

## 演算器レベル・パケット転送方式に基づく 高並列 VLSI プロセッサの構成

藤岡 与周<sup>†</sup> 苫米地 宣裕<sup>‡</sup> 亀山 充隆<sup>†</sup>

<sup>†</sup> 八戸工業大学 工学部 〒031-8501 八戸市大字妙字大開 88-1

<sup>‡</sup> 東北大学 大学院 情報科学研究科 〒980-8579 仙台市青葉区荒巻字青葉 6-6-05

E-mail: <sup>†</sup> {fujioka, tomabech}@hi-tech.ac.jp, <sup>‡</sup> kameyama@ecei.tohoku.ac.jp

あらまし 不規則かつ高並列な処理の演算遅れ時間を減少するためには、細粒度高並列処理が有用であるが、通常の VLIW 制御方式ではルータの個数とともにデータ転送などの制御メモリ容量もまた膨大になる。そこで本稿では、あらかじめ処理アルゴリズムが与えられていると仮定して、スケジューリングやアロケーションをパケット衝突が起らずかつ演算遅れ時間を減少するという制約下でオフラインで実施する、セミオートノマスパケットルーティングの概念を提案している。これにより、ルータ構造を大幅に簡略化できるとともに、パケット転送による自律的な制御タイミング生成により制御メモリを大幅に減少可能な高並列 VLSI プロセッサアーキテクチャを提案している。

キーワード ネットワークオンチップ、並列 VLSI プロセッサ、セミオートノマスパケットルーティング、制御メモリ容量の減少

## Design of a Highly Parallel VLSI Processor Based on Functional-Unit-Level Packet Data Transfer Scheme

Yoshichika FUJIOKA<sup>†</sup> Nobuhiro TOMABECHI<sup>‡</sup> and Michitaka KAMEYAMA<sup>†</sup>

<sup>†</sup> Faculty of Engineering, Hachinohe Institute of Technology 88-1 Myo Ohbiraki, Hachinohe 031-8501 Japan

<sup>‡</sup> Graduate School of Information Sciences, Tohoku University 6-6-05, Aoba, Aramaki, Aoba, Sendai 980-8579 Japan

E-mail: <sup>†</sup> {fujioka, tomabech}@hi-tech.ac.jp, <sup>‡</sup> kameyama@ecei.tohoku.ac.jp

**Abstract** Until now, network on chip technology based on coarse grain packet data transfer was proposed. In this paper, fine grain packet data transfer scheme is introduced to make intra-chip data transfer flexible and programmable in micronetwork.

A protocol based on hybrid utilization of autonomous packet data transfer and offline scheduling/allocation is effectively employed for making a router as simple as possible, so that packet collision in the micronetwork does not occur. Because the timing control of packet-receive is automatically done in the router, complexity of VLIW control can be greatly reduced. A special control module to control the packet-send timing is proposed to realize effective packet data transfer.

**Keyword** Network-on-Chip, Parallel VLSI Processor, Semi-Autonomous Packet Routing, Reduction of Control Complexity

### 1. まえがき

これまで、粗粒度パケット転送に基づくネットワークオンチップが提案されている。[1]-[3]。本稿では、マイクロネットワークでのチップ内データ転送をフレキシブルかつプログラマブルとするため、細粒度パケット転送方式を提案している[4]-[5]。

ルータ構造をできるだけ簡単にするため、自律的パケットデータ転送とオフラインのスケジューリング・アロケーションの組合せに基づくプロトコルを採用しており、マイクロネットワーク内でのパケット衝突が起らない。また、ルータ内でパケット受信制御が自

動的に行われるため、VLIW 制御メモリ容量を大幅に減少できる。さらに、効果的なパケットデータ転送を実現するため、パケット送信タイミング制御のための新しい制御モジュールを提案している。

