再構成可能並列プロセッサと 知能ロボット制御への応用

藤岡 与周^{*1} 亀山 充隆^{*2} 苔米地 宣裕^{*1}

Reconfigurable Parallel Processor and Its Application to Intelligent Robot Control

Yoshichika Fujioka^{*1}, Michitaka Kameyama^{*2} and Nobuhiro Tomabeche^{*1}

In the sensor feedback control of intelligent robots, the delay time must be reduced for a large number of arithmetic operations. In addition to many multiply-additions, the arithmetic operations such as absolute value calculation and maximum value selection are often used in the dynamic control of robot manipulators. To reduce the delay time for multi-operand arithmetic operations, the architecture of the reconfigurable parallel processor is proposed. In each processor element, a switch circuit is used to change the connection between the multipliers, adders and arithmetic units (AUs), so that the multi-operand AUs having desired numbers of operands can be reconfigured. Since the data transfer is accomplished by the direct connection between the multipliers, adders and AUs, the overhead for data transfer is reduced. The chip evaluation based on 0.8 [μm] CMOS design rule shows that the delay time for differential inverse kinematics computation and inverse dynamics computation of a six-degrees-of-freedom manipulator become 1.7 [μsec] and 11.5 [μsec], respectively.

Key Words: Dynamic Control, VLSI Processor, Reconfiguration, Multi-Operand Arithmetic Operation, Delay Time

1. ま え が き

自律的動作を目指した知能ロボット実現のためには、外界の環境の認識から柔軟なマニピュレータ制御に至るまで多種多様な処理が必要となる [1]。これらの一連の処理は情報の流れが直列的であると共にセンサフィードバックが多いため、個々の処理に対してスループットの向上のみではなくセンサ信号が入力されてから制御出力が計算されるまでの、演算遅れ時間が小さい VLSI プロセッサの開発が重要な問題となる [2]~[11]。

これまで、逆運動学演算や逆動力学演算などの専用 VLSI プロセッサアーキテクチャ [3]~[7] が提案されているが、これらは知能ロボット制御に必要な個々の演算を高速に実行するものである。これに対し、種々の演算の遅れ時間を減少可能な VLSI プロセッサが開発されれば、所望の制御アルゴリズムを用いた知能ロボット制御を容易に実現できる。

従来、ロボット制御にはマイクロプロセッサやデジタル信号処理プロセッサ (DSP) など [12]~[16] が主に用いられてきたが、冗長マニピュレータなどの動的制御を実時間で行うことは計算量が多いため困難である。演算性能を向上させるため複

数個の汎用マイクロプロセッサや DSP を要素プロセッサ (PE) とした並列プロセッサを用いた場合、PE の個数が増大すると演算遅れ時間に対する PE 間のデータ転送時間の割合、すなわち通信オーバーヘッドが増大し演算器の稼働効率が減少するため、演算遅れ時間の減少が困難であるという問題があった。

ロボット制御に必要な種々の入力数の多入力積和演算については、再構成可能並列プロセッサを用いて演算遅れ時間を減少できる [8]~[10]。本論文では、微分逆運動学演算などに必要となる、絶対値計算や最大値/最小値選択などを含んだ種々の入力数の多入力算術演算の遅れ時間を減少するためにアーキテクチャを拡張した、再構成可能並列プロセッサを提案している。この再構成は、複数個の PE を用いて種々の入力数の積和演算器や最大値/最小値選択器など、所望の入力数の多入力算術演算器をスイッチ回路の切換えによりダイナミックに構成できることを意味している。PE 間通信が、スイッチ回路の切換えによる乗算器、加算器、算術演算器 (AU) などの各入出力間の直接接続に帰着されるため、PE 間通信のオーバーヘッドが減少し乗算器、加算器や AU の稼働効率が向上する特長がある。また、多入力算術演算結果の転送に要するクロックサイクルを減少するため、再構成された多入力算術演算器の出力を直接他の多入力算術演算器入力に接続するスイッチ回路を各 PE に備えている。知能ロボット制御に必要な制御アルゴリズムに応じて、二次元構造並列アーキテクチャか多重バス結合方式並列アーキテクチャのいずれかを適切に選択することにより、

