計算機アーキテクチャ 133－9
（1999. 5. 21）

光インターコネクションを有するシステムにおける並列処理アルゴリズム

成瀬 誠, 石川正俊

東京大学大学院工学系研究科計数工学専攻

〒113-8656 文京区本郷 7-3-1

e-mail:naruse@k2.t.u-tokyo.ac.jp

WWW: http://www.k2.t.u-tokyo.ac.jp/~sfoc/

TEL: 03-5841-6902 FAX: 03-5841-8604

チップ間の配線遅延など計算機システムにおける配線の問題を解決する技術として, 集積化された光デバイスを用いる自由空間光インターコネクションの利用が検討されているが, 物理層の性能の向上を生かしたアルゴリズムの設計が求められる. 本稿は, 光インターコネクションによりもたらされる物理層の特徴を踏まえ, ネットワーク理論に基づいたシスィティマティックなアルゴリズム導出の枠組と手法を示すと共に, 光インターコネクションの導入による性能の向上を定量的に示す.

Parallel Processing Algorithms for Optically Interconnected Systems

Makoto Naruse and Masatoshi Ishikawa

Department of Mathematical Engineering and Information Physics,
Graduate School of Engineering, University of Tokyo

The introduction of tremendous bandwidth supplied by free-space optical interconnection to computing systems is expected to solve the lack of communication capability in computing systems. The algorithm design should be re-considered so that the merit of free-space optical interconnection is fully applied. We show how to implement given algorithms onto optoelectronic systems. The contribution of the improvement at the physical layer to the processing time is also evaluated.

1 背景

LSIの集積度及び動作速度の飛躍的な向上により，チップ間の配線遅延やそれに伴う通信のバンド幅の不足が計算機システムの性能改善を妨げる要因として顕在化してきた．Fig.1に示されるように，半導体集積回路と光入力及び光出力デバイスを組み合わせた光電子VLSI [1]と，信号を並列に伝送する自由空間 光インターコネクション[2]は，通信のボトルネックを解消する技術として，多くの研究機関で研究が行われている．

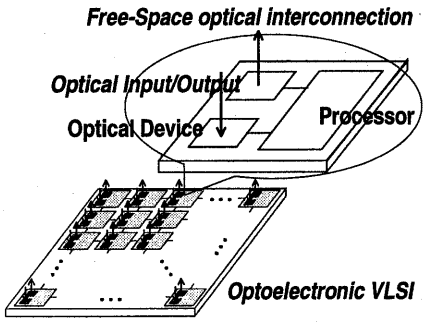

Fig. 1 光電子VLSIの概念図

2 光インターコネクションとアルゴリズム設計

光インターコネクションによる物理層の能力の改善を生かすためには，アルゴリズムの設計において，光インターコネクションの利用により顕在化する次の3つの評価基準あるいは特性が適切に考慮される必要がある．

2.1 通信のコストと計算のコストの適切な評価

光インターコネクションの導入により，チップ間の通信における遅延時間 t_c が低減し，チップ上でのインストラクション実行時間 t_p に近づいてくる．これにより，ある演算の実行時間は，演算をチップで実行するのに要する時間だけでなく，演算の実行に必要な変数をチップに転送する時間を含めて評価しなければならない．

2.2 「密度」を基本とする評価

光電子VLSIにおいては，光学素子の分解能に相当する配線密度がLSI 上に確保される．パッケージに実装されたピンを利用する電気配線との本質的違いを表現する上でも，LSIの単位面積当たりでの通信能力の評価が重要となる．

2.3 自由空間利用による配線の自由度の評価

自由空間に信号の伝播経路が構築できるため，例えば，複数個の LSI 間に再構成可能な相互接続網を構築できる[2]．このような自由空間に信号の伝搬経路が実装されることの特徴を適切に評価し利用する必要がある．

3 アルゴリズム設計の枠組みと手法

3.1 アルゴリズムの定量

2.1節で示された通信と計算の実行時間の評価指針に基づけば，$y = f(x_1, x_2)$ と書かれる演算の実行時間は，次の二つの要素により構成されると考えられる．

