2009－AL－123 (8)
2009／3／5

配線遅延を考慮した回路モデル上での加算及び乗算の計算複雑さ

長瀬 哲也† 高木 一義† 高木 直史†

† 名古屋大学大学院情報科学研究科

概要 ある問題を解くアルゴリズムの性能を評価するためには，その問題を解くために本質的に必要となる計算資源の量，すなわち問題の計算複雑さを知ることが重要となる．集積回路上での問題の計算複雑さを議論する殆どの研究においては，計算時間の評価において配線遅延を考慮していない．しかし，集積回路の微細化の進展に伴い，論理素子の遅延に対して配線遅延が相対的に増大している．従って，配線遅延を考慮した回路モデル上での計算複雑さを議論することが重要であると考えられる．本稿では，加算及び乗算について，配線遅延を考慮した回路モデル上での計算複雑さを示す．計算量の下界の評価の結果，配線遅延の影響が大きなモデルほど加算の計算量は大きくなったが，乗算の計算量には変化がなかった．計算量の上界の評価の結果，下界に近い計算量を実現する回路を示すことができた．

Computational Complexity of Addition and Multiplication on a Circuit Model Considering Wire Delay

Tetsuya NAGASE†, Kazuyoshi TAKAGI†, and Naofumi TAKAGI†

† Graduate School of Information Science, Nagoya University

Abstract For evaluating the performance of an algorithm to solve a problem, it is important to clarify the amount of the computational resource required to solve the problem. In previous studies of the complexity of integrated circuits, wire delay is not considered in the evaluation of computation time. However, with the recent process scaling of integrated circuits, wire delay become significant relative to delay of logic elements. Therefore, it is necessary to discuss the complexity of integrated circuits on a circuit model with wire delay. In this report, we show the complexity of addition and multiplication on a circuit model with wire delay. As a result of evaluation of the lower bound of computational complexity, we found that computational complexity of addition is large on a model with a large influence of wire delay, while computational complexity of multiplication remain unchanged. As a result of evaluation of the upper bound, we could show a circuit that achieves near the lower bound of the computational complexity.

1. はじめに

集積回路上での専用回路や演算回路の設計において，優れた回路を設計するためには，目的の処理を実現するための最適なアルゴリズムの設計が重要となる．ある問題を解くアルゴリズムの設計，選択において，アルゴリズムの性能を評価するためには，その問題を解くために本質的に必要となる計算資源の量，すなわち問題の計算複雑さを知ることが重要となる．集積回路上での計算の複雑さを解析するためには，現実の回路の性質に基づいて回路を数学的にモデル化し，その回路モデル上で，対象の問題や，それを解くアルゴリズムに必要な計算資源の量，すなわち計算量を評価する．

近年の集積回路の微細化の進展により，論理素子の遅延に対して配線遅延が相対的に大きくなってきており，集積回路設計において，配線遅延の考慮は重要になってきている．しかし，従来の集積回路上での計算複雑さの研究では，回路の計算時間の評価において配線遅延を考慮しない回路モデルを用いるものが殆どである [1]～[4].

従って，集積回路上での計算複雑さについても，配線遅延を考慮した回路モデルを定義し，その上で議論する必要があると考えられる．従来研究において計算複雑さの議論に配線遅延の概念を導入した例として，[5] では，配線長に比例する配線遅延を仮定した回路モデル上での様々な問題に対する計算量の下界を示している．本研究では，より現実に即したモデルとして，配線の物理的な性質に基づき，配線遅延の評価を行う回路モデルを考える．本稿では，加算及び乗算について，配線遅延を考慮した回路モデルの上での計算量の下界及び上界を示す．

2. 準　　備

2.1 従来の回路モデル

回路の計算複雑さを評価するために，回路を数学的にモデル化する．本稿では，[1] などで使用された VLSI 回路モデルを考える．この回路モデルの仮定を以下に示す．

A1. 回路は平面上の凸領域にレイアウトされる．

A2. 回路内と回路外とのデータの通信を行う入出力ポートは回路の周上に存在する．

A3. 配線幅及び配線間隔の最小値は $\lambda > 0$ である．λ はテクノロジに依存する定数である．

A4. 配線層数は $\nu \geqq 2$ であり，高々 ν 本の配線が任意の点で交差できる．

A5. 論理素子及び入出力ポートは一定の面積 ($\geqq \lambda^2$) を占有する．

A6. 配線を通って情報が伝達するために必要な最小時間は $\tau > 0$ である．配線遅延は長さに依存しない．

A7. 入力ビットが有効となる時間と場所は入力の値に依存しない．

A8. 1 ビットの情報を保持するために一定の面積 ($\geqq \lambda^2$) を占有する．

A9. 各入力ビットは 1 度だけ使用可能である．回路の外にメモリはないものと仮定する．

A10. 論理素子のファンインを定数に制限する．ファンアウトは制限しない．

2.2 計算複雑さの評価方法

集積回路上での計算複雑さを評価するために，回路の面積 A や計算時間 T などの計算量が用いられる．また，面積と計算時間のトレードオフを見積もるために，AT や AT^2 などの計算量が用いられる．本稿では，問題の計算複雑さの評価に，これらの計算量を使用する．また，2.1 節の仮定により，回路の面積 A と計算時間 T を以下のように定義する．

定義 1. 回路の面積 A は，計算に使用する論理素子，配線及び余白の領域からなる，仮定 **A1** の領域の面積である．

定義 2. 回路の計算時間 T は，ある入力ポートに最初の入力ビットが出現してから，最後の出力ビットが出力ポートに到達するまでの時間である．

表 1　従来研究における加算及び乗算の計算複雑さ

		AT	AT^2
加算	下界 [6]	$\Omega(n)$	$\Omega(n \log^2 n)$
	上界 [1]	$O(n)$	$O(n \log^2 n)$
乗算	下界 [1]	$\Omega(n^{3/2})$	$\Omega(n^2)$
	上界 [2]	$O(n^{3/2})$	$O(n^2)$