原稿受付 1994 年 9 月 15 日

^{*1} 八戸工業大学工学部

^{*2} 東北大学大学院情報科学研究科

^{*1} Faculty of Engineering, Hachinohe Institute of Technology

^{*2} Graduate School of Information Sciences, Tohoku University

計算量が多い種々の制御アルゴリズムの演算遅れ時間減少に有用であることを明らかにしている。

0.8[μm] CMOS設計ルールにより、本 PE のレイアウト設計を行った結果、PE のチップ面積は 8.9 [mm] × 12.4 [mm] と現在の VLSI 技術で十分実現可能である。また、1 回の乗算サイクルを 1 クロックサイクルに定めると、PE のクロック周波数は 10 [MHz] 程度になる。さらに、マイクロプログラムレベルでの並列プログラミングによる評価の結果、6 自由度マニピュレータの動的制御に必要な基本的演算である微分逆運動学演算および順/逆運動学・逆動力学演算をそれぞれ 1.7 [μsec] と 11.5 [μsec]、すなわち DSP を用いた並列プロセッサと比較して数分の 1 程度の演算遅れ時間で実行できる。

2. 再構成可能並列プロセッサアーキテクチャ

計算量の多いロボット制御には、複数個の DSP からなる並列プロセッサが多く用いられている。しかし、DSP 間のデータ転送のためにハンドシェークなどのソフトウェア手続きが必要であり、通信のオーバーヘッドが大きい。演算遅れ時間を減少することが困難であるという問題がある。ロボット制御のための種々の多入力算術演算を高並列に実行し演算遅れ時間を減少するため、Fig. 1 に示す再構成可能並列アーキテクチャを提案する。各 PE はそれぞれ二つの乗算器、加算器、AU とそれぞれ一つのローカルメモリ、スイッチ回路を備えている。このスイッチ回路は、乗算器、加算器、AU 間の直接接続を切換えて、所望の個数の乗算器、加算器や AU を含む多入力算術演算器を再構成するためのものであり、リング状に結合されている。ここで、AU は絶対値演算器や最大値/最小値選択器などとして動作する。一例として、6 個の乗算器、3 個の加算器および 2 個の AU を含む多入力算術演算器を再構成する場合を Fig. 2 に示す。このように、スイッチ回路の接続を切り換えることにより、所望とする個数の乗算器、加算器や AU を含む多入力算術演算器を備えた PE' を複数個の PE で再構成し、多入力算術演算の並列処理をソフトウェアによる通信手順なしに高速に行うことができる。したがって、PE 間通信のオーバーヘッドが大幅に減少し、乗算器や加算器などの稼働効率が向上するため、知能ロボット制御に必要な種々の多入力算術演算の遅れ時間を減少できる。

Fig. 3 は再構成可能並列プロセッサにおいて、次式により表される多入力算術演算を行う様子を示している。

$$g_i = |a_i b_i + c_i d_i| \quad (i = 1, \dots, 6) \quad (1)$$

$$h = \max\{g_1, \dots, g_6\} \quad (2)$$

Fig. 3 に示すように、ステップ 1 で乗算が実行された後、ステップ 2 で乗算結果 e_i, f_i の加算、絶対値 g_i の計算および g_i の最大値 h の選択が実行される。ここで、 g_i を求めるために用いられた AU は絶対値演算器として、その他の AU は最大値選択器として動作している。乗算器の演算時間を 1 クロックサイクルと定義すると、加算器と AU の演算時間が乗算時間に比べて十分小さい場合には、各ステップがそれぞれ 1 クロックサイクル内に実行される。したがって、式 (1) と式 (2) で表される多入力算術演算は 2 クロックサイクルで実行される。

Fig. 1 Reconfigurable parallel processor architecture

Fig. 2 Reconfiguration for a multi-operand arithmetic unit

一方、再構成を行わない並列プロセッサにおいてこの積和演算と算術演算を行う場合を Fig. 4 に示す。中間結果 g_1, g_3, g_5, k_1, k_3 を PE 間で通信する必要があるため、式 (1) と式 (2) で表される多入力算術演算は 6 ステップで実行される。また、PE 間通信に少なくとも 1 クロックサイクル必要であることから、式 (1) と式 (2) で表される積和演算と算術演算の実行には 8 クロックサイクル以上必要となる。したがって、提案する再構成可能並列プロセッサと比較すると、演算遅れ時間は 4 倍以上となる。