1. 変数を演算 f が実行されるプロセッサまで転送する(これを「**要素的通信**」と定義する)ためのコスト．
2. 演算 f をプロセッサで実行する(これを「**要素的演算**」と定義する)ためのコスト．

ここで，与えられたアプリケーション全体の実行時間 PT を，N_c を要素的通信の総数，N_p を要素的演算の総数として，

$$PT = N_c \cdot t_c + N_p \cdot t_p \qquad (1)$$

と評価することとする．ここで，システムの基本的な能力を表現し，かつ2.2節で述べた性質を満足する量として，「LSIの単位面積当りに実装されるトランジスタ数とトランジスタの動作周波数の積」(これを「**計算能力密度**」と定義する)，及び「LSIの単位面積当りに実装される入出力のチャンネル数とチャンネルの動作周波数の積」(「**通信能力密度**」と定義)を考える．さらに，2.3節で述べた性質を包含するために，上記の計算能力密度と通信能力密度で特徴付けられる複数個の LSI が結合することにより構成されたシステムを仮定する．

このようなシステム上で，(1)式で与えられる実行時間を有するアプリケーションを実行するときに，いくつかの要素的演算又は要素的通信が同時に実行で

きれば, 実効的には, $ET = EN_c \cdot t_c + EN_p \cdot t_p$(「実効的計算時間」) で与えられる時間内で計算を実行できると考えられる. ここで EN_c, EN_p はそれぞれ実効的な通信及び計算の個数である. 実際には, 要素的通信は通信能力密度 CD を有する LSI 間に実装され, 要素的計算は計算能力密度 PD を有する LSI で実行されることから, $\lceil * \rceil$ を $*$ 以上の最小の整数を表す記法として, 実効的計算時間は,

$$\sum_{\alpha} c_{ij} \left\lceil \frac{I_j}{CD_j} \right\rceil \cdot t_c + \sum_{\alpha} \left\lceil \frac{A_j}{PD_j} \right\rceil \cdot t_p \qquad (2)$$

となる. ここで I_j 及び A_j は, 指標が j である LSI の単位面積に割り当てられた通信量及び演算量であり, $\lceil\ \rceil$ を含む部分は, 通信能力密度あるいは計算能力密度よりも過剰の通信量あるいは計算量が割り当てられた場合に要求される繰り返し回数である. α は, 演算の開始時点から終了までの手順で決まる一連の過程を示す指標である. $c_{ij}t_c$ は, 指標が i, j で示される LSI 間の通信時間示す.

実効的計算時間を最小とするには, Fig.2 に示される概念図のように, システム内部の一部に過剰に通信や計算が集積することを避け, 実効的に同時実行される通信量と計算量が増加するように, アルゴリズムの開始時点 (Fig.2 中の "start") から演算終了 (同図 "Result Obtained") までの通信量と計算量をシステム内部へ割り当てればよいと考えることができる.

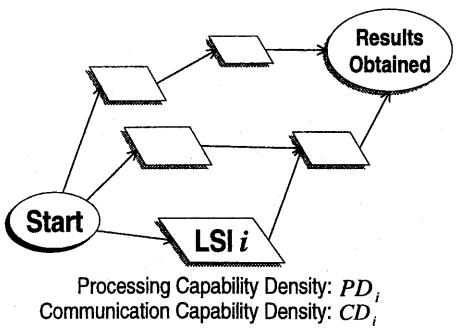

Fig. 2 アルゴリズム設計の基本的考え方の概念図

3.2 最適なアルゴリズムの設計

(2) 式を最小とするアルゴリズム設計の基本的方法を説明する. ある LSI の単位面積に割り当てられた計算量が, その LSI の単位面積及び単位時間内に処理できる計算量 (すなわち計算能力密度) より大きい場合には, 割り当てられた計算を時分割して実行する必要が生じる. 同様に, ある LSI 間の単位面積及び単位時間内に割り当てられた通信量が, その通信能力密度よりも大きい場合には, 通信を時分割して行う必要がある. このように, システムの内部で過剰に通信あるいは計算が割り当てられた箇所が存在するとき, システムの他の部分に計算あるいは通信を分散させることによって, 実効的計算時間を短縮できる可能性がある.

