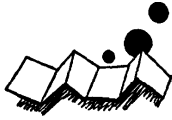


解説



2. ハードウェアアルゴリズムの基礎理論

2.3 VLSI モデルと面積複雑度†

都 倉 信 樹†† 萩 原 兼 一†† 和 田 幸 一†††

1. はじめに

この解説は VLSI 上に実現するハードウェアアルゴリズムを評価するための VLSI モデルを紹介し、その上での面積時間複雑度などの諸結果と、その基礎となるグラフの埋め込みについての最近の結果などについて述べる。なお、本文は(32)の続編として、重複をできるだけ避けることとして、より新しい結果について主として述べることにしたい。

2. VLSI モデルとアルゴリズムの評価尺度

VLSI 上でハードウェアアルゴリズムを実現する際、いくつも考えられるアルゴリズムの中からどれがよいかを決めるためには、その評価基準が問題となる。多くの評価基準が考えられる³⁵⁾。

1. 回路を収容するのに必要な全体面積 A
2. 計算時間(最悪時評価, 平均値評価) T
3. 一つの具体问题の入力を開始してから, 次の具体问题の入力を開始するまでの時間(入力周期) P_{period}
4. 並列度
5. 入出力ポート数
6. 適用範囲の広さ
7. 規模拡大の容易性
8. データ規模, データ量に対する柔軟性
9. VLSI アルゴリズムが検証しやすい構造をしていること。特に
 - a) 再帰的な構造
 - b) 規則的な接続
 - c) 規則的データの流れ
 - d) 構成要素の種類の少なさ
 - e) 構成要素の機能の単純さ

† VLSI Models and Area-time Complexity by Nobuki TOKURA, Ken'ichi HAGIHARA (Department of Information and Computer Sciences, Osaka University) and Koichi WADA (Department of Electrical and Computer Engineering Nagoya Institute of Technology).

†† 大阪大学基礎工学部情報工学科

††† 名古屋工業大学電気情報工学科

10. VLSI アルゴリズムのテストやデバッグが行いやすいこと

11. システム全体としての調和性

a) 入出力方法やデータの流れの一貫性

b) VLSI アルゴリズムの有効性

この評価項目には客観的評価のむずかしいものもある。実際には、このように多数の項目で評価し、適当な VLSI アルゴリズムを選ぶことになるのであるが、すべての項目を同時に扱うのは難しいので、最も重要なものからいくつかを選んで考える。こうして、理論的に取り扱える程度に単純化する。VLSI の場合、最も重要なものは、コストに大きく影響するという意味で、回路の占める面積、そして、その回路のパフォーマンスを代表する計算時間の2つに絞られる³⁵⁾。もちろん、多数の評価項目をとりこんだモデルの方が現実的であろうが、あまりに複雑ではなにも理論的な結果が出ない。次に述べるモデルは面積と時間の最も重要な2点に着目して抽象化し、多くの理論的結果を生み出したという意味で定式化が成功したといえよう。

問題. VLSI アルゴリズムはある問題を解くべく設計される。例えば、2つの行列の積を求める問題と単に言えば、どんな要素のデータをもち、それぞれの行、列の数はいくつかは述べていない。そのような問題の規模を定めるいくつかのパラメータをその問題の shape と呼ぼう。また、これに関連して、問題のサイズということばがよく使われる。これは、問題の規模を表現するのにふさわしい shape に依存して決まる、一つの評価値をいう。それに対応する回路の複雑さなどをそのサイズパラメータの関数として評価する。行列積の場合なら、要素のデータ型が固定されていると考えるなら、行列の要素数がサイズとして採用されるであろう。

shape が決まり、入力、出力データの2進表現法が決まると、入力変数 (x_1, x_2, \dots, x_n) 、出力変数 (y_1, y_2, \dots, y_m) が決まり、入出力対応を決める関数

$$F(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) = 1 \quad (*)$$

が決まる。 F はここでは陰関数の形で書いたが通常は $y_i = f_i(x_1, x_2, \dots, x_n)$ ($i=1, \dots, m$) の形に表現することが多い。なお、各入力変数の値を一つ定めたととき、その問題の具体問題 (instance) と呼ぶ。VLSI 回路 C がある shape の問題 P を解くというのは、入出力変数 $(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$ の C の入出力ポートへのわりあてを決めたときに、どの具体問題に対しても、入出力対応が(*)を満たすことをいう。この入出力のわりあてを、スケジュールと呼ぶ。なぜなら、VLSI では、ピン数の制限があり、一つのポートを複数個の入力変数にわりあて、時刻も決めるからである。

VLSI モデル

以下、VLSI モデルの仮定とその背景、および、他の仮定 (より強いもの、弱いものには、+, - をつけて) について説明してみる。

A1. 回路は面積 A の凸平面領域 R 上にある。

現在の VLSI は長方形のチップ上に平面的に構成される。チップ面積はコストに大きく影響する。ウェハには数千個という欠陥部分があり、これが回路の動作に影響し不良品となる。歩止り (欠陥のないチップの割合) は製造当初は 5~10% ということも普通のようにであり、この歩止りで単に面積を m 倍にすると、 $(0.1)^m$ というような良品率ということになり、面積に対して指数的にコストがかかるという議論³⁵⁾ になる。このために面積をできるだけ小さくすることが重要とされる。

+ a) R は長方形格子で配線は x, y 方向に平行なもののみとし、面積は回路をとりかこむ長方形の面積とする。

- b) 回路の素子、配線などの面積のみを考え、空地は A に含まない。下界の議論はこの形の仮定も用いるが、あるトポロジ (接続関係) の回路を平面上に埋め込むには、空地が生じることが避けられないことがあり、このことを含めて考えるにはこの仮定は適当でない。

A2. 配線は最小幅 λ をもつ。

Meed, Conway の設計ルールでは、ポリシリコン、拡散領域、メタル配線のそれぞれの配線幅 (間隔) は、 $2(2), 2(3), 3(3)$ λ となっている。 λ は最小加工線幅であり、現在は 2~3 ミクロンが実用レベルで、1 ミクロンをめざして競争されているという。ここでは、配線は一律に λ と単純化しているが、モデルとしては本質を失わない。