計算量の多いデジタル制御には、Fig. 5 に示すような複数個の DSP により構成された並列プロセッサが用いられることが多い。多くの DSP は、乗算器、加算器、パラレル入出力ポートを各一つずつしか備えていないため、各 DSP が単一共有バス結合された構造となっている。また、Fig. 3 および Fig. 4 の並列プロセッサと同数の乗算器と加算器を備えるよう、Fig. 5 に示す並列プロセッサは 12 個の DSP により構成されている。この並列プロセッサを用いて式 (1) と式 (2) で表される多入力算術演算を行う場合には、各 DSP にそれぞれ 1 個の乗算器と加算器しか備えていないため、さらに中間結果 e_i ($i = 1, \dots, 6$) を PE 間で通信する必要がある。また、各 DSP が単一共有バス結合されているため、同時に一つの中間結果しか転送できない。したがって、式 (1) と式 (2) で表される算術演算を行うためには少なくとも 17 クロックサイクル必要となる。提案する再構成可能並列プロセッサを用いる場合と比較すると、演算遅れ時間は 8 倍以上となる。

以上より、提案する再構成可能並列プロセッサを用いることにより、複数個の DSP により構成された並列プロセッサと比

Fig.3 Execution of the arithmetic operation in the reconfigurable parallel processor

Fig.4 Execution of the arithmetic operation with communication between PEs

べて演算遅れ時間を大幅に減少できることが明らかである。さらに、多入力算術演算器の再構成のための接続切換えに加えて、ローカルメモリと乗算器、加算器、AU との接続切換え、および各 PE のローカルメモリ間の接続切換えのためのスイッチをスイッチ回路内に備えることにより、PE 間のデータ転送を小さい遅れ時間で行うことができる。

Fig.6 は、行列/ベクトル演算などを行うための二次元構造再構成可能並列プロセッサを示している。各 PE は縦方向、横方向にそれぞれリング状に結合されており、縦方向と横方向の多入力算術演算器の再構成ができるようになっている。また、

Fig.5 Parallel processor composed of conventional DSPs

縦方向、横方向の PE をそれぞれ共有バス結合することにより、行列/ベクトル演算で必要となる行および列方向のブロードキャ

Fig. 6 Two-dimensional reconfigurable parallel processor

スト通信が並列にできるようになっている。さらに、すべての PE に対するブロードキャスト通信を行うために、すべての PE を共有バスで結合している。したがって、各 PE には、共有バス結合用に 3 本と、多入力算術演算器の再構成用に 4 本の計 7 本の入出力ポートが備えられている。知能ロボット制御では行列/ベクトル演算が多く用いられるため、このようにメッシュ結合プロセッサを構成することにより、効率よく PE 間のデータ転送を行うことができる。また、各 PE の入出力ポート間の接続を制御アルゴリズムに応じて決定することにより、容易に専用並列プロセッサを構成することができ、種々のロボット制御用に広く応用できる。

3. PE チップの構成

Fig. 7 に PE の構成を示す。多入力算術演算器を再構成するためのスイッチ回路の切り換えを各乗算サイクルごとに行うため、もし、再構成された多入力算術演算器の加算時間と算術演算時間が乗算時間より大きくなる場合には、加算器と AU の部分をいくつかの部分にパイプラインレジスタで分割し、各部分の演算時間が乗算時間以下となるように構成されている。

PE 内部のデータ転送を容易にするため、ローカルメモリは Fig. 7 に示すように多ポートメモリ構造となっている。ローカルメモリのメモリアクセス時間を減少するため、6 個の 2 ポートメモリを組み合わせており、二つの書き込みポートと 12 個の読み出しポートを備えた構成となっている。ただし、書き込みと読み出しが同時に行えないため、1 クロックサイクルの前半でローカルメモリへの書き込みを行い、後半で読み出しを行うことにより、1 クロックサイクルで書き込みと読み出しを行うことができるように構成されている。ローカルメモリへの書き込みと読み出しがそれぞれ半クロックサイクルで行われるため、メモリアクセスも含めた多入力算術演算の遅れ時間は 3 クロックサイクルとなる。多くの DSP は、各 1 回の乗算と加算からなる積和演算の実行にメモリアクセスを含めて少なくとも 4 クロックサイクル以上必要であり、また積和演算に含まれる乗算の個数が増えるほど演算遅れ時間が増加する。これに対し、提案する再構成可能並列プロセッサは多入力算術演算の演算遅れ