より具体的には, 以下の手順で最小実行時間でのアルゴリズム設計が可能になる. システムの能力とシステムに実際に割り当てられている計算量及び通信量から, システム各部の能力を超過している, 又はシステム各部が許容可能な計算量及び通信量を表現する行列 (ここでは, これを「コスト行列」と呼ぶ) を作成する. ここで, 許容能力を超過した計算あるいは通信を割り当てられた要素には, 負の値を取る要素を適当に割り当てる.

例えば, Fig.3 左上に示すように, 3 個の LSI(start, LSI1,LSI2 とする) が同図左上のように結合した状況でのコスト行列を考える. 簡単のために, $t_c = t_p = 1$ であるとする. それぞれの LSI の通信能力密度及び計算能力密度は 1 であるとし, 今, 同時実行可能なすべての計算が LSI2 に割り当てられ, その計算量が 7 であり, start に存在する変数 (通信量 1) を要請するとする. このとき, LSI2 には, 計算能力密度である 1 を上回る処理が割り当てられていると言える. 同図中 (*) 部に示される "-6"は, 能力を超過した演算量を示している.

上のようにして定義したコスト行列から, アルゴリズムの実行開始時点から終了までの一連の計算時間及び通信時間を枝の重みとして有するネットワークを考えることができる. このネットワークにおいて, アルゴリズムの実行開始時点から終了までの最短経路を求めることにより, 最短実行時間のアルゴリズムが求まる. 最短経路の探索においては, コスト行列が表現するネットワークにおける「負の閉路」とい

う概念を用いる．この量は，アルゴリズムの設計途中においては，「より短い実行時間のアルゴリズムの存在」を示唆する量であると捉えることができる．すなわち，許容能力を超過した量の計算あるいは通信を割り当てられている箇所から，まだ許容能力のある箇所へ，計算あるいは通信を分散させることができることを示している．最短時間の実行アルゴリズムが得られた時には負の閉路が存在しないと言える．

また，ネットワークフロー理論 [3] によれば，「負の閉路」が存在しないときには，コスト行列によって表現されるネットワークにおいて，アルゴリズムの開始時点から終了までの各時点に至るまでの最短経路を与えることができる．従って，このなかにはアルゴリズムの開始から終了までまでの最短経路も含まれる．従って，負の閉路が存在しないことが，(2) 式が最小となる必要十分条件であると言える．

例えば，Fig.3 においては，LSI2 に過剰に割り当てられた計算量を，LSI1 へ分散できる可能性がある．start から LSI1 への通信には，同図中 (**) 部に示される “2” の時間がかかり，start から LSI2 への伝送よりもコストがかかる．しかしながら，同図右下に示されるように，LSI2 から start を経由して LSI1 へ至る経路において，負の閉路が存在している (この場合は，その値は “-4”)．すなわち，LSI2 に割り当てられた処理を LSI1 へ移動させることが，その移動のコストを考慮に含めたとしても全体の処理時間を短縮できることを示している．従って，この閉路に沿って，LSI2 で過剰になっている計算量を分散させればよい．

この操作を繰り返すことにより，最終的には，Fig.4 に示されるように，LSI1 及び LSI2 にそれぞれ演算量 3 及び 4 が割り当てられた状態となる．この状態では，「負の閉路」が存在していない．すなわち，最短時間でアルゴリズムが実装されていることを示す．

上記手法の全体の概要を Table 1 にまとめた．

4 「通信」と「計算」のバランスの効果

光電子 VLSI，光インターコネクション及び既存の代表的な MPU が与える物理層の具体的な能力を参考にして，物理層の能力の違いがアプリケーションの実行時間に与える影響を評価する．ここでは議論を簡単にするために，アプリケーションの性質として，「全ての要素的演算が互いに独立に実行可能であり，