### 2. パケットデータ転送に基づく並列 VLSI プロセッサアーキテクチャ

#### 2.1. パケットデータ転送に基づくマイクロネットワーク構造

図 1 に、複数の PE と、それらを接続するマイクロ

ネットワークを備えた細粒度 MIMD (Multi-Instruction Multi-Data)型並列プロセッサの提案構造を示す。ここで、処理アルゴリズムが与えられ、それがコントロールデータフローグラフ (CDFG) で表現されており、スケジューリング・アロケーションが予め決定されていると仮定する。CDFG におけるそれぞれのノードは PE に割当てられ、エッジに対応する PE 間データ転送はマイクロネットワークを経由してなされる。

マイクロネットワークには、図 2 に示すようにパケット転送に用いる 2 本のデータ転送ラインがあり、一つは左から右への方向のパケット転送に用いられ、もう一つは右から左への方向のパケット転送に用いられる。2 つの PE 間のパケット転送は PE に直接接続されたルータを経由してなされる。

プロセッサの基本操作はレジスタ間データ転送に帰着されるため、もしパケットがマイクロネットワークに適切に送信されると、データ受信のための自動的なタイミング生成がなされる。このことはデータ受信のタイミング制御をマイクロプログラム中で省略できることを意味する。したがって、VLIW (Very Long Instruction Word) 制御メモリに要求されるメモリ容量を大幅に減少できる。

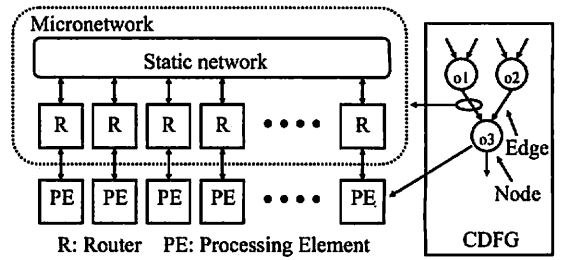


図 1 並列プロセッサアーキテクチャ

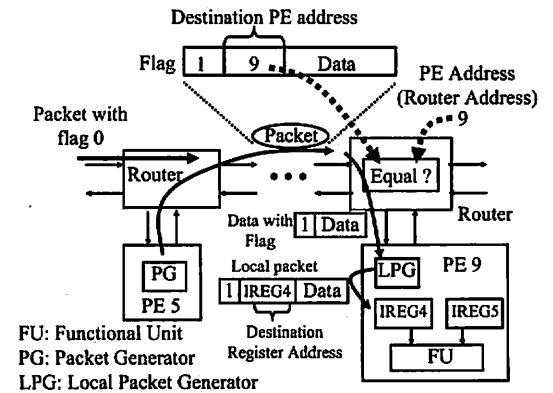


図 2 パケット転送の例

## 2.2. セミオートノマスパケットルーティング

細粒度並列処理における頻繁なパケット転送のためには多くのルータが要求されるため、ルータ内の制御機構はできるだけ単純でなければならない。そこで、オフラインのスケジューリング・アロケーションがセミオートノマスパケット転送に用いられる。これに必要なとなる最適問題はパケット転送が起こらないという制約下で全体の処理時間を最小化することである。もしこの制約が満たされるなら、ルータ内のバッファキューメモリが不要になる。

## 2.3. パケットフォーマット

セミオートノマスパケットルーティングに基づくことにより、1 ビットのフラグ、宛先アドレスと 1 個のデータからなり、優先順位など他のヘッダ情報が不要である、単純なパケットフォーマットを定義できる。フラグはパケットが有効なものか否かを表し、パケット送信制御を自律的にするために効果的に採用されている。