□: Pipeline register

Fig. 7 Structure of the PE

時間が小さい。さらに、各 PE のローカルメモリ間はスイッチ回路により直接接続されるため、各 PE のローカルメモリ間のデータ転送を 1 クロックサイクルで行うことができる。DSP を用いた Fig. 5 に示すような並列プロセッサでは、ローカルメモリの書き込み、読み出しにそれぞれ 1 クロックサイクルずつ必要であるとともに、DSP 間のデータ転送のためのハンドシェイクなどのソフトウェア手続きに数クロックサイクル必要であることが多い。これと比較すると、提案する再構成可能並列プロセッサは多入力算術演算の演算遅れ時間が小さいのに加えて、PE 間のデータ転送も小さい遅れ時間で行うことができる。すなわち、PE 間の通信オーバーヘッドが小さい。さらに、再構成された多入力算術演算器の出力を他の多入力算術演算器入力に直接接続し、多入力算術演算結果の転送に要するクロックサイクル数を減少するため、乗算器、加算器、AU はそれぞれ 7 本のバスとスイッチ回路を通じて直接接続できるようになっている。以上の結果、乗算器、加算器、AU の稼働効率が向上し、種々の知能ロボット制御に必要となる多入力算術演算を小さい遅れ時間で実行することができる。

本 PE の 0.8 μm CMOS 設計ルールによるレイアウトを Fig. 8 に、また諸元を Table 1 に示す。乗算時間を約 100 [ns] とすることができるため、本 PE のクロック周波数は約 10 [MHz] となる。再構成された多入力算術演算器に直列に挿入されるスイッチの数を減らし演算遅れ時間を減少するため、スイッチ回路をクロスバススイッチにより構成している。通常クロスバススイッチは入出力数が増大すると回路規模が非常に大きくなるが、チップレイアウトによる面積評価を行った結果、スイッチ回路の大きさは 2.4 [mm] $\times$ 2.7 [mm] であり、また PE チップの大きさは 8.9 [mm] $\times$ 12.4 [mm] となる。したがって、種々の入力数の多入力算術演算器を再構成可能な PE を現在の VLSI 技術により十分シングルチップ化可能である。

4. 知能ロボット制御への応用

ロボットマニピュレータの高速かつ高精度な軌道制御や力制御などにおいては、逆運動学演算の代わりにヤコビ行列の逆行列を用いた微分逆運動学演算がしばしば必要となる [17]。ま

Fig. 8 Chip layout of the PE

Table 1 Features of the PE

Technology	0.8- μ m two-level-layer CMOS
Clock frequency	10 [MHz]
Chip size	12.4 [mm] $\times$ 8.9 [mm] (except I/O)
Arithmetic	32bit fixed-point representation
Local memory	256 $\times$ 32 [bit] two-port RAM $\times$ 6
Multiplier	32 $\times$ 32 [bit] array type $\times$ 2
Adder	32bit binary lookahead carry $\times$ 2
Switch circuit	32bit 24 $\times$ 17 crossbar switches
I/O data port	32 [bit] $\times$ 7
Control	VLIW

た、関節自由度の多い冗長マニピュレータなどの動的制御においては、ヤコビ行列の擬似逆行列を用いた微分逆運動学演算が必要となる。これらの演算においては、ヤコビ行列の各行および各列の要素へのデータ転送が膨大となる。この場合、Fig. 6に示すような二次元構造再構成可能並列プロセッサを用い多入力算術演算結果の転送を効率よく行うことにより、演算遅れ時間を減少できる。さらに、ニュートン・オイラー法による逆動力学演算などは計算量が少ない反面、多入力積和演算結果の転送が複雑となる。この多入力積和演算結果の転送を少ないクロックサイクル数で行うため、Fig. 11に示すように各PEは多重バスで結合されている。これにより、複数個の多入力積和演算結果の転送を1クロックサイクル程度で効率よく行うことができ、逆動力学演算の遅れ時間を減少できる。このように、知能ロボット制御のための種々の制御アルゴリズムに応じてPE間の通信ネットワークを選択することにより、演算遅れ時間を減少することを考える。

4.1 微分逆運動学演算

手先の微小変位から関節の微小変位をヤコビ行列の逆行列や擬似逆行列を用いて求める座標変換、すなわち微分逆運動学演算には、計算量の少ない連立一次方程式の数値解法が用いられることが多い。連立一次方程式 $Ax = b$ ($A = \{a_{ij}^{(1)}\}$, $b = \{b_i^{(1)}\}$, $x = \{x_i\}$, $i, j = 1, \dots, m$) の数値解法にはガウスの消去法やガウス・ジョルダンの掃出し法などが知られているが、これらと比較して並列性が高く積和演算が主体の除算後置形連立一次方程式数値解法アルゴリズムが提案されている [8]。