A3. R 中のどの点においても、高々 $\nu (\geq 2)$ 本の配線が交差する。

+ a) $\nu=2$ 。

A4. (一定遅延の仮定) 1 ビットのデータを配線を通して転送するのに、単位時間 ($\tau > 0$) 必要とする。ゲートでの計算時間も τ に含める。 τ は配線の長さによらず、一定とする。

この仮定は最も議論の多いところである。ある程度の規模までは、この仮定を満たすと考えられている。もし、長い配線と多くのファンアウトが必要なときは、それに合う大きさのドライバを使えばよい。しかも、そのために必要なドライバのための面積は、配線部分の面積の増大に比べてゆるやかにしか増大しないので、評価の際効いてこないという考え方である。より、現実的な仮定にたつとする立場の人は次のように仮定する。

+ a) (対数的遅延の仮定) 配線長の対数に比例した遅延があるとする。ただし、この際、配線長というのはあるドライバからその負荷となるノードへいたる全配線の総長を考える。

+ b) (線形遅延の仮定) 配線長に比例した遅延がある。

この仮定は現在の VLSI ではあまり適合しないといわれるが、この仮定のもとに議論する著者もある。a, b のように配線長によって時間が変わるといふモデルの場合は、回路がどのようにレイアウトされるかも決める必要があり、議論はそれだけ複雑になる。

A5. 入出力ポート、ゲート、フリップフロップなどの素子はいずれも $\rho (\geq \lambda^2)$ の面積を必要とし、お互いに R 上の点を共有することはない。配線とも重ならない。

A6. 入出力ポートは R 内のどこにあってもよい。この仮定より現実的な仮定として、

+ a) 入出力ポートは R の境界上におく (境界配置 (boarder layout) の仮定)

A7. スケジュール。ある具体問題の計算開始時刻を起点にして、各入力、出力のわりあてられる時刻とポートまたは、メモリを指定したものをスケジュールという。入力変数の値によって、スケジュールが変わらないとき、固定スケジュールといい、入力変数の値を見てから、以後の対応が変わるようなスケジュールを動的スケジュールとよぶ。さらに、ある (入出力) 変数がわりあてられるポートが固定されているなら、その変数は位置固定 (where-determinate) であると

いい、複数個のポートへわりあてられているなら、位置非固定という。また、変数値の印加される、あるいは、出力として現われる時刻が一つ決まっているとき、時刻固定 (when-determinate) であるといい、そうでないとき、その変数は時刻非固定であるという。すべての変数が時刻固定であるようなスケジュールを時刻固定スケジュールという。位置固定スケジュールはすべての変数が位置固定であるようなスケジュールをいう。

- a) 位置、時間ともに固定スケジュール
- b) 可変スケジュール

A8. 入力データは一度しか与えられない。

- a) 複数回のデータ入力が許される

A9. チップの外にメモリをもつ外部メモリ形の回路と、外部メモリをもたない独立回路とに分ける。さらに、外部メモリとして、RAM (ランダムアクセスメモリ) と、シフトレジスタ形のメモリとに分けて考える。これらはアクセス時間に差が生じる。

内部回路ももちろん記憶機能はある。このための面積はもちろん評価にいれる必要があるので、各ノードは $O(1)$ の面積を必要とすると仮定されている (A5)。外部メモリはそれが利用できると考えるだけで、対象回路の面積には含めない。外部メモリはアクセス時間が評価に効いてくる。

独立回路のときは、可変スケジュールというのは多少奇異かもしれないが、外部メモリ形回路では、メモリに中間結果を貯えたりするので、メモリへのアクセスは、データによって変わり、可変スケジュールという考え方はむしろ自然でもあろう。実用上、VLSI アルゴリズムを実現するチップは外部メモリを想定することが多いであろう。この場合、入力および出力変数の値はメモリ上において、やりとりすると考える。

なお、ある問題の具体問題の計算中に、チップのある部分回路が「あき」になり、次の具体問題のデータを読み込んで処理を始められるということもある。このときは、チップを有効に利用するために、パイプライン的に次々とデータを流しこんで、スループットを上げる。

- a) 独立回路
- b) 外部メモリ回路

以上、いろいろのモデルで使われる仮定を挙げたが、Brent と Kung のモデル⁹⁾ は、A1., A2., A3., A4., A5., A6., A7. a), A8.-a), A9. a) の仮定を採用している。最近の Thompson のモデル³¹⁾ は、A1., A2., A3.+a, A4.+a), A5., A7. a), A8.-a), A9. b) の仮定を

基礎として、外部メモリとしては、RAM 形と、シフトレジスタ形の2通りを用い、ソーティングについて詳しく検討している。

パフォーマンスの評価について

Thompson は面積と時間の評価をさらにきめこまかくするために次のように定義している³¹⁾。

イ) チップの全処理部分面積 $A_{\text{processing}}$ はその部分を囲む最小の長方形の面積。

ロ) 面積効率 A は全面積を並列度で割ったもの。

ハ) 並列度の個数の具体問題の計算の始まりから終わりまでの経過時間の平均値を全時間 T_{total} という。

ニ) チップの時間効率 T_d は、各具体問題のためのメモリアクセスの初めの時刻から終わりの時刻までの平均時間 (すべての具体問題についての……)

ホ) チップのピリオド T_p は T_{total} を並列度で割ったものである。 $T_p \leq T_d \leq T_{\text{total}}$ に注意。並列度=1 なら、 $T_p = T_d = T_{\text{total}}$ である。