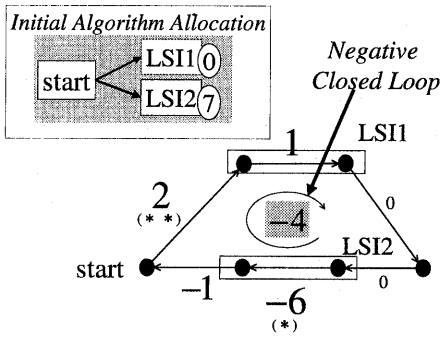

Fig. 3 通信と演算のコスト (コスト行列) により規定されるネットワークと「負の閉路」の概念図 (初期状態)

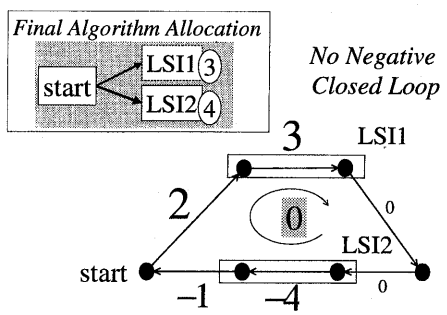

Fig. 4 通信と計算の最終的割り当て

各々の要素的演算は個別の要素的通信を要請する」と仮定した．このような性質を有する演算には，例えば画像のフィルタリングなどの行列演算が挙げられる．仮定した特性を有する演算で，要素的計算の総数が $N_p = 100,000$ である場合について，(3) 式に従って実効的計算時間を求めた．Table 2 の第 1 列は，試作された代表的な MPU 及び光電子 VLSI の事例であり，第 2 列は「通信能力密度 - 計算能力密度比」，第 3 列が実効的計算時間である．「通信能力密度 - 計算能力密度比」とは，3.1 節で定義した，通信能力密度と計算能力密度の比であり，各々のデバイスのトランジスタ数，チップの動作周波数，ダイサ

Table 1 最小実行時間での実装アルゴリズム

ステップ 1(初期化) 適当な実行可能解 (システムに実装可能な通信量及び計算量の割り当て) を与える.

ステップ 2(負の閉路の検出) システムの各部が有する能力と, 割り当てられた通信量及び計算量から, 「コスト行列」を導き, 「負の閉路」を探索する. 負の閉路が存在しなければ, (2) 式は最小化されたと判断できるため, 終了.

ステップ 3(アルゴリズム変更) ステップ 2 で求めた「閉路」に沿って, アルゴリズムを変更し, ステップ 2 に戻る.

イズ, オフチップ通信の周波数, パッケージピンあるいは光入出力デバイスの数を元に計算した. 光電子 VLSI では, LSI の単位面積上の高い配線密度により, 通信能力密度 - 計算能力密度比が, MPU の場合よりも, 10^2 から 10^3 倍程度大きくなっていることがわかる. ここで仮定したアプリケーションの実効的計算時間は, 光電子 VLSI を用いた場合, MPU を用いたときの 10^{-1} から 10^{-3} 倍程度となっている.

Fig.5 は, 実効的計算時間を, 通信能力密度 - 計算能力密度比の関数として示した図である. 同図内右下に光電子 VLSI により与えられるデータが表われ, 同図内左上に, MPU により与えられるデータが表れている. このことは, 光電子 VLSI がもたらす高い通信能力密度 - 計算能力密度比, すなわち通信と計算のバランスが, ここで仮定しているような高い並列度を持ち, 多くの要素的通信を要求するアプリケーションの実効的計算時間の短縮に, 大きく貢献することを示していると言える.