Fig. 9 Subsequent-division algorithm for solution of simultaneous linear equations

Fig. 9に示すように、本アルゴリズムはピボットと呼ばれる $a_{ii} \neq 0$ を選択する部分、積和演算を行う部分と最後の除算を行う部分とから構成されている。従来の数値解法アルゴリズムはピボット選択、除算、積和演算を繰り返し実行するのに対し、本アルゴリズムは除算をまとめて最後に行うことができる。このため、演算回数の少ない除算をホストプロセッサで実行するものとする、本アルゴリズムは積和演算が主体で並列性が高いという特長を有する。

本アルゴリズムから除算を除いた部分は a_{ji} ($j = i, \dots, m$) の中から $a_{ji} \neq 0$ を選択し j 行目と i 行目の各要素を交換する部分、すなわちピボット a_{ii} を選択する部分と、積和演算を行う部分の繰り返しとなる。積和演算を行う部分は再構成可能並列プロセッサを用いることにより演算遅れ時間を減少できる。また、ピボットを選択する部分では、解 x の演算誤差をできるだけ少なくするため、絶対値がもっとも大きい $a_{ji} \neq 0$ を a_{ii} として選択することが望ましい。このピボット選択には、積和演算、絶対値計算や最大値選択を含む種々の多入力算術演算が必要となる。再構成可能並列プロセッサを用いる場合、AUを絶対値演算器または最大値選択器として動作させ、一例としてFig. 3に示すように多入力算術演算器を再構成することにより、積和演算、絶対値計算と最大値選択からなる多入力算術演算の演算遅れ時間を減少できる。したがって、高精度な微分逆運動学演算の演算遅れ時間を減少できる。これに対し、DSPや汎用プロセッサなどで実行する場合には、最大値選択を条件分岐の繰り返しにより実現する必要があるため、演算遅れ時間が大きい。

本アルゴリズムを用いた場合の多入力積和演算結果の転送は次のようになる。まず、 $m \times (m+1)$ 個のPEを備えた二次元構造再構成可能並列プロセッサの各PEに、 $m \times (m+1)$ 行列

[Ab] の各要素が格納されているものとする。選択されたピボット $a_{ii}^{(i)}$ を PE_{jk} に転送するためには、すべての PE を結合している共通バスを用いることができる。また、 $a_{ji}^{(i)}$ を j 行目の各 PE_{jk} に転送するためには、横方向の PE を結合している共通バスを用いることができる。さらに、 $a_{ik}^{(i)}$ を k 列目の各 PE_{jk} に、 $b_i^{(i)}$ を $m+1$ 列目の各 $PE_{j,m+1}$ にそれぞれ転送するためには、縦方向の PE を結合している共通バスを用いることができる。ただし、 $j \neq i, k > i$ である。このように、二次元構造再構成可能並列プロセッサを用いることにより、本アルゴリズムにおける多入力算術演算結果の転送を効率よく行うことができる。

ロボットマニピュレータの自由度が 6 である場合 $m = 6$ となる。6 行 $\times$ 7 列の二次元構造再構成可能並列プロセッサにより除算後置形連立一次方程式の数値解を求める並列プログラムをマイクロプログラムレベルで作成し演算遅れ時間を評価したところ、除算以外の演算遅れ時間は $1.7 [\mu\text{sec}]$ となる。これに対し、42 個の DSP を用いた場合の演算遅れ時間は $12.0 [\mu\text{sec}]$ となるため、再構成可能並列プロセッサを用いることにより演算遅れ時間を約 7 分の 1 に減少できる。

冗長マニピュレータの微分逆運動学演算には、ヤコビ行列の逆行列のかわりに擬逆行列が用いられる。擬逆ヤコビ行列を計算しそれを用いて微分逆運動学演算を行う方法は計算量が多いため、Fig. 10 に示すような連立一次方程式の数値解法に基づく積和演算主体の微分逆運動学演算アルゴリズムが提案さ