ヘ) あるチップの I/O バンド幅はメモリに書いたり、読み出ししたりした (平均) 全ビット数を T_{total} で割ったものである。

3. グラフの埋め込み問題

2. の A1 で述べたように、安価に VLSI チップを作成するには、そのチップの面積を小さくすることが重要である。この問題を扱うために、グラフの埋め込み問題と呼ばれるものがある。設計した回路 C を VLSI チップとしてレイアウトすることを、次のように抽象化する。レイアウトされるウェハ面として2次元格子状グラフ G_s を考え、格子上の点の位置だけに素子を配置でき、素子間の結線は G_s 上のパスを対応させることにより行う。 G_s の1点には、高々1つの素子だけを配置でき、高々1つの結線のみが G_s の1辺を占有できる (2層配線の技術を想定していると考えてよいが、一般に ν 層 (ただし、 ν は定数) にすることを許しても、オーダの議論では変わらない)。 C を、素子を頂点、ワイヤを辺とするグラフ G_c と見なす。上記規則を満たしつつ、 G_c を G_s 上にどのように (特に面積が小さくなるように) 描くかを、グラフの埋め込みという (2章における、A1. の a), A3. の + a), A5. の仮定を参照)。

解説 32) で、すでにグラフの埋め込み問題に関するいくつかの結果を紹介したが、ここでは、一般的な埋め込み方法と、特に興味深い平面的グラフに関する未解決問題を簡単に述べる。本節では、次数 (グラフの

頂点に接続している辺数の最大値) 4 以下のグラフ (以降, 単に“グラフ”と略す) のみ対象とする。

3.1 セパレータと埋め込みアルゴリズム

“セパレータ¹⁹⁾” という概念と, セパレータを用いた分割統治法による埋め込みアルゴリズム^{16), 36)} を紹介する。グラフ G の頂点数を $n(G)$ と表わす。

【定義】 ある関数 $s(n)$ に関して, グラフ G が 1 頂点から成るか, あるいは次の①及び②を満たすとき, G は $s(n)$ セパレータ (separator) を持つ, あるいは G は $s(n)$ 分離可能 (separable) であると言う。

① 高々 $s(n(G))$ 個の辺を見いだすことができ, それらの辺を除去すると, G を二つのグラフ G_1 と G_2 に分離 (G_1 と G_2 の間に辺がないように) できる。ただし, G_1 と G_2 は次の条件を満たす。

$$n(G_1) \geq n(G)/3 \text{ かつ } n(G_2) \geq n(G)/3$$

② $G_i(i=1, 2)$ は, $s(n(G_i))$ 分離可能である。

特に, ①より強い次の条件③を満たす G_1 と G_2 とを見いだすことができるとき, G は $s(n)$ 強セパレータを持つ, または, G は $s(n)$ 強分離可能であると言う。

$$③ \quad |n(G_1) - n(G_2)| \leq 1$$

【定義】 グラフ族 G に属するすべてのグラフが $s(n)$ 分離可能のとき, G は $s(n)$ 分離可能と言う。 $s(n)$ 強分離可能に対しても同様である。

【定義】 簡単のため n は 4 の巾乗とする。任意の関数 $s(n)$ に対して, 汎関数 d を次のように定義する。

$$d(s(n)) = \sum_{i=0}^k 2^i \cdot s(n/4^i)$$

ここで, $k = \log_4 n$ である。また, d は次のようにも書ける。

$$d(s(n)) = s(n) + 2d(s(n/4)) \tag{式1}$$

表-1 に汎関数 d の性質を示す。

【埋め込みアルゴリズム】

入力: $s(n)$ 強分離可能なグラフ G 。ただし, 関数 $s(n)$ は, 単調非減少とする。以降, $n = n(G)$ とする。

出力: 一辺が $L(n) = \sqrt{n} + 6d(s(n))$ の正方形領域への G の埋め込み。

方法: 簡単のため n は 4 の巾乗とする (そうでなけ

れば, ダミー頂点を追加すればよい)。

① G から高々 $s(n)$ 本の辺を除去して, 2 つのグラフ G_1 と G_2 ($n(G_1) = n(G_2) = n/2$) にする (G は $s(n)$ 強分離可能であるから, 必ずこのようにできる)。さらに, $G_i(i=1, 2)$ から, 高々 $s(n/2)$ の辺を除去して, 2 つのグラフ G_{i1} と G_{i2} ($n(G_{i1}) = n(G_{i2}) = n/4$) にする。

② 一辺が $L(n/4) = (n/4)^{1/2} + 6d(s(n/4))$ の正方形領域に, $G_{ij}(i, j=1, 2)$ を再帰的に埋め込む。

③ G_{i1} と G_{i2} ($i=1, 2$) に, ①で削除した $s(n/2)$ 本の辺を追加して埋め込むことにより, G_i の埋め込みを作成する。

④ G_1 と G_2 に, ①で削除した $s(n)$ 本の辺を追加して埋め込むことにより, G の埋め込みを作成する。■

上の③と④で $2s(n/2) + s(n) (\leq 3s(n))$ 本の辺の追加埋め込みするが, 1 本当たり縦横それぞれ高々 2 単位スペース増加させる (1 単位スペース増加させるとは, そのときの埋め込みの“図”を縦あるいは横方向に切り裂いて, 1 単位平行移動させることを示す) だけで, 追加できる (切り裂かれて空いた領域に追加すべき辺を配線する)。したがって, 式 1 より G は一辺

$$2 \cdot L(n/4) + 6s(n) = L(n) = \sqrt{n} + 6d(s(n))$$

の正方形領域に埋め込み可能である (定理 1)。

【定理 1】 関数 $s(n)$ は, 単調非減少とする。頂点数 n のグラフ G が $s(n)$ 強分離可能ならば, G は一辺が $O(\max(\sqrt{n}, d(s(n))))$ の正方形領域に埋め込み可能である。■

上で示したアルゴリズムを用いてグラフを埋め込んだときの強セパレータと面積の関係を表-2 にまとめる。

グラフの埋め込みに関しては, 頂点数が n であることより, $O(n)$ の面積は必要である。したがって, 上の $\alpha < 1/2$ のセパレータのグラフ族の面積は, オーダ的に最適である。また, $\alpha = 1/2$ と $\alpha > 1/2$ のグラフ族において, 表に示した面積をどうしても必要とするグラフが存在すること, すなわち, グラフ族に関しては, 上のアルゴリズムはオーダ的に最適である (existen-

表-1 汎関数 d の性質

$s(n) = n^\alpha$	$d(s(n))$
$\alpha < 1/2$	$O(\sqrt{n})$
$\alpha = 1/2$	$O(\sqrt{n} \cdot \log n)$
$\alpha > 1/2$	$O(n^\alpha)$