面積や計算時間を評価する上で以下で定義されるパラメータを使用する．

定義 3. 回路の素子数とは，回路全体の論理素子の個数である．

定義 4. 回路の段数とは，入力から出力への経路のうち，最多の素子を通る経路の素子数である．2.1 節の回路モデルでは，組合せ回路の計算時間は回路の段数に比例する．

2.3 関 連 研 究

VLSI 回路モデル上での計算複雑さについての従来研究について，加算及び乗算に関する結果を表 1 に示す．

また，配線遅延を考慮した回路モデルについても一部研究されており，[5] では，配線長に比例する配線遅延を考慮した回路モデル上で以下が示されている．

- 加算の計算量の下界 [5]

$AT = \Omega(n), \qquad AT^2 = \Omega(n^2)$

- 乗算の計算量の下界 [5]

$AT = \Omega(n^{3/2}), \qquad AT^2 = \Omega(n^2)$

3. 配線遅延を考慮した回路モデル

3.1 回路モデル

集積回路の微細化により，論理素子の遅延時間はスケーリングに対して減少しているのに対し，演算器内部で使用されるローカル配線ではスケーリングに対して一定となり，長距離配線で使用されるグローバル配線ではスケーリングに対して遅延時間が増大する．このようなことから，論理素子の遅延に対して配線遅延が支配的になりつつある [7]．従って，配線遅延を考慮した回路モデル上での計算複雑さの議論を行うことが重要であると考えられる．本稿では，配線長に依存した配線遅延を仮定して，2.1 節で述べた従来の回路モデルに配線遅延の概念を導入するために，仮定 **A6** の代わりに，以下の仮定を導入する．

A6'. 論理素子間の長さ l の配線には配線遅延 $d_w(l)$ が存在する．

上界の議論においては，回路の配線遅延を，回路の入力から出力までの経路のうち，論理素子間の合計配線遅延の最大値とする．従って，回路全体の配線遅延 D_w は以下の式により計算される．

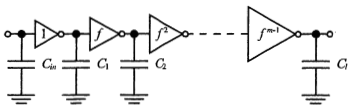

図 1　m 段のバッファの挿入

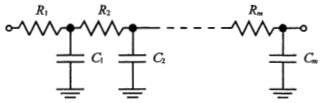

図 2　RC 配線モデル

$$D_w = \max_{p \in P} \left(\sum_{k=1}^{K_p} d_w\left(l_{p_k}\right) \right)$$

P：回路の入力から出力に至る経路の集合

K_p：経路 $p \in P$ における回路の段数

l_{p_k}：経路上の各段の論理素子間の配線長

$d_w(l)$：長さ l の配線に対する配線遅延

一方下界の議論においては，回路の入力から出力までの経路の最大長を l，及びその経路の論理素子の段数を K とすると，合計の配線遅延が $\Omega(K + d_w(l))$ であるとして計算時間を評価する．

3.2　配線遅延モデル

配線遅延 $d_w(l)$ の値を設定するために，配線の物理的な性質を考慮する．配線の物理的なモデルとして，配線遅延を配線容量 C のみでモデル化する方法と，抵抗 R と容量 C でモデル化する方法について述べる．

3.2.1　配線容量によるモデル化

配線遅延について，配線容量による影響が支配的で，配線抵抗による影響が無視できるモデルを考える．配線容量 C_l が配線長 l に比例するため，このモデルでは，配線遅延は配線長に比例する．これに対し，バッファ挿入により配線遅延を改善できる [3]．図 1 のように，バッファを m 個挿入し，各段で容量が f 倍になるようにバッファサイズを定める．i 段目のバッファの遅延は $C_{i+1}/C_i = f \cdot t_0$ となる．t_0 は最小サイズのインバータの遅延時間である．従って，入力容量 C_{in}，配線容量 C_l に対して，$C_l/C_{in} = f^m \cdot t_0$ となる．配線遅延は $m \cdot f \cdot t_0$ となる．最適な f は自然対数の底 e であり，遅延時間は $e \cdot \log(C_l/C_{in}) \cdot t_0 = c_C \cdot \log l$ (c_Cは定数) となる．

3.2.2　配線抵抗及び配線容量によるモデル化

配線が配線抵抗 R と配線容量 C でモデル化されるとし，配線遅延の評価を Elmore 遅延モデル [8] により行う．この RC モデルにおいて配線が図 2 のようにモデル化される場合，Elmore 遅延モデルに基づく配線遅延は $R_1 \cdot C_1 + (R_1 + R_2) \cdot C_2 + \cdots + (R_1 + \cdots + R_m) \cdot C_m$ となる．$R_i = R, C_i = C$ $(1 \leqq i \leqq m)$ と仮定すると，遅延は $R \cdot C \cdot m \cdot (m+1)/2$ となり，m^2 に比例する．単位長さ当たりの抵抗を R_{unit}，単位長さ当たりの容量を C_{unit} とすると，長さ l の配線では遅延が $R_{\text{unit}} \cdot C_{\text{unit}} \cdot l^2$ に比例する．従って，配線遅延は配線長の 2 乗に比例する．これに対し，バッファ挿入により配線遅延を改善できる．長さ l の配線に $w = l/l_0$ 個のバッファを挿入すると（l_0 は定数），w 本の長さ l_0 の配線に分割される．長さ l_0 の配線の遅延は定数であり，d_0 とすると，全体の配線遅延は $d_0 \cdot w = (d_0/l_0) \cdot l = c_{RC} \cdot l$ （c_{RC} は定数）となる．