れている [8]。ここで、 $6 \times n$ 行列 $J(\theta)$ はヤコビ行列であり、 n 次元ベクトル $k, \theta, \delta\theta$ は任意ベクトルと n 自由度冗長マニピュレータの関節変位、関節微小変位をそれぞれ表している。また、6 次元ベクトル δp は作業座標空間における手先の微小変位を表している。さらに、 $J_1(\theta), J_r(\theta), J_2(\theta)$ はそれぞれ階数が r の $6 \times r$ 行列、 $r \times r$ 行列、 $r \times n$ 行列を表しており、 r はヤコビ行列 $J(\theta)$ の階数である。本アルゴリズムは、連立一次方程式の数値解法に除算後置形連立一次方程式数値解法アルゴリズムを用いることにより、並列性が高く計算量が少ないという特長を有する。除算をホストプロセッサで行うこととすると、ロボットマニピュレータの自由度が 7 である場合、6 行 $\times$ 7 列の二次元構造再構成可能並列プロセッサを用いることにより、本アルゴリズムの除算以外の部分の演算遅れ時間を $4.9 [\mu\text{sec}]$ とすることができる。これに対し、42 個の DSP を用いた場合の演算遅れ時間は $28.4 [\mu\text{sec}]$ となるため、演算遅れ時間を約 6 分の 1 に減少できる。

4.2 順/逆運動学演算と逆動力学演算

ニュートン・オイラー法による逆動力学演算などは、計算量が少ない反面、多入力算術演算結果の転送が複雑となる。そこで、各算術演算間の入出力関係を調べ、ある多入力算術演算の結果が他の多入力算術演算の加算入力としてのみの一度しか使われない場合には、前者を後者に代入し、一つの多入力算術演算とする。これにより、多入力算術演算結果の転送回数が減少するとともに、計算量を増大させることなく高並列に多入力算術演算を実行できる。例えば、式 (3)～(5) に示す多入力積和演算が与えられた場合、本手法を用いることにより式 (6) が得られる。

$$x = a_1 b_1 + a_2 b_2 \quad (3)$$

$$y = a_3 b_3 + a_4 b_4 + x \quad (4)$$

$$z = a_5 b_5 + a_6 b_6 + y \quad (5)$$

$$z = a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4 + a_5 b_5 + a_6 b_6 \quad (6)$$

この結果、式 (3)～(5) により与えられた多入力積和演算により z を計算するためには 3 回の多入力積和演算を逐次的に実行する必要があるのに対し、与えられた多入力積和演算に本手法を用いることにより式 (6) に示すように z を 1 回の多入力積和演算で求めることができる。これは、式 (3)～(5) の三つの多入力積和演算を並列に実行することに相当する。また、 z を求めるための中間結果 x, y の転送が多入力算術演算の実行に含まれるため、多入力算術演算結果の転送回数が減少する。

順/逆運動学演算と逆動力学演算の重複分を省いた計算量の少ない計算法 [18] に本手法を適用することにより、Table 2 に示すような種々の入力数の多入力積和演算が必要となる。これらの多入力積和演算は計算量が少ないため、演算遅れ時間の減少に有用であるが、除算後置形微分逆運動学演算アルゴリズムとは異なりデータの流が規則的ではない。したがって、順/逆運動学演算や動力学演算の遅れ時間を減少するためには、個々の多入力算術演算の遅れ時間が小さいのみならず、再構成された種々の入力数の多入力算術演算結果の転送を柔軟に行える並列プロセッサアーキテクチャが必要となる。

*: Subsequent division algorithm is used for solution of simultaneous linear equations.

Fig. 10 Subsequent-division algorithm for differential inverse kinematics computations

Table 2 Multi-operand multiply-additions for inverse dynamics computation

Multi-operand multiply-addition		Number of multi-operand multiply-additions
Number of multiplications	Number of additions	
0	2	126
0	6	6
1	0	66
1	1	12
1	2	33
2	1	72
2	2	36
3	2	206
4	5	18
7	6	6

Fig. 11 Reconfigurable parallel processor using multiple-bus

そこで、Fig. 11 に示すように、PE を多重バスで結合することにより、再構成される多入力算術演算の入力数にかかわらずデータ転送を柔軟に行える再構成可能並列プロセッサを提案する。再構成された多入力算術演算器の出力を、共通バスを通じて直接他の多入力算術演算器の入力に接続することにより、多入力算術演算とその演算結果の転送を、入力数にかかわらず複数個の PE を用いて 2 クロックサイクル程度で行うことができる。このため、順/逆運動学演算や逆動力学演算に必要な種々の多入力算術演算を高並列に実行でき、演算遅れ時間を減少できる。これに対し、DSP を用いた並列処理では、ある一つの多入力算術演算を複数個の DSP で並列に実行すると、多入力算術演算中の中間結果の転送に多くのクロックサイクルが必要となる。また、DSP 間通信オーバーヘッドをできるだけ減少するためには、多入力算術演算を各 DSP で逐次的に実行することが望ましい。この場合、種々の多入力算術演算とその演算結果の転送には数クロックサイクルが必要となるため、演算遅れ時間が大きい。