表-2 強セパレータと面積の関係

$s(n) = n^\alpha$	埋め込み面積
$\alpha < 1/2$	$O(n)$
$\alpha = 1/2$	$O(n \log^2 n)$
$\alpha > 1/2$	$O(n^{2\alpha})$

表-3 汎関数 Γ の性質

$s(n)$	$\Gamma(s(n))$
定数	$O(\log n)$
$\log n$	$O(\log^2 n)$
n^a	$O(n^a)$

tially optimal) ことが知られている¹⁷⁾。しかし、例えば2次元格子状グラフは、 $\alpha=1/2$ のクラスに属するが、明らかに $O(n)$ で埋め込める。ところが、上のアルゴリズムを用いると $O(n \log^2 n)$ の面積を使ってしまう。したがって、universally optimal ではない。文献3)では、セパレータでは精密に議論し切れないところを、“bifurcator” という概念を導入して、埋め込み方法を改善している。このような改善を進めることが今後の課題である。

3.2 平面的グラフの $O(n \cdot (\log n)^2)$ 面積の埋め込みとその応用

本節では、次数4以下の平面的グラフを、単に“平面的グラフ”と言う。

【定義】 任意の関数 $s(n)$ に対して、汎関数 Γ を次のように定義する。

$$\Gamma(s(n)) = \sum_{i=0}^k s((2/3)^i n)$$

ここで、 $k = \lceil \log_{3/2} n \rceil$ である。■

表-3 に汎関数 Γ の性質を示す。

【補題1】 グラフ族 G が $s(n)$ 分離可能ならば、 G は $\Gamma(s(n))$ 強分離可能である。■

【補題2】¹⁹⁾ 平面的グラフの族は、 $(8n)^{1/2}$ 分離可能である。■

補題1, 2 及び定理1より次の定理が成り立つ。

【定理2】 平面的グラフは、 $O(n \log^2 n)$ の面積で埋め込み可能である。■

グラフ G を平面上に描写したとき、どのように描写しても少なくとも c 個の場所で2辺が交差するとき、 G の交差数は c であると言う。そのような埋め込みにおいて、 c 個の交差点を、仮想的な頂点で置き直してできるグラフは、頂点数 $n+c$ の平面(的)グラフとなる。したがって、次の定理と系が成り立つ。

【定理3】 ある関数 $A(n)$ が存在して、すべての平面的グラフが面積 $A(n)$ で埋め込み可能とすれば、次数4かつ交差数 c の任意のグラフは、面積 $O(A(n+c))$ で埋め込み可能である。■

【系】 交差数 c の任意のグラフは、面積 $O((n+c) \log^2(n+c))$ で埋め込み可能である。■

3.3 面積 $O(n)$ で埋め込めるグラフ族

面積 $O(n)$ で埋め込める平面的グラフの族はどのようなものかを解明することも興味深い問題である。今までに、知られているそのようなグラフ族として、解説32)では、木、外平面的グラフ、 X 木を紹介したが、文献10)では、外平面的グラフ族を真に含む次のグラフ族 $DT(k)$ も面積 $O(n)$ で埋め込めることを示している。

【定義】 グラフ族 $DT(k)$: 定数 k に対して、グラフ族 $DT(k)$ は、次の性質を持つ平面的グラフ G のすべての集合とする。

平面的グラフを平面上に交差なしに描写したとき、グラフの外側の無限に広がった領域を外面と言う。 G の任意の頂点 v に対して、外面に位置するある頂点 u が存在して、 u と v を結ぶパス長が高々 k となるような、 G の平面上での交差のない描写が存在する。■

文献34)では、文献10)と類似な議論を“準外平面的”と呼ばれるグラフを対象として行っている。準外平面的グラフの族も、外平面的グラフ族を真に含む。

面積 $O(n)$ で埋め込める平面的グラフの族を、どこまで広くできるかということも今後の研究課題である。

3.4 平面的グラフの埋め込みの面積下界

【定義】¹⁷⁾ m を2の巾乗とする。 $m \times m$ の行列状に全非連結に m^2 個の頂点を置く。各行の m 頂点に対して、それらを葉とする完全2分木(ただし、葉以外の $m-1$ 個の頂点は新たに追加する)をそれぞれ m 個作る。各列に対しても同様にする。このようにしてできた頂点数 $3m^2 - 2m$ のグラフを、 $m \times m$ メッシュ状木(mesh of trees)と言う。*以降では、 $n = 3m^2 - 2m$ とおく。

【 $m \times m$ メッシュ状木の性質】¹⁷⁾

- ① このグラフ族は、 $O(\sqrt{n})$ 分離可能である。
- ② 交差数は、 $O(n \log n)$ である。

【定理4】¹⁷⁾ $m \times m$ メッシュ状木の埋め込みには、 $\Omega(n \cdot \log^2 n)$ の面積を必要とする。■

$m \times m$ メッシュ状木の交差数と定理3より、 $m \times m$ メッシュ状木は、面積 $A(h \cdot n \cdot \log n)$ (ただし、 h はある定数) で埋め込める。ところで、定理4より、

$$A(h \cdot n \cdot \log n) = \Omega(n \cdot \log^2 n)$$

である。したがって、 $A(N) = \Omega(N \cdot \log N)$ となり、次の定理が成り立つ。

【定理5】³⁵⁾ 埋め込みに $\Omega(n \log n)$ の面積を必要

* 最初に行列状に置いた頂点は、行方向の木と、列方向の木とに葉として共有されている。

とする平面的グラフの無限族 (infinite family) が存在する。■

また 17) は、定理 5 の平面的グラフの無限族として、具体的イメージが明確な、メッシュの木 (tree of meshes) と呼ばれるグラフを示している。さらに、メッシュの木は面積 $O(n \log n)$ で実際埋め込めることも示している。

3.5 平面的グラフに関する未解決問題

① 任意の平面的グラフを面積 $O(n \log^2 n)$ 以下のオーダで埋め込めるか (埋め込みアルゴリズムがあるか)?