計算量の評価では，回路に適切なバッファの挿入がなされるものと考える．従って，配線遅延の評価においては，バッファ挿入後の配線遅延である遅延モデルを使用する．配線遅延が配線長 l に対して $\log l \sim l$ 程度になると考えられるため，本稿では，以下の 3 つの配線遅延 $d_w(l)$ を考える．なお，配線遅延 $d_w(l) = c$ モデルは，従来の配線遅延を考慮しないモデルである．

$$d_w(l) = \begin{cases} c & \\ c \cdot \log^{\alpha} l & (\alpha \geqq 1) \\ c \cdot l^{\beta} & (0 < \beta \leqq 1) \end{cases} \tag{1}$$

長さ l の配線遅延が l よりも大きな場合も考えられるが，バッファ挿入により，長さに比例する配線遅延モデルに帰着することができるため，本稿では考慮しない．

4.　加算の計算複雑さ

4.1　加算の計算量の下界

本節では，配線遅延を考慮した回路モデル上での加算の計算量の下界について，次の定理を示す．

定理 1. 加算の計算量の下界について，以下が成立する．

$$AT = \Omega(n) \tag{2}$$

$$AT^2 = \begin{cases} \Omega(n \log^2 n) & (d_w(l) = c) \\ \Omega(n \log^{2\alpha} n) & (d_w(l) = c \cdot \log^{\alpha} l) \\ \Omega(n^{1+2\beta/(\beta+1)}) & (d_w(l) = c \cdot l^{\beta}) \end{cases} \tag{3}$$

[5] において，定理 1 の $d_w(l) = c \cdot l$ の場合に関しての証明を行っている．本節では，[5] の証明と同様の方法を用いて定理 1 を証明する．

n ビット整数加算において，Y を $[11 \cdots 1]$ とする．この計算を行う回路の出力は，回路の全ての入力ビットに依存する．入力 X が m 回に分けて入力されるとする．時刻 t_j $(1 \leqq j \leqq m,\ t_j < t_{j+1})$ に入力されたビットの集合を Z_j とする．$\sum_{j=1}^{m} |Z_j| = n$ である．$X_i = [x_i \cdots x_0]$ とし，X_i に含まれるビットが最後に回路に入力される時刻を t_{f_i} とする．$1 \leqq j \leqq f_i$ である j に対して，$L_j^i = |X_i \cap Z_j|$ を，時刻 t_j に入力され，$\{x_i, \cdots, x_0\}$ に含まれるビット数と定義する．このとき，$L_1^i + \cdots + L_{f_i}^i = |X_i| = i + 1$ である．以下の補題を証明する．

補題 1. [5]　周の長さが p であるような任意の凸多角形を P とする．このとき，P の各頂点から P 内部の任意の点までの最長距離は $\Omega(p)$ となる．

証明 (補題 1). P の各頂点から P 内部の任意の点までの最長距離が最短となるのは，P が直径 p/π の円の場合である．円周上のある点から，円内部の任意の点までの最長距離は直径である p/π に等しく，$\Omega(p)$ である． □

関数 T_x を以下のように定義する．

$$T_x(l) = \begin{cases} \log l & (d_w(l) = c) \\ \log^\alpha l & (d_w(l) = c \cdot \log^\alpha l) \\ l^\beta & (d_w(l) = c \cdot l^\beta) \end{cases}$$

s_i を計算するために必要な時間の下界を求める．

補題 2. $0 \leqq i < n$ である任意の i に対して，s_i は時刻 $t = \max\{t_1 + T_x(L_1^i), \cdots, t_{f_i} + T_x(L_{f_i}^i)\}$ より早く計算することはできない．

証明 (補題 2). s_i は $|X_i| = L_1^i + \cdots + L_{f_i}^i = i+1$ ビットに依存する．任意の j $(1 \leqq j \leqq f_i)$ に対して，各時刻 t_j に入力される L_j^i ビットの計算に，L_j^i 変数の論理関数の計算が必要となる．仮定 **A10** より，回路の段数は少なくとも $\log L_j^i$ となり，回路の L_j^i 個の全ての入力ポートについて関数値が出力される出力ポートへのパスが存在するため，仮定 **A2** 及び補題 1 により，このパスの最長距離が少なくとも L_j^i となる．従って，時刻 t_j^i に L_j^i ビット入力され，L_j^i 変数の論理関数を計算する回路の計算時間 $d_w(L_j^i) + c_{i,j} \log L_j^i \geqq T_x(L_j^i)$ ($c_{i,j}$は定数) となる．以上より，s_i は任意の j に対して，時刻 $t_j + T_x(L_j^i)$ より早く計算することはできない． □

時刻 t に既に入力された X のビット数を $X(t)$，時刻 t までに出力された S のビット数を $S(t)$ とする．定義により，$X(t_j) = |Z_1| + \cdots + |Z_j|$ である．関数 $S(t)$ の増加を評価するために，上述の結果を使用する．

補題 3. $1 \leqq k \leqq m$ 及び $t_k \leqq t < t_{k+1}$ に対して，次式が成り立つ．ただし，T_x^{-1} は関数 T_x の逆関数である．

$$S(t) \leqq \sum_{j=1}^{k} \min\left(|Z_j|, T_x^{-1}(t - t_j)\right)$$

証明 (補題 3). 時刻 t までに出力されたビットのうち，最上位ビットを s_i とすると，次式が成り立つ．

$$i + 1 \geqq S(t) \tag{4}$$

L_j^i の定義から，$L_j^i \leqq |Z_j|$ が成り立つ．また，補題 2 から，$1 \leqq j \leqq f_i$ に対して $t \geqq t_j + T_x(L_j^i)$ である．従って，関数 T_x の逆関数 T_x^{-1} を使って，$L_j^i \leqq T_x^{-1}(t - t_j)$ が成り立つ．従って，L_j^i の定義から以下が成り立つ．

$$i + 1 = \sum_{j=1}^{f_i} L_j^i \leqq \sum_{j=1}^{f_i} \min\left(|Z_j|, T^{-1}(t - t_j)\right) \tag{5}$$

