要するが、この種の問題では、1つの反応モデルを繰り返して用いてさまざまな条件のもとでシミュレーショ

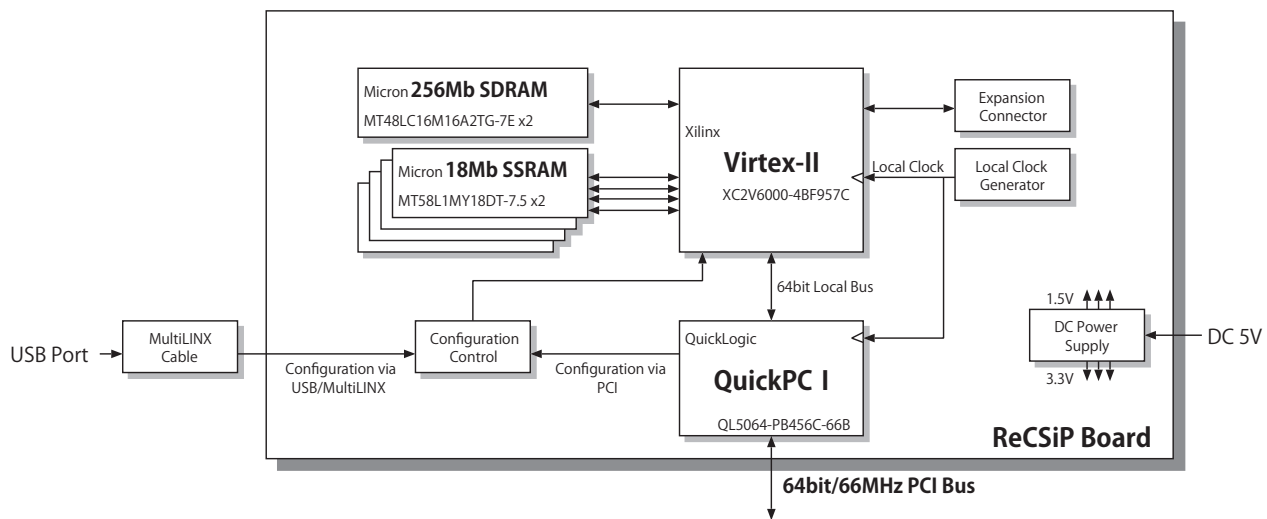


図-5 ReCSiPボードのブロック図

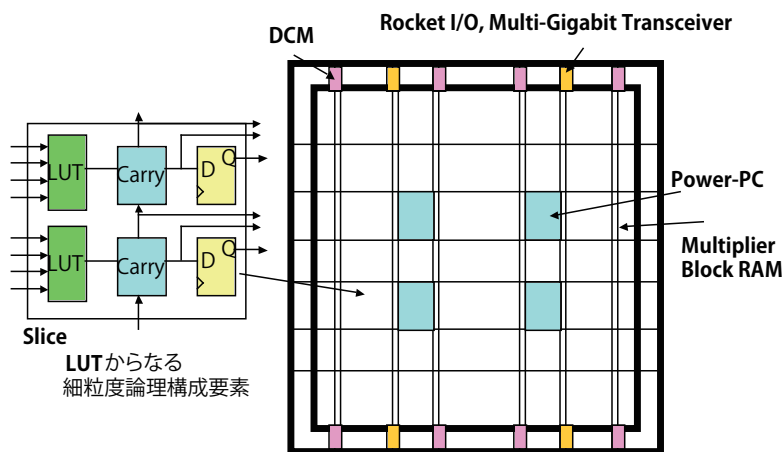


図-6 組み込みCPUとFPGAの混載例(Xilinx Virtex-II Pro)