② 平面的グラフ族の埋め込みの面積下界は $\Omega(n \log n)$ 以上にできないか? すなわち、平面的グラフで、埋め込み面積が $\Omega(n \log n)$ 以上のものが存在するか?

③ 面積 $O(n)$ で埋め込める平面的グラフの族を拡げる。

3.6 制約のある埋め込み問題

グラフの埋め込み問題は、電子回路の VLSI 化を抽象化した問題であるので、VLSI の観点から、単に面積を小さく埋め込むだけでなく、次に挙げるような制約条件を満たすような埋め込みも重要視される。

I 埋め込んだときに、最長の辺が最短になるようにする。

次のことが知られている。ここで、埋め込みにおける、“地理的に最も離れた” 端点をもつ辺の長さのことを、最長辺長と言う。

① 最長辺長が $\theta((n/\log n)^{1/2})$ で埋め込み面積が $\theta(n \log n)$ の平面的グラフが存在する¹⁷⁾。

② 最長辺長が $\theta(n^{1/2} \log n / \log \log n)$ 、埋め込み面積が $\theta(n \cdot \log^2 n)$ で $O(n^{1/2})$ 分離可能なグラフ (平面的グラフではない) が存在する。その一例はメッシュ状木である¹⁷⁾。

③ 任意の $r \geq 3$ に対して、 $\alpha = 1 - 1/r$ とおくと、最長辺長が $\theta(n^\alpha)$ 、埋め込み面積が $\theta(n^{2\alpha})$ で $O(n^\alpha)$ 分離可能なグラフ (平面的グラフではない) が存在する¹⁷⁾。

④ 完全 2 分木の最長辺長は $\theta(n^{1/2} / \log n)$ である²⁴⁾。

II 交差数が最小になるようにする。
次のことが知られている¹⁵⁾。

① 平面的グラフのある埋め込みにおける交差点数 X (グラフの交差数ではない) と埋め込み面積 A に関して、 $AX = \theta(n^2)$ となる。

② 3.1 のアルゴリズムを用いて平面的グラフを埋め込んだときに、その埋め込みにおける交差点数が $\Omega(n/\log^2 n)$ を必要とするものが存在する。

III 配置が規則的になるようにする。

IV いくつかの頂点の配置に制約 (例えば、境界上に位置しなければならない⁴²⁾ など) がある。

V グラフ族のすべてのグラフが、(頂点数などが大きくなっても) 数種類のグラフの簡単な組み合わせで構成できる。

また、2次元ではなく3次元立体格子状グラフへの埋め込みも研究^{11), 26), 27)} されているが、別の機会に述べる。

4. 面積時間複雑度

4.1 面積時間複雑度の下界

この節では、2. で述べた VLSI モデルに基づいて面積時間複雑度の下界を導出するのに用いられる手法について述べる。以下では、A1. +a), A2. ~A6., A7. の a), A8., A9. の a) を仮定し、次のソーティングの問題を例に用いる。

【例 1】与えられた自然数 n と k に対して、 n 個の k ビット 2 進数のソーティングの問題の具体問題は次のように表される。 $X_1 = x_{11} \dots x_{1k}$ を 1 番目の 2 進数、 $X_2 = x_{21} \dots x_{2k}$ を 2 番目の 2 進数、一般に $X_i = x_{i1} \dots x_{ik}$ を i 番目の 2 進数とする。出力 $Y_j = y_{1j} \dots y_{nj}$ を入力の j 番目に小さい数とする。ソーティングの問題はこれら具体問題の無限列である。このソーティングの問題を $\text{SORT}(n, k)$ と表す。■

問題に対する解とは、回路の無限列で、各回路が具体問題を計算することである。ここでは、特定のサイズに依存した回路ではなく、すべてのサイズに対する回路を考えている。

回路によって実行される計算の過程は、図-1 の直方体によって表現できる。図-1 において、 $x-y$ 平面は、回路の面積 A を表わし、 z 軸は計算時間 T を表わす。この直方体を用いて、次の 3 種類の面積時間複雑度の下界を導出できる³⁵⁾。

1. A : A はこの直方体を $x-y$ 平面に平行な平面で切った断面に対応する。回路が問題を解くとき、情報がこの断面を横ぎって上へ流れる。断面の単位面積を通しては高々 1 ビットの情報しか流しえないという性質を利用して、 A の下界を導出する。

2. AT : AT はこの直方体の体積に対応する。直

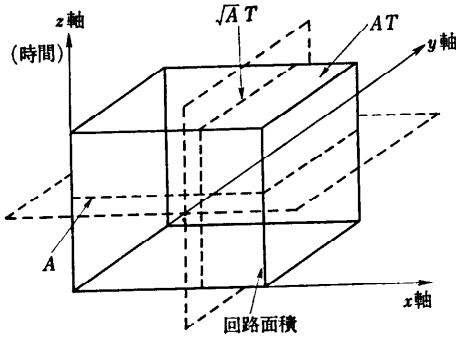


図-1 面積時間積の下界導出のための直方体

立方体の任意の点において、高々1ビットの入出力しかできないので、入出力ビットの合計は AT を越えないことを用いて AT の下界を導出する。

3. $\sqrt{AT}(AT^2)$: \sqrt{AT} はこの直方体を $y-z$ 平面に平行な平面で切った断面に対応する*。ある量の情報がこの断面を横ぎるならば、この情報量は \sqrt{AT} を越えないことを利用して \sqrt{AT} (したがって AT^2) の下界を導出する。

Aの下界

Aの下界の導出における基本的な考えは、面積Aの回路は高々Aビットの情報しか記憶できないということである。時刻tまでにAビット以上の入力回路に読みこまれたとする。読みこまれた入力変数への異なった2つの値の割当てに対して、もしtにおいて、回路が同じ状態にあるならば、回路はこの2つの状態を区別できずに同じ出力を出さざるをえない。このことから次の定理が成り立つ。

【定理6】^{35),39)} 問題を解く回路の面積をAとする。 $f(n)$ をある関数とし、ある時刻t以前に入力された入力変数への異なった割当てが $2^{f(n)}$ 個あり、t以後の同じ入力に対して、それぞれ異なった出力をするならば、 $A \geq f(n)$ が成り立つ。■