時刻 t において，$x_1, \ldots, x_i$ は既に入力されているため，$t_{f_i} \leqq t$ である．また，$t_k \leqq t < t_{k+1}$ であることから，時刻 t_k に回路にビットが入力された後，時刻 t までに新たなビットが入力されていない．従って，$t_{f_i} \leqq t_k \leqq t$ が成り立ち，式 (4) 及び式 (5) より，補題が証明された． □

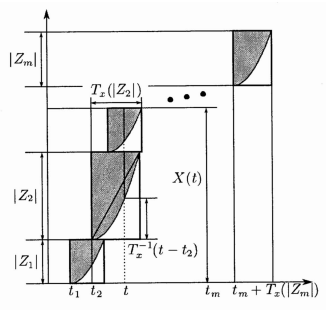

図 3　補題 3 の幾何学的な説明

以上の補題を用いて，定理 1 を証明する．

証明 (定理 1). 図 3 は補題 3 を幾何学的に説明したものである．このとき，$S(t)$ は点線の長さの合計で抑えられる．図において，m 個の網掛け部分の面積の合計は，$\int_{t_1}^{t_m + T_x(|Z_m|)} [X(t) - S(t)]dt$ となり，図中の底辺 $T_x^{-1}(|Z_j|)$，高さ $|Z_j|$ の三角形よりも大きく，その倍の大きさの長方形よりも小さい．また，補題 2 より，n ビットの加算に必要な計算時間 T は，少なくとも $t_m + T_x(|Z_m|)$ である．従って，$t_1 = 0$ とすると，以下が得られる．

$$\int_0^T [X(t) - S(t)]dt \geqq \frac{1}{2} \sum_{j=1}^{m} |Z_j| \cdot T_x(|Z_j|)$$

右辺が最小となるのは全ての $|Z_j|$ が等しいときであり，$\sum_j |Z_j| = n$ であるため $|Z_j| = n/m$ である．また，仮定 **A8** から，任意の t に対して回路中に $X(t) - S(t)$ ビット保持しておかなければならない．従って $A \geqq X(t) - S(t)$ であり，以下が成り立つ．

$$\begin{aligned} AT &\geqq \int_0^T [X(t) - S(t)]dt \geqq \frac{m}{2} \cdot \frac{n}{m} \cdot T_x\left(\frac{n}{m}\right) \\ &= \frac{n}{2} \cdot T_x\left(\frac{n}{m}\right) = \Omega(n) \end{aligned} \tag{6}$$

また，計算時間について，以下が成り立つ．

$$\begin{aligned} T &\geqq t_m + T_x(|Z_m|) \geqq m + T_x(n/m) \\ &= \begin{cases} \Omega(\log n) & (d_w(l) = c) \\ \Omega(\log^\alpha n) & (d_w(l) = c \cdot \log^\alpha l) \\ \Omega(n^{\beta/(\beta+1)}) & (d_w(l) = c \cdot l^\beta) \end{cases} \end{aligned} \tag{7}$$

式 (6) 及び式 (7) より，AT^2 について以下が成り立つ．

$$\begin{aligned} AT^2 &= \Omega(n \cdot T_x(u)(n/u + T_x(u))) \\ &= \begin{cases} \Omega(n \log^2 n) & (d_w(l) = c) \\ \Omega(n \log^{2\alpha} n) & (d_w(l) = c \cdot \log^\alpha l) \\ \Omega(n^{1+2\beta/(\beta+1)}) & (d_w(l) = c \cdot l^\beta) \end{cases} \end{aligned}$$

従って，式 (3) が成り立ち，定理 1 が証明された． □

以上より，式 (1) の任意の配線遅延モデルに対して，$AT = \Omega(n)$ を得た．AT^2 計算量は，配線遅延の大きなモデルほど大きいという結果を得た．配線遅延 c モデルでは [6] と，配線遅延 $c \cdot l$ モデルでは [5] と同じ結果を得た．

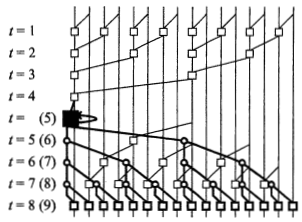

図 4　Brent-Kung 加算器

4.2　加算の計算量の上界

本節では [1] の加算器を示すことにより，配線遅延を考慮した回路モデル上での加算の計算量の上界を示す．

4.2.1　Brent-Kung 加算器 [1]

[1] の加算器は，パイプライン式のプレフィクス加算器であり，n ビットの被加数及び加数を，連続する w ビットごとに n/w 個に分割し，w ビットの加算器を使用して計算を行う．回路は図 4 のようになる．図の□ではプレフィクス計算を行い，○は分岐を示す．■ではプレフィクス計算を行うが，パイプラインの i 段目のプレフィクス計算の結果を保持し，$i+1$ 段目のプレフィクス計算の一方の入力として使用する．■の回路を通るデータは図 4 の括弧内の時刻にそれぞれのモジュールに到達する．

w ビットの加算器 $\mathrm{BK}(w)$ の面積は $O(w\log w)$ であり，段数は $O(\log w)$ となる．配線遅延を考慮した評価を行うために，次の 2 つのパイプラインの構成方法を考える．

(i)　パイプライン 1 段で通信可能な配線長を定数に制限する．回路の段数は $O(w)$，面積は $A=O(w^2)$，計算時間は $T=O(n/w+w)$ となる．

(ii)　パイプラインの周期を，論理素子間の最大配線遅延である $O(d_w(w))$ とする．面積は $A=O(w\log w)$，計算時間は $T=O(d_w(w)(n/w+\log w))$ となる．