ンを行うため、一度生成したソルバーを繰り返し利用することができる。このソルバーは、図-5の Virtex-II Pro上にマップされ、周辺のSSRAMをデータメモリとして用いて細胞シミュレーションを行う。ホストとのデータ交換は、アンチヒューズ型のFPGAであるQuickLogic上のPCIインタフェースを用いて行う。このPCIインタフェース、SSRAMインタフェースは、ライブラリの形で提供され、自動的にソルバーと接続される。ReCSiP上でのソルバーは、全体として深くパイプライン化された構造を持ち、さまざまな条件のもとでの独立したシミュレーションをマルチスレッドとして同時に走らせることが可能である。このように(1)問題に適した構造を自動生成する。(2)多数のパラメータのもとで同時に多数のシミュレーションを並列処理する。という2つの効果によって、ユーザにプログラム上の負担をかけずに、PCの数十倍の性能向上を達成することができる。

リコンフィギャラブルシステムは、ホストのソフトウェアで実行するよりもはるかに高性能な処理が必要とされるが、専用ハードウェアや専用プロセッサチップを新たに開発するほど数量が多く必要とされない分野、すな

わちニッチ分野で主に利用されてきた。ReCSiPが対象とするバイオインフォマティクス分野はその代表であり、他にもパターンマッチング、ネットワーク制御、大規模画像処理、ソフトウェア無線の基地局制御などのアプリケーションがこれに当たる。しかし、ここ1、2年で状況は大きく変化しつつある。最近のSRAM型FPGAは乗算器やマイクロプロセッサを搭載した大規模高性能なハイエンドの製品と、ローコストの製品に分化が進んでいる。前者の代表例であり、ReCSiPにも利用されているXilinx社のVirtex-II Proの構成を図-6に示す。基本的な論理素子であるSliceのアレイに対して一定間隔で、SRAMや乗算器が備えられており、中心部にはPowerPC、外縁部にはハイスピードリンクが内蔵されている。これらの内蔵されたハードウェアモジュールを利用することで、リコンフィギャラブルシステムは、浮動小数点演算や大規模データベース処理などのスーパーコンピューティングの世界に進出している。CRAY-XD1に代表される大規模システムや、単一基板でよりコスト対性能比に優れたシステムが登場している。

一方で、後者のローエンド製品を用いたリコンフィギャ

ラブルシステムは組み込み機器上で実際に利用されるようになってきている。ローエンド FPGA の大幅なコストダウンにより、組み込み製品であっても数量が一定以下ならば専用チップを開発するコストを考えると、FPGA 上にリコンフィギュラブルシステムとして実装した方が有利な状況になっている。最近のローエンド FPGA は、性能面でも専用チップにほとんど遜色がなくなっているが、唯一消費電力の点で不利である。このため、消費電力が特に重視される分野以外は、FPGA の利用が急速に進んでいる。

このように、リコンフィギュラブルシステムはニッチ分野から汎用分野へ飛躍する時期を迎えつつあると言える。リコンフィギュラブルシステムは個々のアプリケーションに対して徹底的にカスタマイズするアプローチである一方、利用するチップは完全に標準化されている点で、標準化とカスタマイズの両方の流れに乗っていると言える。

動的リコンフィギュラブルプロセッサ⁴⁾

標準部品である FPGA を利用したリコンフィギュラブルシステムは、柔軟性は高いが、その構成を変更するには msec オーダの時間を要する。このため、1つのアプリケーションを終了するまではその構成を変更しないのが普通である。これに対して動的リコンフィギュラブルプロセッサは、動的再構成、すなわちアプリケーションの動作中にその構造を変更することによって、同一の半導体面積を複数の目的に利用することを可能とし、これによってコスト対性能比の向上を狙っている。リコンフィギュラブルプロセッサは、FPGA と違って、その基本構成要素は、図-7 に示す演算器、レジスタ、マルチプレクサ等から構成される。これは、以下の利点に基づいている。(1) 動的リコンフィギュラブルプロセッサが対象とするメディア処理、ネットワーク制御などでは、演算処理が多いため、LUT よりも高速で低コストである。(2) 構成データは、演算の種類、演算器間の接続、構成要素間の接続の情報となり、FPGA に比べて量が少なく済むので、動的再構成が容易である。(3) C レベル設計と適合している。(4) 配置配線が LUT ベースの FPGA に比べれば容易であり、特許の問題も生じにくい。このため、動的リコンフィギュラブルプロセッサは、全体として、CPU のデータパスを 2 次元アレイ状に並べた構造を持つ場合が多くなる。このため、これらのシステムは「プロセッサ」、構成要素は「PE (Processing Element)」と呼ばれ、構成データのことも「命令」と呼ばれる場合がある。