入力数にかかわらず多入力算術演算とその演算結果の転送を 2 クロックサイクル程度で行えるため、実行可能順に並べられた多入力算術演算を、まだ多入力算術演算が割り当てられていない PE と共通バスに割り当てるといった簡単なスケジューリングにより、演算器の稼働効率が高く演算遅れ時間の小さい

並列プログラムを容易に作成できる。6 個の PE を 5 本の共通バスで結合した再構成可能並列プロセッサを用いることにより、6 自由度ロボットマニピュレータの逆動力学演算の遅れ時間は 115 クロックサイクル時間、すなわち 11.5 [μsec] となる。ここで、再構成可能並列プロセッサ内の 12 個の乗算器のうち、並列に利用される乗算器数は平均して 8.1 個である。これに対し、乗算器と加算器の個数が等しくなるように 12 個の DSP を 1 本の共通バスで結合した並列プロセッサを用いた場合では、演算遅れ時間は 45.2 [μsec] となる。多入力算術演算を各 PE で逐次的に実行するため、並列に利用される乗算器数が平均で 2.0 個と少なく、再構成可能並列プロセッサを用いる場合と比較して演算遅れ時間が約 4 倍となる。仮に DSP 間を 5 本の共通バスで結合したとしても、演算遅れ時間は 21.3 [μsec] となる。並列に利用される乗算器数は平均で 4.2 個であり、再構成可能並列プロセッサを用いる場合と比較して演算遅れ時間が約 2.0 倍となる。以上の結果より、PE 間を多重バスで結合した再構成可能並列プロセッサは順/逆運動学演算や逆動力学演算の遅れ時間減少に有用である。

5. む す び

マニピュレータ動的制御において必要となる、種々の入力数の積和演算や最大値/最小値選択などの多入力算術演算の遅れ時間を小さくする、再構成可能並列プロセッサアーキテクチャを提案した。再構成可能並列プロセッサを用いることにより、微分逆運動学演算、順/逆運動学演算や逆動力学演算のほかにも、計算量が多く従来実現が困難であるとされていた種々の制御アルゴリズムを高速に行える。種々の制御アルゴリズムに要求される性能を達成するためには、PE 数やネットワーク構造を適切に選択するとともに、演算器稼働効率が高く演算遅れ時間が小さい並列処理を実現するソフトウェアを作成するための開発環境が重要である。このため、ロボット制御などに必要となる種々の入力数の多入力算術演算を効率よく実行するためのハイレベルシンセシスやスケジューリングアルゴリズムの開発が今後重要な課題となる。

参 考 文 献

- [1] 亀山, 樋口: “ロボットと VLSI コンピュータ”, 日本ロボット学会誌, vol.6, no.4, pp.332-338, 1988.
- [2] B. Kim, M. Kameyama and T. Higuchi: “Parallel VLSI processors for robotics using multiple bus interconnection Networks,” Trans. IEICE, vol.E75-A, no.6, pp.712-719, 1992.
- [3] M. Kameyama, T. Matsumoto, H. Egami and T. Higuchi: “A special-purpose LSI for inverse kinematics computation,” Trans. IEICE, vol.E74, no.11, pp.3829-3837, 1991.
- [4] S. Kittichaikoonkit, M. Kameyama and T. Higuchi: “Design of a matrix multiply-addition VLSI processor for robot inverse dynamics computation,” Trans. IEICE, vol.E74, no.11, pp.3819-3828, 1991.
- [5] M. Kameyama, T. Amada and T. Higuchi: “Highly parallel collision detection VLSI processor for intelligent robots,” IEEE Journal of Solid-State Circuits, vol.27, no.4, pp.500-506, 1992.
- [6] M. Amin-Javaheri and D.E. Orin: “Systolic architectures for the manipulator inertia matrix,” IEEE Trans. Syst. Man. & Cybern., vol.18, no.6, pp.939-951, 1988.
- [7] T.B. Yeung and C.S.G. Lee: “Efficient Parallel Algorithms and VLSI Architectures for Manipulator Jacobian Computation,”