(i)，(ii) の場合について AT，AT^2 計算量を評価する．

(i) の場合について，以下が成り立つ．

$AT=O(n)$　　($w=$ const のとき)

$AT^2=O(n^2)$　　($w=\sqrt{n}$のとき)

(ii) の場合について，以下が成り立つ．

$AT=O(n)$　　($w=$ const のとき)

配線遅延 $d_w(l)=c$ モデルにおいて

$AT^2=O(n\log^2 n)$　　($w=n/\log n$ のとき)

配線遅延 $d_w(l)=c\cdot\log^\alpha l$ モデルにおいて

$AT^2=O(n\log^{2+2\alpha} n)$　　($w=n/\log^{1+\alpha} n$ のとき)

配線遅延 $d_w(l)=c\cdot l^\beta$ モデルにおいて

$AT^2=O(n^{1+2\beta}\log^{2-2\beta} n)$　　($w=n^{1/(\beta+1)}$のとき)

(i)，(ii) の場合をまとめると，以下のようになる．

$$AT = O(n)$$

$$AT^2 = \begin{cases} O(n\log^2 n) & (d_w(l)=c) \\ O(n\log^{2+2\alpha} n) & (d_w(l)=c\cdot\log^\alpha l) \\ O(n^{1+2\beta}\log^{2-2\beta} n) & (d_w(l)=c\cdot l^\beta, \beta<1/2) \\ O(n^2) & (d_w(l)=c\cdot l^\beta, \beta\geqq 1/2) \end{cases}$$

AT は，式 (1) の任意の配線遅延モデルにおける下界と一致する．AT^2 は，配線遅延 $d_w(l)=c$ モデルでは，4.1 節及び [6] の下界と一致する．配線遅延 $d_w(l)=c\cdot l$ モデルの場合は，[5] 及び 4.1 節で示した下界と一致する．

下界に近い構成を得るために以下の (iii) のような理想的なパイプラインを仮定する．

(iii)　パイプラインの周期を $O(1)$，回路を組合せ回路と考えたときの遅延時間を $T_a(w)=O(\sum_{i=1}^{\log w} d_w(2^i))$ とする．回路に入力されたデータが，$T_a(w)$ 時間後に出力されると仮定する．回路は $O(w\cdot T_a(w))$ ビットのデータを保持する必要があるため，面積は $A=O(w\cdot T_a(w))$ となる．計算時間は $T=O(n/w+T_a(w))$ となる．

$T_a(w)$ は以下のようになる．

$$T_a(w) = \begin{cases} O(\log w) & (d_w(l)=c) \\ O(\log^{1+\alpha} w) & (d_w(l)=c\cdot\log^{1+\alpha} l) \\ O(w^\beta) & (d_w(l)=c\cdot l^\beta) \end{cases}$$

従って，(iii) の場合について，以下が成り立つ．

$AT=O(n)$　　($w=$ const のとき)

配線遅延 $d_w(l)=c$ モデルにおいて

$AT^2=O(n\log^2 n)$　　($w=n/\log n$ のとき)

配線遅延 $d_w(l)=c\cdot\log^\alpha l$ モデルにおいて

$AT^2=O(n\log^{2+2\alpha} n)$　　($w=n/\log^{1+\alpha} n$ のとき)

配線遅延 $d_w(l)=c\cdot l^\beta$ モデルにおいて

$AT^2=O(n^{1+2\beta/(1+\beta)})$　　($w=n^{1/(\beta+1)}$のとき)

以上より，(iii) の状況を仮定すると配線遅延 $d_w(l)=c$，$d_w(l)=c\cdot l^\beta$ モデルでは計算量の下界と上界が一致した．

4.3　考　　察

配線遅延 $d_w(l)=c\cdot\log^\alpha l$ モデルにおいて，AT^2 計算量の下界と上界が一致しない．これは，配線遅延 $d_w(l)=c\cdot\log^\alpha l$ モデル上での n 変数の論理関数の計算時間の下界と上界のギャップによると考えられる．回路の n 個の入力ポートから関数値が出力される出力ポートへのパスが存在するため，仮定 **A2** 及び補題 1 より，回路の入出力間の最大経路長は $\Omega(n)$，仮定 **A10** より回路の段数が $\Omega(\log n)$ であるから，計算時間が $\Omega(\log^\alpha n)$ となる．一方，4.2.1 項の w ビット加算器の構成と同様に 2 分木状に回路を構成することを考えると，k 段目と $k+1$ 段目の分岐の間の配線長が $O(2^k)$ となり，計算時間が $O(\log^{1+\alpha} n)$ となる．従って，計算時間の下界と上

界が一致しない．配線遅延 $d_w(l) = c \cdot \log^{\alpha} l$ モデル上での n 変数の論理関数の計算時間について下界と上界が一致することが明らかとなれば，加算の計算時間についても，下界と上界が一致する構成が得られると考えられる．

配線遅延 $d_w(l) = c \cdot l^{\beta}$ モデルにおいて，(i)，(ii) パイプラインの構成では AT^2 計算量の上界と下界が一致せず，理想的なパイプラインを仮定した (iii) の場合では，AT^2 計算量の下界と上界が一致した．従って，この理想的なパイプラインを実現する回路により，これらの計算量の下界を実現できると考えられる．

5. 乗算の計算複雑さ

5.1 乗算の計算量の下界

本節では，配線遅延を考慮した回路モデル上での乗算の計算量の下界について，次の定理を示す．

定理 2. 乗算の計算量の下界について，以下が成立する．

$$AT = \Omega(n^{3/2}), \quad AT^2 = \Omega(n^2) \tag{8}$$

定理 2 を証明するにあたり，[4] で定義されている transitive function の計算複雑さを考える．

定義 5. [4] ブール関数 $f(x_0, \ldots, x_{N-1}, s_0, \ldots, s_{b-1}) = (z_0, \ldots, z_{N-1})$ が N 要素の推移的置換群を計算する，すなわち，全ての $0 \leqq i, j < N$ の添え字の組 (i, j) に対して，x_i を z_j に割り当てる置換を計算するならば，f は N 次の transitive function である．