動的リコンフィギュラブルプロセッサの最大の特徴である動的再構成には図-8 に示す手法がある。(a) は、複数の内部メモリから構成データを順にバスで配ってい

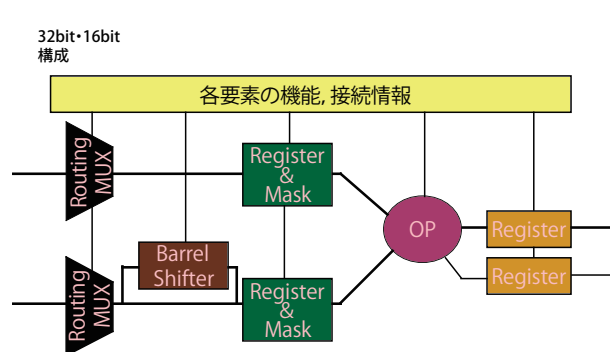


図-7 動的リコンフィギュラブルプロセッサの構成要素の例 (Chameleon CS2112)

く方法である。PACT 社の XPP、Elixent 社の DFA などがこれに相当し、再構成には数十から数百クロックを要する。(b) は、構成データをそれぞれの PE ごとに備えて、これに対するポインタをブロードキャストする方法であり、NEC エレクトロニクス社の DRP-1、アイピーフレックス社の DAPDNA2 が用いている。この方法を用いると、ポインタを切り替えることで、構成の変更は 1 クロックで可能であるが、構成データを保持するコストは増加する。この方式をマルチコンテキスト方式と呼び、それぞれの構成のことをハードウェアコンテキストと呼ぶ。コンテキストへのポインタの切り替えは、状態遷移テーブルやコンテキスト制御用カウンタなどを持つ専用のコントローラによって制御される。最後に (c) は、それぞれの PE が構成データを自律的にメモリから取ってくる機構を持つ場合で、この場合 PE は単なるデータパスではなく、プログラムカウンタを持った本格的な CPU となり、この方式は動的リコンフィギュラブルプロセッサというよりはマルチプロセッサである。

しかし、Quicksilver 社の ACM や PiCoChip では、この命令フェッチ機構がきわめて簡単なものであり、しかも、全体的にプログラムカウンタを切り替えることによってコンテキストを切り替える機構を持つため、動的リコンフィギュラブルシステムときわめて似た性質を持っているため、一緒に議論される場合もある。この種のプロセッサは本号に掲載される「1.6 タイルプロセッサ」で、詳しく紹介される。(b) のマルチコンテキスト型は、データパスが 2 次元アレイとなり、命令メモリが分散されて小容量となった VLIW (Very Long Instruction Word) 型と考えることもできるし、通常の命令フェッチをコンテキスト切り替えと考えると、普通のプログラム格納型プロセッサというのは命令ごとに構成を切り替えている並列性の限定された動的リコンフィギュラブルシステムと考えることもできる。このような視点からこれらのシステムを分類したのが図-9 である。この図は基本構成要素の大きさ (おおざっぱなゲート数)、並列