## 2.4. 階層的パケット転送

階層的に、PE 内でもパケット転送が利用される。PE 間および PE 内の両方の階層に関係するパケット転送の例を図 2 に示す。

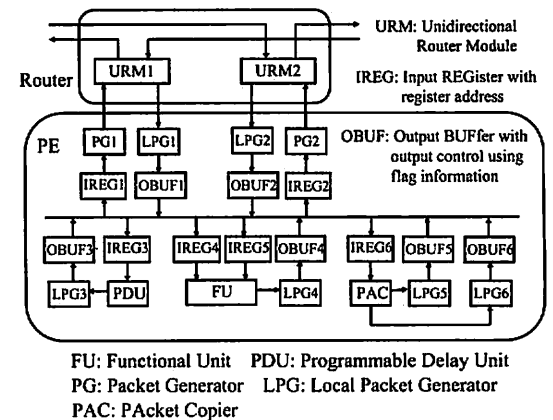


図 3 パケット転送のためのルータと PE の構成

[手続き 1] 宛先 PE アドレス 9 とデータを含むパケットがパケットジェネレータ (PG) で生成され、それが PE5 から送出される。PE5 からのフラグが有効であり、かつ隣のルータからのフラグが無効であるため、PE5 からのパケットが自律的に PE9 へ送られる。

[手続き 2] パケットはルータ間をパイプライン方式で転送される。宛先アドレスと各ルータアドレスの比較がなされ、それらが等しくない場合パケットは隣のルータへ転送される。

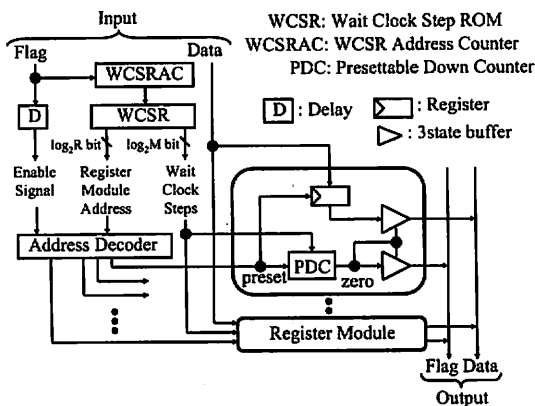


図 4 プログラマブルディレイユニット

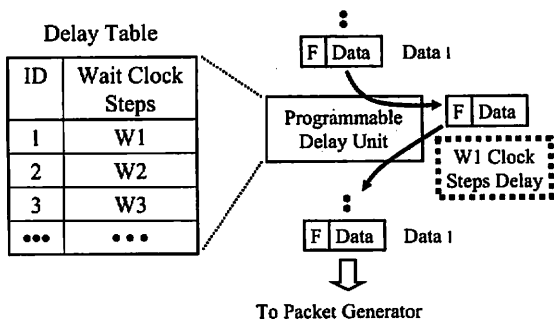


図 5 PDU の動作

[手続き 3] 宛先 PE アドレスとルータアドレス 9 が等しいため、パケットは PE9 に転送され、フラグ情報が PE 内データ転送でのパケット送信のきっかけとして用いられる。

[手続き 4] PE9 では、新しいローカルパケットがローカルパケットジェネレータ(LPG)で生成され、IREG4 に送られる。

### 3. プロセッシングエレメントの構成

図 3 は PE の構成を示しており、レジスタファイルがプログラマブルディレイユニット (PDU) 内に中間結果の一時記憶のために備えられている。最も重要な制御機能は PDU からパケットをセミオートノマスパケット転送法に基づく指定されたタイミングで送出することである。PDU では、遅延制御がプログラムされた遅延情報に従ってなされる。