つまり，変数 $s_0, \ldots, s_{b-1}$ に特別な値を割り当てることにより，出力が単に入力の置換となるならば，関数 f は N 次の transitive function である．また，f の任意の入力ビット x_i が任意の出力位置 z_j に出現する可能性がある．以下で定義される N ビット巡回シフトは出力が入力の置換となり，各入力ビットが $2^b = N$ 個の出力位置に出現する．従って，N 次の transitive function である．

入力 : $X = [x_{N-1} \cdots x_0], S = [s_{b-1} \cdots s_0]$ $(b = \lceil \log N \rceil)$

出力 : $Z = [z_{N-1} \cdots z_0]$

$$z_i = x_{(i-S) \bmod N}$$

乗算と transitive function の計算複雑さの関係を示すために以下の補題を示す．

補題 4. n ビット × n ビットの乗算により，$n/2$ ビットの巡回シフトの計算が可能である．

証明 (補題 4). 簡単のため n は偶数であるとする．n ビット × n ビットの乗算において，被乗数 X の下位 $n/2$ ビットを $X' = [x'_{n/2-1} \cdots x'_0]$ とし，上位 $n/2$ ビットは 0 とする．乗数 Y は上位 $n/2$ ビットと下位 $n/2$ ビットを，ともに 2^i $(0 \leqq i < n/2)$ とする．このとき，積 P の下位から $n/2+1$ ビット目から n ビット目が $[x'_{n/2-(i+1)} \cdots x'_0\ x'_{n/2-1} \cdots x'_{n/2-i}]$ となり，X' の巡回シフトになっている． □

補題 4 により，n ビット × n ビットの乗算の計算複雑さは $n/2$ 次の transitive function 以上である．従って，$N = n/2$ 次の transitive function の計算量の下界は n ビット × n ビットの乗算の計算量の下界となる．

N 次の推移的関数について，従来研究により既に証明されている補題を用いて定理 2 の証明を行う．以下の補題で使用するパラメータとして，回路のデータレート D 及び周期 P を定義する．パイプライン化された回路により，対象の問題を繰り返し計算することを考える．このとき，連続する 2 つの計算の開始時刻の差の最小値を周期 P とし，単位時間当たりに処理可能な最大ビット数をデータレート D と呼ぶ．$D = N/P$ が成り立つ．[4] では，transitive function を計算する回路について，面積 A とデータレート D の関係として，以下が示されている．

補題 5. [4] N 次の transitive function を計算する回路のデータレートを D とすると，以下が成り立つ．

$$A = \Omega(N + D^2)$$

証明 (補題 5). transitive function では，回路の最後の入力ビットが，最初に出力されるように入力パラメータを設定することが出来る．従って，仮定 **A8** より，回路中に N ビットの情報を記憶するために $\Omega(N)$ の面積が必要となる．従って $A = \Omega(N)$．

関数 $f(x_0, \ldots, x_{N-1}) = (x_{g(0)}, \ldots, x_{g(N-1)})$ が，関数 g で指定される置換を計算する transitive function であると仮定する．このとき，面積 $A = a \times b$ $(a \geqq b)$ の回路において，回路の出力ビットの数がほぼ同じになるように，領域を L, R に分割する直線 C を考えると，関数 f の計算に $\Omega(N)$ のデータが直線 C を横切る必要があることが示せる．C を横切る配線の本数を ω とすると，$b \geqq \omega$ より，$A = a \times b \geqq \omega^2$ である．このとき，データの移動時間は $\Omega(N/\omega)$ であるが，回路の周期がこれより大きくなければならないことから，$N/\omega \leqq P = N/D$，従って，$\omega = \Omega(D)$ である．従って，$A \geqq \omega^2 = \Omega(D^2)$． □

N 次の transitive function を計算する回路の計算時間に関して，[5] の証明の拡張により以下が得られる．

補題 6. N 次の transitive function を計算する回路をのデータレートを D とすると，以下が成り立つ．

$$T = \Omega(N/D + d_w(D) + \log D)$$

証明 (補題 6). 回路のデータレートを D とすると，入出力ポート数は少なくとも D である．回路は transitive function を計算するため，入力ポートから，回路の境界上にある D 個の出力ポートへのデータパスが存在する．補題 1 よりこのデータパスの最大長は $\Omega(D)$ である．逆に，任意の出力ポートに対して D 個の入力ポートからのデータパスが存在する．従って，任意の出力の計算に D 変数の論理関数を計算が必要となり，段数が $\Omega(\log D)$ となる．また，ポート数が D であることから，