5 まとめ

自由空間光インターコネクションと光電子 VLSI の導入によってもたらされる物理層の特徴を生かしたアルゴリズムの設計の基本的考え方と具体的手法を示した. ここで示した手法によれば, 与えられた

Table 2 MPU 及び光電子 VLSI の事例と, 通信能力密度 - 計算能力密度比及び実効的計算時間

	通信能力密度 - 計算能力密度比	実効的計算時間
MPU		
Intel Pentium II, 66-MHz bus (Deschutes)	6.40×10^{-6}	2.72×10^{1}
Intel Pentium II(Klamath)	7.10×10^{-6}	2.49×10^{1}
Intel Pentium II, Xeon Processor	7.17×10^{-6}	1.20×10^{1}
Intel Pentium II, 100-MHz bus (Deschutes)	8.07×10^{-6}	1.21×10^{1}
IDT WinChip C6	1.37×10^{-5}	1.86×10^{1}
Intel Pentium with MMX	1.86×10^{-5}	2.27×10^{1}
Intel Pentium Pro	2.11×10^{-5}	3.24×10^{1}
Cyrix 6x86MX	1.82×10^{-5}	2.32×10^{1}
Motorola PowerPC750	1.89×10^{-5}	4.34×10^{0}
Motorola PowerPC740	1.51×10^{-5}	6.24×10^{0}
Optoelectronic VLSI		
UNC WARRP	8.57×10^{-3}	2.00×10^{-1}
UNC & Lucent Page Buffer [4]	5.04×10^{-3}	6.25×10^{-3}
Univ. Tokyo SPE [5]	4.53×10^{-3}	8.90×10^{-1}
Lucent & UNC MPU [6]	1.75×10^{-2}	5.47×10^{-3}
Lucent & UNC AMOEDA	2.56×10^{-2}	1.41×10^{-2}
USC OMNI	4.00×10^{-2}	6.00×10^{-3}

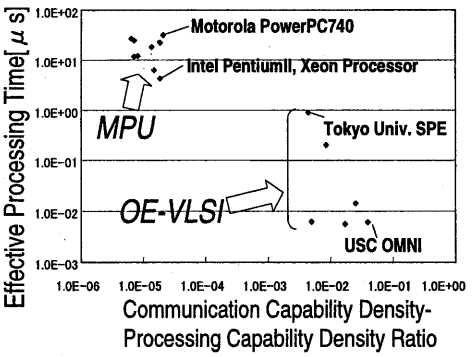

Fig. 5 「通信能力密度 - 計算能力密度比」の関数としての「実効的計算時間」

物理層の特徴を踏まえたシステムのモデルを用いることにより, システィマティックに最短の実行時間を与えるアルゴリズムを導くことができる. また, 光インターコネクションの導入がアプリケーションの実行時間に与える影響を, 既存の MPU と比較評価した. 自由空間光インターコネクションを有するシステムの具体的実装も進行しており [7], 今後は, 実在するシステム上での実験的検証などが必要である.

参考文献

[1] A. V. Krishnamoorthy et al. Scaling Optoelectronic-VLSI Circtuits into the 21st Century: A Technology Roadmap. *IEEE Journal of Selected Topics in Quantum Electronics*, Vol. 2, No. 1, pp. 55–76, 1996.

[2] M. Ishikawa et al. Optically Interconnected Parallel Computing Systems. *IEEE Computer*, Vol. 31, No. 2, pp. 61–68, 1998.

[3] R. K. Ahuja et al. *NETWORK FLOWS*. Prentice Hall, 1993.

[4] A.V.Krishnamoorthy et al. CMOS Static RAM Chip with High-Speed Optical Read and Write. *IEEE Photonics Technology Letters*, Vol. 9, No. 11, pp. 1517–1519, 1997.

[5] T. Komuro et al. Vision Chip Architecture Using General-Purpose Processing Elements for 1ms Vision System. *CAMP'97*, pp. 276–279, 1997.

[6] F.E.Kiamilev et al. Design of a 64-Bit, 100 MIPS Microprocessor Core IC for Hybrid CMOD-SEED Technology. *Proc. MPPOI'96*, pp. 53–60, 1996.

[7] N.McArdle et al. Implementation of a Pipelined Optoelectronic Processor: OCULAR-II. *Technical Digest, Optics in Computing 99*, pp. pp.72–74, 1999.