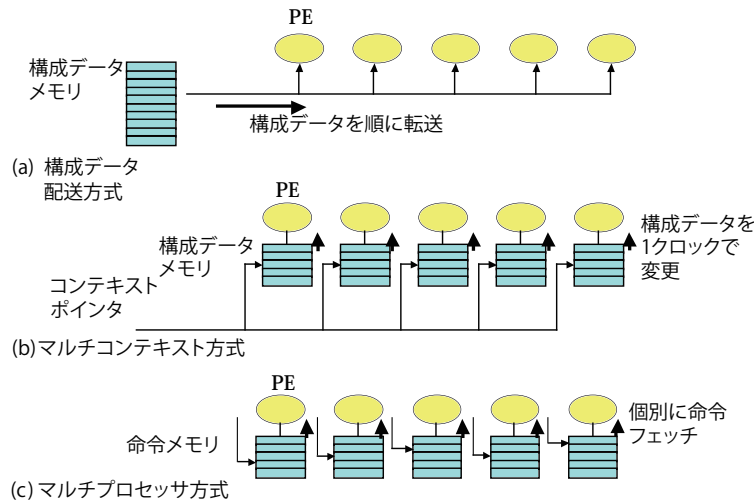


図-8 動的再構成の方式

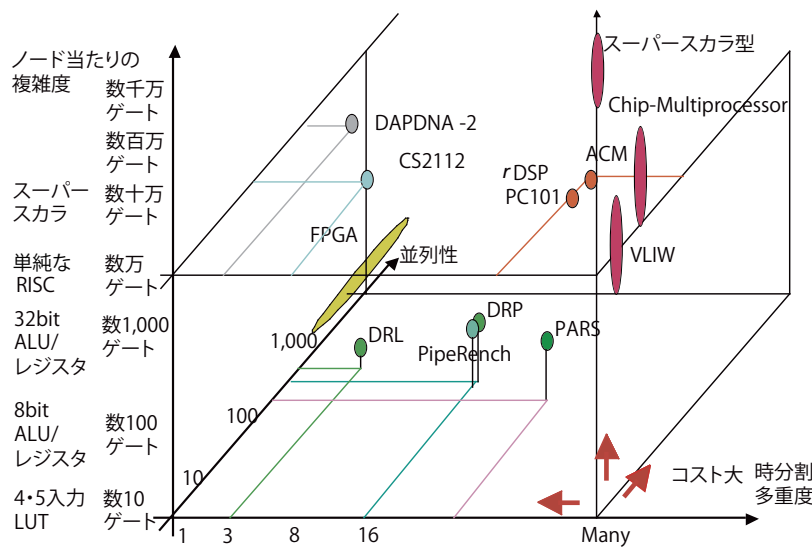


図-9 動的リコンフィギャラブルプロセッサの位置づけ

性（個数），そして，時分割多重度を軸にとったものである。時分割多重度は1つの問題を解く際，データバス構成を何回変更するか，という視点であり，まったく切り替えないFPGAは1となり，命令実行ごとにデータバス構成を切り替えると考えた場合のプログラム格納型プロセッサは「非常に多数」ということになる。

個々のシステムについては「1.6 タイルプロセッサ」 「3.4 動的再構成プロセッサ (DRP)」で詳細が紹介されるため，ここでは深くは議論しないが，マルチプロセッサ，VLIWマシンを含めて，これらのシステムが連続的に分布していることがご理解いただけるであろう。

おわりに

柔軟性を武器として標準化とカスタム化の両立を狙うアーキテクチャを紹介した。今回紹介した領域の研

究開発ではFPGAデバイス以外は，日本は世界でトップレベルであり，活発な研究が行われている。しかし，CAD，デバイス領域の研究者に比べてアーキテクチャ領域の研究者が少ない傾向にある。従来のプロセッサアーキテクチャから考えると若干奇異な感じを受けるが，偏見を持たずに研究に参加してくれる計算機アーキテクトが増えることを願っている。

参考文献

- 1) Makimoto, T.: The Rising Wave of Field Programmability, Proc. of Field-Programmable Logic and Applications, pp.1-6 (2000).
- 2) 宮森: コンフィギャラブルプロセッサ MeP とその SoC 開発事例, 信学報 VLD2004-115 (2005).
- 3) Toelandts, W.: FPGAs and Era of Field Programmability, Proc. of Field-Programmable Logic and Applications, pp.1-2 (2004).
- 4) 末吉, 天野: リコンフィギャラブルシステム, オーム社 (2005).
- 5) Osana, Y., Fukushima, T., Yoshimi, M. and Amano, H.: An FPGA-Based Acceleration Method for Metabolic Simulation, IEICE Trans. on Inf. & Syst., E87-D, No.8, pp.2029-2037 (2004).
- 6) <http://www.xilinx.com>

(平成 17 年 8 月 12 日受付)