#### 3.1. プログラマブルディレイユニット

図 4 は複数のレジスタモジュールを備えた PDU の構造を示している。それぞれのレジスタモジュール内には、データの保存と遅延制御それぞれのために、ひと

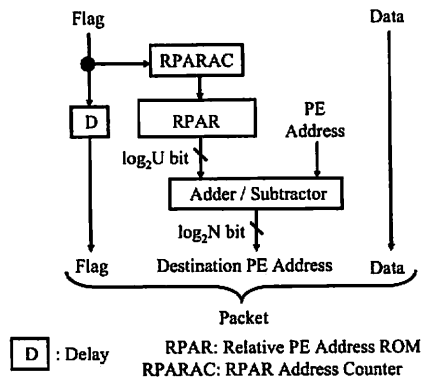


図 6 パケットジェネレータ

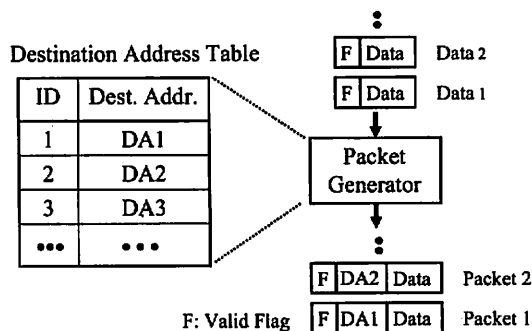


図 7 PG の動作

つのレジスタとプリセットダウンカウンタ (PDC) が備えられている。もし有効なフラグがデータとともに PDU に到着すれば、以下に示す遅延制御がなされる。

[手続き 1] レジスタモジュールのアドレスと待ちクロックステップ数がウェイトクロックステップ ROM (WCSR) により生成される。

[手続き 2] 入力データがレジスタモジュール内のデータレジスタに保存され、待ちクロックステップ数が PDC にセットされる。

[手続き 3] 待ちクロックステップ数のカウントダウンが開始される。もしカウントが零になると、データは PDU から LPG に有効フラグとともに送られる。

以上の手続きにより、図 5 に示すように自律的に遅延操作とデータ送出が行われる。

#### 3.2. パケットジェネレータ

図 6 は有効フラグを伴うデータが入力後すぐにパケットを生成する PG の構成を示している。また、図 7 にその動作概念を示す。制御メモリ容量を減少するために、あらかじめプログラムされた相対 PE アドレス

が導入されており、それが相対 PE アドレス ROM (RPAR)に保存されている。この相対 PE アドレスはもしパケットデータ転送方向が右方向であれば PE アドレスに加算される。一方、もしそれが左方向であれば、PE アドレスから減算される。

#### 4. プログラミングの比較

スケジューリングおよびアロケーションが行われた CDFG のプログラミングを比較する。提案するセミオートノマスパケットルーティングアーキテクチャでは、パケットの宛先アドレスを指定することによりパケット受信タイミングなどを自律的に生成可能となり、プログラミングに必要な制御メモリ容量を大幅に減少できる特徴を有する。

一例として、図 8 に示す CDFG について、(1)から(4)の各エッジは、図 3 の PE 上でそれぞれ以下のようなパケットの流れに対応する。

- (1) PDU → LPG3 → OBUF3 → IREG4 → FU
- (2) PDU → LPG3 → OBUF3 → IREG5 → FU
- (3) FU → LPG4 → OBUF4 → IREG3 → PDU
- (4) PDU → LPG3 → OBUF3 → IREG2 → PG2 → URM2 → PE5

ここで、IREG<sub>x</sub> はアドレス x をもつ入力レジスタであり、LPG で生成された宛先アドレス x のローカルパケットを選択的かつ自律的に受信する機能を有する。また、OBUF は PE 内共通バスへの出力バッファであり、LPG で生成されたパケットの有効フラグにより自律的にバスへのパケット送出を制御する機能を有する。さらに、FU は入力データのフラグの状態から、入力データがそろってから演算結果を出力するまでの遅延タイミング (図 8 の例では 3 クロックステップ) を自律的に生成できる機能を備えている。