- IEEE Trans. Syst. Man. & Cybern., vol.19, no.5, pp.983-999, 1989.
- [8] 藤岡, 亀山, 樋口: “冗長マニピュレータ制御用座標変換 VLSI プロセッサ”, 電子情報通信学会論文誌, vol.J75-D-I, no.10, pp.909-916, 1992.
- [9] Y. Fujioka, M. Kameyama and T. Higuchi: “240MOPS reconfigurable parallel VLSI processor for robot control,” Proc. of 1992 Int. Conf. Ind. Electron. Contr. Instr. Automation, pp.1385-1390, 1992.
- [10] Y. Fujioka and M. Kameyama: “2400-MFLOPS reconfigurable parallel VLSI processor for robot control,” Proc. of 1993 IEEE Int. Conf. Robotics Automation, pp.149-154, 1993.
- [11] P. Sadayappan, Y.L.C. Ling, K.W. Olson and D.E. Orin: “A restructurable VLSI robotics vector processor architecture for real-time control,” IEEE Trans. Robotics Automation, vol.5, no.5, pp.583-599, 1989.
- [12] R. Nigam and C.S.G. Lee: “A Multiprocessor-Based Controller for the Control of Mechanical Manipulators,” IEEE Trans. Robotics Automation, vol.RA-1, no.4, pp.173-182, 1985.
- [13] C.L. Chen, C.S.G. Lee and E.S.H. Hou: “Efficient Scheduling Algorithms for Robot Inverse Dynamics Computation on a Multiprocessor System,” IEEE Trans. Syst. Man. & Cybern., vol.18, no.5, pp.729-743, 1988.
- [14] K.P. Jankowski and H.V. Brussel: “An approach to discrete inverse dynamics control of flexible-joint robots,” IEEE Trans. Robotics Automation, vol.8, no.5, pp.583-599, 1992.
- [15] M. Koga, K. Kosuge, K. Furuta and K. Nosaki: “Coordinated motion control of robot arms based on the virtual internal model,” IEEE Trans. Robotics Automation, vol.8, no.1, pp.77-85, 1992.
- [16] B.W. Drake and T.C.S. Hsia: “Implementation of a unified robot kinematics and inverse dynamics algorithm on a DSP chip,” IEEE Trans. Ind. Electron., vol.40, no.2, pp.273-281, 1993.
- [17] C.A. Klein and C.H. Huang: “Review of pseudoinverse control for use with kinematically redundant manipulators,” IEEE Trans. Syst. Man. & Cybern., vol.SMC-13, no.3, pp.245-250, 1983.
- [18] 中村, 横小路, 吉川: “ロボットマニピュレータの運動学と動力学の統合化計算”, 計測自動制御学会論文集, vol.23, no.5, pp.71-78, 1987.

藤岡 周 (Yoshichika Fujioka)

1967 年 5 月 19 日生。1990 年東北大学工学部電子工学科卒業。1995 年同大学院博士課程修了。同年八戸工業大学工学部講師, 現在に至る。ロボットエレクトロニクスに関する研究に従事。博士 (情報科学)。(日本ロボット学会正会員)

苫米地 宣裕 (Nobuhiro Tomabechi)

1939 年 11 月 10 日生。1962 年東北大学工学部電子工学科卒業。1974 年八戸工業大学工学部講師, 1985 年同大学教授, 現在に至る。フォールトトレラントシステム, VLSI 向きシステム構成法, デジタル信号処理システムなどに関する研究に従事。工学博士。IEEE, 情報処理学会各会員。

亀山 充隆 (Michitaka Kameyama)

1950 年 5 月 12 日生。1973 年東北大学工学部電子工学科卒業。1978 年同大学院博士課程修了。同年同大学工学部助手, 1981 年同大学助教授, 1991 年同大学教授, 現在に至る。多値情報処理システム, フォールトトレラントシステム, 知能集積システム, ロボットエレクトロニクスなどの研究に従事。

多値論理国際シンポジウム優秀論文賞受賞計 5 回 (IEEE Computer Society 主催)。昭和 61 年度計測自動制御学会技術賞受賞。平成元年度電子情報通信学会論文賞受賞。平成 2 年度日本ロボット学会技術賞受賞。工学博士。(日本ロボット学会正会員)