【例2】 ソーティング回路の面積の下界

$\text{SORT}(n, k)$ ($k > \log n$) を解く回路のAの下界を考える。入力2進数のn個の最下位ビットに着目すると、ソーティングの性質より、このn個の入力をすべて読み込むまでは何も出力ができない。このn個の入力変数には、 2^k 個の異なった割当てがあるので、定理6より、 $A = \Omega(n)$ が得られる。ここで、 $k > \log n$ という条件は本質的である。例えば、 $k=1$ ならば $A = O(\log n)$ で $\text{SORT}(n, 1)$ を解くことができ $A = \Omega(n)$

*ここで回路の長い方の辺、短い方の辺をそれぞれx軸、y軸に沿っておくとする。

とはならない。■

ATの下界

ATの下界を導出する原理は、面積Aの回路が、nビットを入力(出力)するのに少なくとも n/A 時間必要とすることである。このことより、次の定理が成り立つ。

【定理7】^{35),40)} Dを問題Pによって用いられる入出力ビットの数とする。このとき、Pを解く任意の回路の面積をA、計算時間をTとすると、 $AT \geq D$ が成り立つ。■

【例3】 ソーティング回路のATの下界

$\text{SORT}(n, k)$ ($k > \log n$) は $2nk$ 個の入出力ビットを持つので $AT = \Omega(nk)$ となる。■

AT^2の下界

これまで、いろいろな問題を解く回路の AT^2 の下界が導出されている^{1),9),13),20),29),35),38)}。導出の際に用いられる手法はそれぞれ異なっているが、本質的な部分には大差はない。以下では、最も一般的でVLSIモデルが変化しても応用がきくと思われる手法を紹介する^{1),35)}。また、その手法を用いて得られる下界はソーティングなどについては、定理6や7で得られる下界よりも、その下界に近いオーダーで実現できるという意味において、強い下界である。

Pを問題、nを問題のサイズパラメータとする。X,Yをそれぞれ入力変数の集合、出力変数の集合とする。Pとnに対する分割(partition)はXを互いに素な XL と XR に分け、Yを互いに素な YL と YR に分ける。ここで、 $X = XL \cup XR$, $Y = YL \cup YR$ である。Pとnに対する入力変数への割当ては、Xから $\{0,1\}$ への写像である。

ある分割を $\pi = (XL, XR, YL, YR)$ とし、 α, β を入力変数への割当てとする。 $\alpha L(\alpha R)$ を α の定義域を $XL(XR)$ に制限したものを表わすとするまた、 $\alpha L\beta R$ は XL は α に一致し、 XR は β に一致するものとする。

P,nと分割 π に対する欺集合(fooling-set)は入力変数への割当ての集合Aで以下の条件を満たす: 任意の $\alpha, \beta (\in A)$ に対して、次の(1)~(4)のうちの一つが成り立つ。

- (1) $\alpha L\beta R$ と α とは、YLのある変数の値が異なる。
- (2) $\alpha L\beta R$ と α とは、YRのある変数の値が異なる。
- (3) $\beta L\alpha R$ と α とは、YLのある変数の値が異なる。

なる。

(4) $\beta L\alpha R$ と α とは, YR のある変数の値が異なる。

Z を X の部分集合とする。分割 π が P, n, Z に対して, 認容的 (acceptable) というのは, Z の $1/3$ 以上 $2/3$ 以下の要素が XL に含まれていることをいう。

$I(P, n, \pi)$ を P, n, π の最大の敷集合の要素数の (2 を底とする) 対数と定義する。 $I(P, n, Z)$ をすべての P に対して認容的な π, n, Z における $I(P, n, \pi)$ の最小値とする。 $I(P, n)$ (P の伝達情報量 (information contents) という) をすべての Z における $I(P, n, Z)$ の最大値と定義する。

【例4】³⁵⁾ $\text{SORT}(n, k)$ ($k > \log n$) に対する伝達情報量 $I(P, n)$ (の下界) を求める。 $I(P, n)$ の下界は定義より, ある特定の Z に対する $I(P, n, Z)$ を求めればよい。例2と同様に, 各入力2進数の最下位ビットに着目して, その集合を Z とすれば, $I(P, n) \geq I(P, n, Z) \geq n/3$ が得られる。■

$I(P, n)$ は問題に対して決まる量であり回路とは直接関係しない。 $I(P, n)$ と回路を結び付けるために, テューリング機械などで用いられる交差列 (crossing sequence) という概念を用いる。

縦の長さ h , 横の長さ w ($w \geq h$ とする) の回路において, 縦の線で回路を2つに分割すると (この線を境界線と呼ぶ), その境界線上には h 個の導線または, 素子が横切る。回路の動作中は導線は0か1の値をとり, 素子は素子への入力変数の値によって定まる。したがって, 各素子への入力数の上限を l とすると, この境界線は高々 l ビットの列で特徴づけられる。ある回路 C に対して, 入力変数への割当て α と境界線が与えられたとき, α に対する境界線上の交差列は, T を C の計算時間, w_t ($0 \leq t \leq T-1$) を時刻 t において境界線上に現れるビット列としたとき, $w_0 \dots w_{T-1}$ である。

このように交差列を定義すると問題 P と分割 π に対する敷集合と P を解く回路の境界線上の交差列の間には1対1の対応がつく。すなわち, 敷集合の異なる2つの要素 (入力変数への割当て) に対しては, 同じ交差列をもたないことが示される。伝達情報量 I を持つ問題 P を解く回路 C において, $I(P, n, Z) = I$ となる分割に対応する C 中の境界線上の交差列を考えると, 交差列中のビット数は高々 lhT ゆえ, $lhT \geq I$ が成り立つ。また, l は定数, $A \geq h$ であるので, 次の定理が成り立つ。

【定理8】^{1), 35)} P を伝達情報量 I を持つ問題とする。 P を解く回路の面積を A , 計算時間を T とすると, $AT^2 = \Omega(I^2)$ が成り立つ。■