以上より、図 8 の例のためにプログラミングが必要となるのは、PDU、LPG3、LPG4、PG2 である。それぞれのプログラミングは以下の通りである。

PDU には、初期値として、レジスタモジュール 1 のデータレジスタと PDC にそれぞれデータ a と待ちクロックステップ数 0 が記憶されている。同様に、レジスタモジュール 2 のデータレジスタと PDC にはデータ b と待ちクロックステップ数 1 が記憶されている。さらに、PDU 内の制御メモリ WCSR には、図 8 の(3)の段階で PDU に入力されるデータ c を記憶するためのレジスタモジュールアドレスと待ちクロックステップ数 7 がプログラムされる。

LPG は、図 6 の PG からアドレス加減算器を省いた構造となっている。LPG3 内のローカルパケット宛先アドレスを記憶する制御メモリには、図 8 の例の場合データ a,b,c それぞれの行き先アドレスである 4,5,2 が

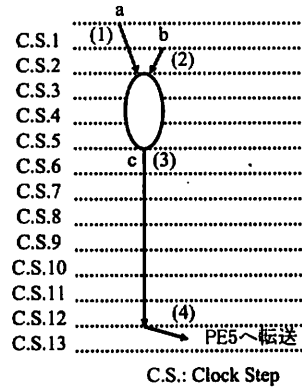


図 8 CDFG の一部の例

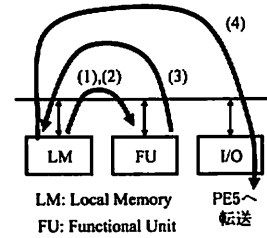


図 9 従来のアーキテクチャによる実行

表 1 従来アーキテクチャでの VLIW プログラム

VLIW メモリ 番地.	LM アド レス	RD	WR	FU 入 力 1	FU 入 力 2	FU 出 力	I/O	
1	a	RD		IN				(1)
2	b	RD			IN			(2)
3								
4								
5								
6	c		WR			OUT		(3)
7								
8								
9								
10								
11								
12	c	RD						
13							OUT	(4)

順にプログラムされる。これによりデータ a,b,c は順に IREG4, IREG5, IREG2 宛のローカルパケットにそれぞれ変換される。同様に、LPG4 には FU から出力されるデータ c の宛先アドレスである 3 がプログラムされる。さらに、PG2 には、宛先 PE アドレスである 5 とこの PE アドレスの差を RPAR にプログラムするだけでよい。

これに対し、従来のアーキテクチャでは、図 9 に示

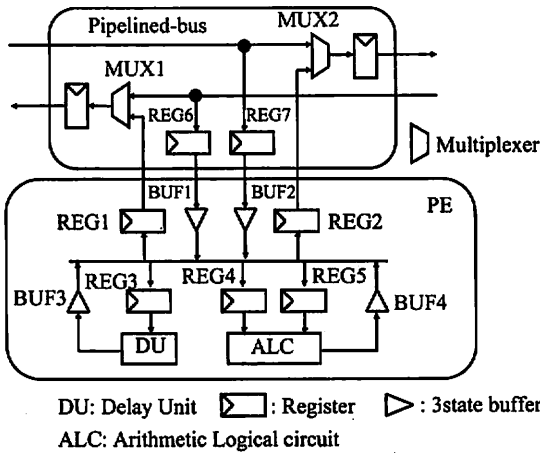


図 10 従来のパイプラインバスアーキテクチャ

すようなデータの移動やメモリアクセスなどが必要となる。このための VLIW プログラムは、表 1 に示すように、特に待ちクロックステップなどのために無駄に制御メモリ容量が必要となる。

### 5. 評価

提案する VLSI プロセッサと従来の VLIW 制御に基づく VLSI プロセッサとの制御メモリ容量を比較する。提案する VLSI プロセッサの制御メモリ容量  $M_p$  は次式で与えられる。

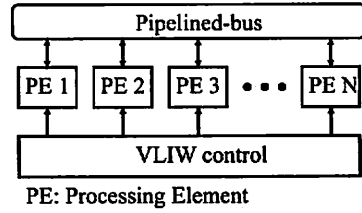
$$M_p = P \log_2 U + 3X + D(\log_2 R + \log_2 M) \quad (1)$$