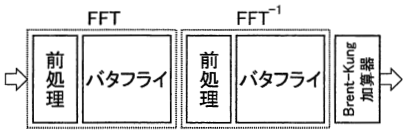

図 5　FFT 乗算器の構成

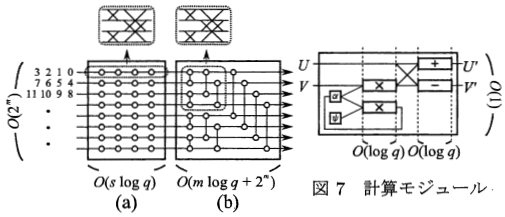

図 6　前処理部，バタフライ部　　図 7　計算モジュール

データの出力に時間 $\Omega(N/D)$ が必要となる．以上より $T = \Omega(N/D + d_w(D) + \log D)$ となる．　□

証明 (定理 2). 補題 5 及び補題 6 より，N 次の transitive function の AT 及び AT^2 について，以下が成り立つ．

$$AT = \Omega(N^{3/2}), \quad AT^2 = \Omega(N^2)$$

$N = n/2$ であることから，n ビット × n ビットの乗算について，式 (8) が成り立つ．　□

以上より，式 (1) の任意の配線遅延モデルにおいて，AT 及び AT^2 計算量の下界が同じという結果を得た．

5.2　乗算の計算量の上界

本節では [2] の乗算器を示すことにより，配線遅延を考慮した回路モデル上での加算の計算量の上界を示す．

5.2.1　FFT 乗算回路 [2]

[2] の乗算器は，FFT を用いて乗算の計算を行う FFT 乗算器である．全体の構成は図 5 のようになる．FFT の逆計算は，FFT と同様の回路となる．最後の Brent-Kung 加算器には，4.2.1 項で示した加算器を使用する．加算器部分は FFT 部分よりも，面積，計算時間ともに小さくなるため，ここでは FFT 部分のみ説明する．

n ビット × n ビットの乗算を行うために，被乗数及び乗数の各 n ビットを $\log n$ ビットの $q/2$ 系列に分割する．すなわち，$q/2 = n/\log n$ である．このような各項 $\log n \simeq \log q$ ビットの q 項（上位 $q/2$ 項は全て 0）に対して FFT を計算する．このとき，$q = s \cdot 2^m$ として，2^m 項を並列に，s 段のパイプラインによって計算する．また，$s \geqq m$ であるとする．

[2] の FFT 乗算器の各部の詳細について説明する．

- バタフライ部

図 5 のバタフライ部の回路の構造をグラフとして表したものを図 6(b) に示す．回路の入力は左側から入り，右方向に処理が進む．垂直方向の辺が存在するノード間では，図 7 のような回路で以下の演算が行われる．

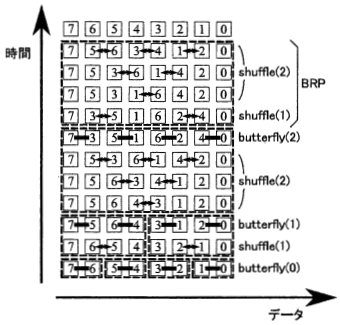

図 8　前処理部の計算

1. $\hat{V} \leftarrow \alpha \cdot V$
2. $(U', V') \leftarrow (U + \hat{V}, U - \hat{V})$
3. $\alpha \leftarrow \alpha \cdot \psi$

この演算は，FFT におけるバタフライ演算になっている．図 7 の計算モジュールでの計算は，高さ $O(1)$，幅 $O(\log q)$ の演算器により，計算時間 $O(\log q)$ で実行できる．垂直方向の配線長が 2^i のノードを次元 i のノードと呼ぶことにする．次元 i のノード間には，長さ 2^i の配線が存在する．この配線による領域を考慮すると，配線部分の幅が次元 i に対して 2^i となる．従って，図 6(b) のバタフライ部は高さ $O(2^m)$，幅 $O(m \log q + 2^m)$ であり，計算モジュールによる計算時間は $O(s \log q)$ である．

- 前処理部

図 5 の前処理部の回路の構造を図 6(a) に示す．前処理部は，$s = 2^r$ 個のモジュールから成る，2^m 本の一次元配列である．各配列は，s 項のデータに対し，perfect-shuffle を使用してバタフライ部で行う演算を模倣する．$r = 3$ のときの回路の動作を図 8 に示す．図中の butterfly(i) は，バタフライ部における i 次元目のバタフライ演算に相当し，太線部分で図 7 による計算を行う．shuffle(i) は，2^{i+1} 項に対する perfect-shuffle に相当し，矢印部分で隣接するノードのデータの交換する．また，BRP (bit-reversal-permutation) では，perfect-shuffle によって入れ替わったデータを元の順番に戻す．一つの項に対する butterfly は r 回実行される．また，図 8 の 1 行分の計算を 1 ステップの計算と呼ぶことにすると，shuffle(i) の計算は $2^i - 1$ ステップである．従って，前処理のステップ数は $O(s)$ になる．butterfly(i) の計算時間が $\log q$ であることから，1 ステップの計算時間を $\log q$ とすると，$s \geqq m$ より，全体の計算時間は $O(s \log q)$ となる．図 6(b) の前処理部は高さ $O(2^m)$，幅 $O(s \log q)$ である．

$s \geqq m$ により，FFT 回路の面積は $A = O(2^m \cdot (s \cdot \log q + 2^m))$，計算モジュールによる計算時間は $T = O(s \log q)$ となる．この FFT 乗算器において，配線遅延の影響が考

えられるのは，次元 i のノード間の配線の長さが $O(2^i)$ となるバタフライ部である．配線遅延を考慮した評価を行うために，次の 2 つパイプラインの構成方法を考える．

(i) パイプライン 1 段で通信可能な配線の長さを定数に制限する．回路の段数は $O(2^m)$，パイプライン 1 段の計算時間は $O(\log q)$ であり，$s \geqq m$ より全体の計算時間は $T = O((s+2^m)\cdot \log q)$ となる．

(ii) パイプラインの周期を，モジュールの計算時間と論理素子間の最大配線遅延である $O(\log q + d_w(2^m))$ とする．$s \geqq m$ より全体の計算時間は $T = O(s\cdot(\log q + d_w(2^m)))$ となる．

(i)，(ii) の場合について AT，AT^2 計算量を評価する．$q = s\cdot 2^m$ より，$2^m = q/s$，$m = O(\log q)$ であり，$s \geqq m$ より $s \geqq \Omega(\log q)$ となることを利用する．