【例5】 定理8と例4より, $\text{SORT}(n, k)$ ($k \geq \log n$) を解く任意の回路に対して, $AT^2 = \Omega(n^2)$ が成り立つ*。

以下では, VLSI モデルの仮定を変更して得られる結果を紹介する。

境界配置 (A6.+a)

通常の回路では, 入出力ポートは回路の境界上に置かれることが多い。この仮定のもとで次の定理が得られる。

【定理9】³⁵⁾ P を伝達情報量 I をもつ問題とし, P を解く回路の面積を A , 計算時間を T とする。また, r をサイズ n の問題の入力ビット数とすると, $AT^2 = \Omega(Ir)$ が成り立つ。■

一般に $r \geq I$ であるので, 境界配置の仮定により強い下界が得られる。また, I が入出力数に比べて小さくなるような n 入力 m 出力論理関数に対しても AT^α ($\alpha > 1$) の下界が境界配置の仮定により影響をうけることが知られている。

【定理10】^{40), 41)} $f: \{0, 1\}^n \rightarrow \{0, 1\}^m$ に対し, 出力変数 b_j が真に依存する入力変数**の数を n_j , 入力変数 a_i に真に依存する出力変数の数を m_i とし,

$$n = \max_{1 \leq j \leq m} n_j, \quad m = \max_{1 \leq i \leq n} m_i$$

とする。境界配置を仮定しなければ,

$$AT^\alpha = \Omega(\max(n, m) \cdot [\max(\log n, \log m)]^{\alpha-1}) \quad (\alpha \geq 1)$$

であり, 境界配置を仮定すれば,

$$AT = \Omega(\max(n, m)),$$

$$AT^\alpha = \Omega(\max(n, m) \cdot [\max(\log n, \log m)]^\alpha) \quad (\alpha > 1)$$

が成り立つ。■

回路内に乱数発生器を備えた確率的回路³⁵⁾, 繰り返し入力を許す回路²⁸⁾などに対する AT^2 の下界も伝達情報量の定義を拡張することにより同様の下界が得られる。

4.2 面積時間複雑度の上界

いろいろな問題に対して, 面積時間複雑度 (特に AT^2) の下界のオーダ (あるいはそれに近いオーダ) で実現する回路をどのように構成するかについては,

* 最近, $k \geq 7 \log n$ に対して $AT^2 = \Omega((n \cdot \log n)^2)$ が得られた¹⁹⁾。

** 出力変数が入力変数に真に依存するとは, その入力変数の値が異なれば, 出力値が異なることである。

表-4 AT^2 に関するいくつかの結果

問題 (入力数/出力数)	AT^2	T の 範囲 [$f(n), g(n)$]*	注
2 進 乗 算 ($n, n/2n$)	$\Omega(n^2)$ $O(n^2)$	— [$\log n, \sqrt{n}$]	定理 8 (22)
ソーティング (kn/kn)** ($k > 7 \log n$)	$\Omega(n^2 \log^2 n)$ $O(n^2 \log^2 n)$	— [$\log n, \frac{\sqrt{n}}{\sqrt{n \log n}}$]	(18) (5) (18)
コンボリューション (kn/kn)	$\Omega((kn)^2)$ $O((kn \log n)^2)$	— —	定理 8 (2)
CFL の 認 識 ($n/1$)***	$\Omega(n^2)$ $O(n^2 \log^2 n)$	— —	定理 8 (8)

* $T = \Omega(f(n))$ かつ $T = O(g(n))$ を満たす T に対して AT^2 の上界が成立する。

** n 個の k ビット 2 進数を入力とする。

*** n は入力列の長さを表す。

多くの研究がなされており、主な結果は文献 32) に示されている。ここでは、それ以後に得られた、または改善された結果を表-4 に示す。最近は単に AT^2 の下界を達成するだけでなく、計算時間を変化させた場合でも、 AT^2 の下界を達成できるかどうかには注意が払われている^{5), 22)}。これらの結果は、各問題の再帰的な性質に着目し、分割統治法 (divide and conquer) を用いて回路を構成しているものが多い^{2), 21)}。

また、用いられるネットワークとしては、データの置換が高速に行えるシャフル・エクステンジ・グラフや CCC (Cube Connected Cycles) の変形したものが利用されている^{4), 37)}。ソーティング、フーリエ変換については、従来のアルゴリズムを VLSI モデルの枠内で実現したとき、 AT^2 がどのようになるかが評価されている^{30), 31)}。また、グラフの問題 (例えば、連結成分を求める問題) に対しても、VLSI モデルのもとで、面積時間積を議論しているものもある^{12), 13)}。

5. む す び

本稿では、ハードウェアアルゴリズムを評価するための VLSI モデルのもとで面積、及び面積時間複雑度に関する最近の結果を概説した。ハードウェアアルゴリズムを 2. で述べた面積時間以外の尺度で定量的に評価できるモデルの確立が今後の課題であろう。

参 考 文 献

1) Aho, A.V., Ullman, J.D. and Yannakakis, M.: On Notions of Information Transfer in VLSI Circuits, STOC, pp. 133-139 (1983).

2) Baudet, F.P., Preparata, F.P. and Vuillemin, J.E.: Area-Time Optimal VLSI for Convolution, IEEE Trans. Comput., Vol. C-32, No. 7, pp. 684-688 (1983).

3) Bhatt, S.N. and Leighton, F.T.: A Framework for Solving VLSI Graph Layout Problems, JCSS, Vol. 2, No. 8, pp. 300-343 (1984).

4) Bilardi, G. and Preparata, F.P.: An Architecture for Bitonic Sorting with Optimal VLSI Performance, IEEE Trans. Comput., Vol. C-33, No. 7, pp. 646-651 (1984).

5) Bilardi, G. and Preparata, F.P.: A Minimum Area VLSI Network for $O(\log n)$ Time Sorting, STOC, pp. 64-70 (1984).

6) Bongiobanni, G.: The VLSI Structure for the Discrete Fourier Transform, IEEE Trans. Comput., Vol. C-32, No. 8, pp. 750-754 (1983).

7) Bongiobanni, G.: A VLSI Network for Variable Size FFT's. IEEE Trans. Comput., Vol. C-32, No. 8, pp. 756-760 (1983).