最初の項は PE 間パケット転送に用いられるパケットの総数  $P$  と相対 PE アドレスのビット長  $\log_2 U$  との積であり、ここで  $U$  はアロケーションで決定される PE 間パケット転送の最大距離である。第 2 項は PE 内パケット転送に用いられるパケットの総数  $X$  と宛先レジスタアドレスのビット長 3 との積である。第 3 項は PDU に保存されるデータの総数  $D$  と、レジスタモジュールアドレスのビット長  $\log_2 R$  とスケジューリングで決定される最大待ちクロックステップ数ビット長  $\log_2 M$  の和との積である。

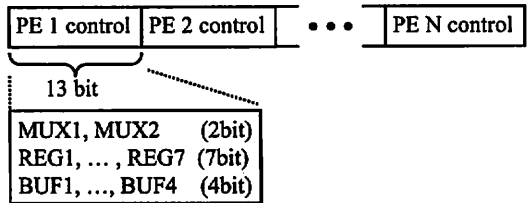
次に、従来の VLIW 制御に基づくプロセッサを評価してみる。図 10 にパイプラインバスアーキテクチャに基づく PE 構造を示す。VLIW 制御法は 2 種類がある。ひとつは図 11 に示すようにそれぞれのクロックステップ毎にオンオフ制御信号が与えられる方法である。この場合、制御メモリ容量  $M_v$  は次式で与えられる。

$$M_v = 13SN + D(\log_2 R + \log_2 M) \quad (2)$$

ここで、 $S$  はクロックステップ総数である。制御メモリ容量  $M_v$  は制御信号がそれほどしばしば変化しな



(a) 構成



(b) VLIW 制御フィールド

図 11 クロックステップベース VLIW 方式パイプラインバスアーキテクチャ

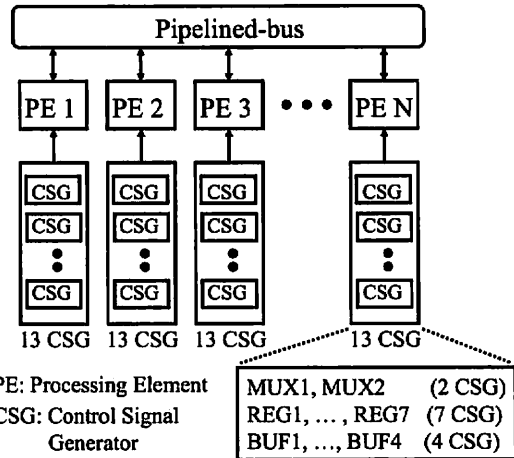


図 12 制御信号ベース VLIW 方式パイプラインバスアーキテクチャ

い場合、メモリ容量が無駄になる。

別の方法は、制御信号がオンになるようなクロックステップ間隔の情報を使うことである。制御信号がクロックステップ  $C1$  でオンになり、再びそれがオンになるもっとも近いクロックステップが  $C2$  であるとする。データ転送のための  $C1$  と  $C2$  のクロックステップ間隔がプログラムされる。制御信号がそれほどしばしば変化しない場合、この方法は制御メモリ容量の減少に有用である。PE 内の制御信号発生器 (CSG) が図 12 に示すようにプログラムされたクロックステップ間隔に応じたそれぞれの制御信号生成に用いられる。この場合、制御メモリ容量  $M_b$  は次式で与えられる。

表2 パケット転送方式とクロックステップベース

VLIW方式の制御メモリ容量					
N	Q	S [c.s.]	Mp [bit]	Mv [bit]	Mv/Mp
64	20	840	34560	712960	20.6
64	100	4200	204800	3596800	17.6
256	20	2760	158720	9251840	58.3
256	100	13800	896000	46361600	51.7
1024	20	10440	716800	139284480	194.3
1024	100	52200	3993600	696832000	174.5