(i) の場合について，以下が成り立つ．

$$AT = O(q^{3/2}\log^2 q) = O(n^{3/2}\sqrt{\log n}) \quad (s = \sqrt{q} \text{のとき})$$
$$AT^2 = O(q^2\log^3 q) = O(n^2 \log n) \quad (s = \sqrt{q} \text{のとき})$$

(ii) の場合について，以下が成り立つ．

配線遅延 $d_w(l) = c$ モデルにおいて

$$AT = O(n^{3/2}) \quad (s = \sqrt{q/\log q} \text{のとき})$$
$$AT^2 = O(n^2) \quad (s = \sqrt{q/\log q} \text{のとき})$$

配線遅延 $d_w(l) = c\cdot \log^\alpha l$ モデルにおいて

$$AT = O(n^{3/2}\log^{\alpha-1} n) \quad (s = \sqrt{q/\log q} \text{のとき})$$
$$AT^2 = O(n^2 \log^{2\alpha-2} n) \quad (s = \sqrt{q/\log q} \text{のとき})$$

配線遅延 $d_w(l) = c\cdot l^\beta$ モデルにおいて

$$AT = O(n^{(3+\beta)/2}/\log n) \quad (s = \sqrt{q/\log q} \text{のとき})$$
$$AT^2 = O(n^{2+\beta}/\log^2 n) \quad (s = \sqrt{q/\log q} \text{のとき})$$

(i)，(ii) の場合をまとめると，以下のようになる．

$$AT = \begin{cases} O(n^{3/2}) & (d_w(l) \leqq c\cdot\log l) \\ O(n^{3/2}\sqrt{\log n}) & (d_w(l) > c\cdot\log l) \end{cases}$$

$$AT^2 = \begin{cases} O(n^2) & (d_w(l) \leqq c\cdot\log l) \\ O(n^2\log n) & (d_w(l) > c\cdot\log l) \end{cases}$$

従って，長さ l の配線に対する配線遅延が $d_w(l) = c\cdot\log l$ 以下の場合は，AT 及び AT^2 の上界が下界と一致する．

下界に近い構成を得るために，以下の (iii) のような理想的なパイプラインを仮定する．

(iii) パイプラインの周期を $O(\log q)$ とする．回路を組合せ回路と考えたときの遅延時間を $T_m(2^m) = O((s+m)\log q + \sum_{i=1}^m d_w(2^i))$ とする．このとき，回路に入力されたデータが $T_m(2^m)$ 時刻後に出力されると仮定する．$s \geqq m$ より，回路全体の計算時間は $T = O(s\log q + \sum_{i=1}^m d_w(2^i))$ となる．

$\sum_{i=1}^m d_w(2^i)$ の値は以下のようになる．

$$\sum_{i=1}^m d_w(2^i) = \begin{cases} O(m) & (d_w(l) = c) \\ O(m^{1+\alpha}) & (d_w(l) = c\cdot\log^{1+\alpha} l) \\ O(2^{\beta m}) & (d_w(l) = c\cdot l^\beta) \end{cases}$$

従って，(iii) の場合について，以下が成り立つ．

$$AT = O(n^{3/2}) \quad (s = \sqrt{q/\log q} \text{のとき})$$
$$AT^2 = O(n^2) \quad (s = \sqrt{q/\log q} \text{のとき})$$

以上より，(iii) の状況を仮定すると，AT 及び AT^2 について，5.1 節で示した下界と一致する．

5.3 考　　察

配線遅延 $d_w(l) > c\cdot\log l$ モデルにおいて，(i)，(ii) のパイプライン構成では AT 及び AT^2 計算量の上界と下界が一致せず，理想的なパイプラインを仮定した (iii) の場合では，AT 及び AT^2 計算量の下界と上界が一致した．従って，この理想的なパイプラインを実現する回路が構成できれば，これらの計算量の下界を実現できると考えられる．

6. ま と め

本稿では，配線遅延を考慮した回路モデル上での加算及び乗算の計算複雑さを明らかとした．計算量の下界を評価した結果，配線遅延の影響が大きなモデルほど加算の AT^2 計算量は大きくなったが，乗算の AT 及び AT^2 計算量には変化がなかった．計算量の上界を評価した結果，加算及び乗算について下界に近い計算量を実現する回路を示すことができた．AT や AT^2 計算量で計算複雑さが議論される問題は加算及び乗算の他にも様々存在する．本稿で述べた加算のように，配線遅延を考慮することによって計算量が変化する問題が存在する可能性がある．

文　　献

[1] R. P. Brent and H. T. Kung. The Chip Complexity of Binary Arithmetic. *Proc. 12th Annu. ACM Symp. on Theory of Comput.*, pp. 190–200, 1980.

[2] F. P. Preparata and J. E. Vuillemin. Area Time Optimal VLSI Networks for Computing Integer Multiplication and Discrete Fourier Transform. *LNCS*, Vol. 115, pp. 29–41, 1981.

[3] C. Mead and L. Conway. *Introduction to VLSI Systems*. Addison-Wesley, 1980.

[4] J. Vuillemin. Combinatorial Limit to the Computing Power of VLSI Circuits. *IEEE Trans. Comput.*, Vol. C-32, No. 3, pp. 294–300, Mar. 1983.

[5] B. Chazelle and L. Monier. A Model of Computation for VLSI with Related Complexity Results. *Journal of the ACM*, Vol. 32, No. 3, pp. 573–588, Jul. 1985.

[6] R. B. Johnson Jr. The Complexity of a VLSI Adder. *Inform. Process. Lett.*, Vol. 11, No. 2, pp. 92–93, 1980.

[7] 吉川 公麿. 多層配線技術とスケーリング. 電子情報通信学会論文誌 C, Vol. 83, No. 2, pp. 105–117, 2000.

[8] W. C. Elmore. The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers. *Journal of Applied Physics*, Vol. 19, p. 55, 1947.