8) Brent, R.P. and Goldschlager, L.M.: Some Area-Time Tradeoffs for VLSI, SIAM J. Comput., Vol. 11, No. 4, pp. 737-746 (1982).

9) Brent, R.P. and Kung, H.T.: The Area-Time Complexity of Binary Multiplication, JACM, Vol. 28, No. 3, pp. 521-534 (1981).

10) Dolev, D. and Trickey, H.: On Linear Area Embedding of Planar Graphs, STAN-CS-81-876, Stanford Univ (1981).

11) 萩原, 鈴木, 都倉: グラフの 3 次元格子空間への埋込みに関する一考察, 信学論(D), Vol. J66-D, No. 12, pp. 1376-1383 (1983).

12) Hambrusch, S.E.: VLSI Algorithms for the Connected Component Problem. SIAM J. Comput., Vol. 12, No. 2, pp. 354-364 (1983).

13) Ja'Ja', J.: The VLSI Complexity of Selected Graph Problems, JACM, Vol. 31, No. 2, pp. 377-391 (1984).

14) Ja'Ja', J. and Prasannakumar, V.K.: Information Transfer in Distributed Computing with Applications to VLSI, JACM, Vol. 31, No. 1, pp. 150-162 (1983).

15) 木本, 丸岡, 木村: グラフの埋込みに必要な面積と交差数の兼ね合いについて, 信学論(D), Vol. J66-D, No. 7, pp. 834-840 (1983).

16) Leiserson, C.E.: Area-efficient Layouts (for VLSI), 21-st FOCS, pp. 270-281 (1980).

17) Leighton, F.T.: New Lower Bound Techniques for VLSI, Mathematical Systems Theory, 17, pp. 47-70 (1984).

18) Leighton, F.T.: Tight Bounds on the Complexity of Parallel Sorting, STOC, pp. 71-80 (1984).

19) Lipton, R.I. and Tarjan, R.E.: A Planar

- Separator Theorem, SIAM J. Applied Math. Vol. 36, No. 2, pp. 177-189 (1979).
- 20) Lipton, R. J. and Sedgewick, R.: Lower Bounds for VLSI. 13-th STOC, pp. 300-307 (1981).
 - 21) Luk, W. K. and Vuillemin, J. E.: Recursive Implementation of Optimal Time VLSI Integer Multiplication, VLSI '83, pp. 155-169 (1983).
 - 22) Mehlhorn, K. and Preparata, F. P.: Area-Time Optimal VLSI Integer Multiplier with Minimum Computation Time, Info. and Control. 58, pp. 137-156 (1983).
 - 23) Nath, D., Maheshwari, S. N. and Bhatt, P. C.: Efficient VLSI Networks for Parallel Processing Based on Orthogonal Trees, IEEE Trans. Comput., Vol. C-32, No. 6, pp. 569-581 (1983).
 - 24) Paterson, M., Ruzzo, W. J. and Snyder, L.: Bounds on minimax edge for complete binary trees. STOC (1981).
 - 25) Preparata, F. P.: An Area-Time Optimal Mesh-Connected Multiplier of Large Integers, IEEE Trans. Comput., Vol. C-32, pp. 194-198 (1983).
 - 26) Preparata, F. P.: Optimal Three-Dimensional VLSI Layouts, Mathematical Systems Theory, 16, pp. 1-8 (1983).
 - 27) Rosenberg, A. L.: Three-dimensional VLSI, I: A case study, RC 8745, IBM T. J. Watson Research Center (1981).
 - 28) Savage, J. E.: Multilective Planar Circuit Size, 20th Ann. Allerton Conf. on Comm., Control, and Comput., pp. 665-671 (1982).
 - 29) Thompson, C. D.: Area-Time Complexity for VLSI, 11-th STOC, pp. 81-88 (1979).
 - 30) Thompson, C. D.: Fourier Transform in VLSI, IEEE Trans. Comput., Vol. C-32, No. 11, pp. 1047-1057 (1983).
 - 31) Thompson, C. D.: The VLSI Complexity of Sorting, IEEE Trans, Comput., Vol. C-32, No. 12, pp. 1171-1184 (1983).
 - 32) 都倉信樹: VLSI アルゴリズムおよび面積時間複雑度, 情報処理, Vol. 23, No. 3, pp. 176-186 (1982).
 - 33) 都倉, 萩原, 増沢: VLSI アルゴリズムとその評価, 信学研資, AL 81-87 (1981).
 - 34) 戸田, 笠井: VLSI モデルへのグラフの埋め込みについて, 信学論(D), Vol. 167-D, No. 5, pp. 571-578 (1984).
 - 35) Ullman, J. D.: COMPUTATIONAL ASPECTS OF VLSI. Computer Science Press (1984).
 - 36) Valiant, L. G.: Universality Considerations in VLSI Circuits, IEEE Trans. Comput. Vol. C-20, No. 2, pp. 135-140 (1980).
 - 37) Vuillemin, J. E.: A Very Fast Multiplication Algorithm for VLSI Implementation, Integration VLSI J., pp. 33-52 (1983).
 - 38) Vuillemin, J. E.: A Combinational Limit to the Computing Power of VLSI Circuits, IEEE Trans. Comput., Vol. C-32, No. 3, pp. 294-300 (1983).
 - 39) 和田, 萩原, 都倉: VLSI 回路モデルにおける面積複雑度, 信学論(D), Vol. 165-D, No. 4, pp. 478-485 (1982).
 - 40) Wada, K., Hagihara, K. and Tokura, N.: Area-Time Optimal Fast Implementation of Several Functions in a VLSI Model, IEEE Trans Comput., Vol. C-33, No. 5, pp. 455-462 (1984).
 - 41) 和田, 萩原, 都倉: VLSI モデルにおける面積時間複雑度に関する一考察, 信学研資, CAS 84 (1984).
 - 42) 安浦, 矢島: 論理回路の VLSI での面積について, 信学論(D), Vol. 165-D, No. 8, pp. 1080-1087 (1982).

(昭和 59 年 12 月 21 日受付)