表3 パケット転送方式と制御信号ベース

VLIW方式の制御メモリ容量					
N	Q	S [c.s.]	Mp [bit]	Mb [bit]	Mb/Mp
64	20	840	34560	116480	3.4
64	100	4200	204800	806400	3.9
256	20	2760	158720	578560	3.6
256	100	13800	896000	3507200	3.9
1024	20	10440	716800	2764800	3.9
1024	100	52200	3993600	16281600	4.1

$$Mb = 2P \log_2 M + 2X \log_2 M +$$

$$D(\log_2 R + \log_2 M) \quad (3)$$

スケジューリングおよびアロケーションされたCDFGの制御メモリ容量の比較を検討する。処理時間に対応するクロックステップ数Sは次式で与えられる。

$$S = Q(W + V) \quad (4)$$

ここで、W、V、Qはそれぞれ次のとおりである。

W: 各ノードのクロックステップ数

V: 並列PE間データ転送の平均クロックステップ数

Q: 処理の繰り返し数。ここで、ひとつの処理はひとつのノードの処理とPE間データ転送にW+Vクロックステップを要する。

図13、表2と表3はP=E, U=N/2, X=4E, D=E, R=Q/2, M=S/2, E=NQ, W=10, V=Uである場合の制御メモリ容量Mp, Mv, Mbの比較を示している。実際の処理応用では、提案アーキテクチャにより制御メモリ容量の大幅な減少が期待できる。

## 6. むすび

提案するVLSIアーキテクチャは、不規則なデータ転送がしばしば発生するような場合に、制御メモリ容量の減少に非常に有用である。従って、提案するマイクロネットワークアーキテクチャ内のより多くのPEを同一面積のチップ内に備えることができ、並列処理

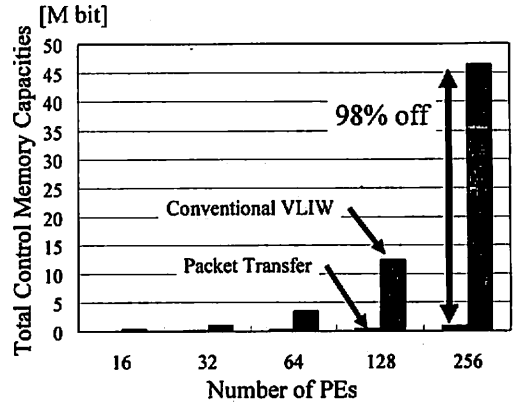


図13 パケット転送方式とクロックステップベース VLIW方式の制御メモリ容量の比較

能力を大幅に改善できる。

## 参考文献

- [1] P. P. Pande, C. Grecu, A. Ivanov, R. Saleh and G. D. Micheli, "Design, synthesis and test of networks on chips," IEEE Design and Test of Computers, Vol. 22, No. 5, pp. 404-413, (2005).
- [2] K. Goossens, J. Dielissen and A. Radulescu, "AETHERAL Network on Chip: Concepts, Architectures, and Implementations," IEEE Design and Test of Computers, Vol. 22, No. 5, pp. 414-421, (2005).
- [3] S. J. Lee, K. Lee and H. J. Yoo, "Analysis and Implementation of Practical, Cost-Effective Network on Chips," IEEE Design and Test of Computers, Vol. 22, No. 5, pp. 422-433, (2005).
- [4] Y. Honma, M. Kameyama, Y. Fujioka and N. Tomabechi "VLSI Architecture Based on Packet Data Transfer Scheme and Its Application," Proc. of 2005 IEEE Int. Symp. on Circuits and Systems, pp.1786-1789, (2005).
- [5] Y. Fujioka, N. Tomabechi and M. Kameyama: "Functional-Unit-Level Packet Data Transfer Scheme for A Highly Parallel VLSI Processor," Proc. of Int. Conf. on Computers and Devices for Communication, CD-ROM, pp.9-13. (2006).