CRCW PRAMの時間計算量の稠密な階層

モルシェド, モンズール
九州大学大学院システム情報工学研究科情報工学専攻 : 博士後期課程

岩本, 宙造
九州芸術工科大学

岩間, 一雄
京都大学

https://doi.org/10.15017/1523863

出版情報：九州大学大学院システム情報科学紀要. 2 (2), pp.265-269, 1997-09-26. 九州大学大学院システム情報科学研究科

九州大学大学院
システム情報科学研究科報告
第2巻 第2号 平成9年9月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.2, No.2, September 1997

CRCW PRAMの時間計算量の稠密な階層

モンズール・モルシェド*・岩 本 宙 造**・岩 間 一 雄***

A Hierarchy of Time-Complexities Based on CRCW PRAMs

Manzur MORSHED, Chuzo IWAMOTO and Kazuo IWAMA

(Received June 20, 1997)

Abstract: A hierarchy theorem for parallel time-complexities between constant and $\frac{\log n}{\log\log n}$ is presented. Suppose that $t(n)$ is a function such that there is a TM making $(t(n))^{\Theta(1)}$ moves and that the inverse function $t^{-1}(n)$ is bounded by $O(\frac{\log n}{\log\log n})$. Then, there exists a language L such that $\Theta(t^{-1}(n))$ time is necessary and sufficient for CRCW PRAMs with polynomially many processors to recognize L.

Keywords: Computational complexity, CRCW PRAM, Parallel computation

1. はじめに

並列計算量の時間的階層性に証明を与えることは非常に難しいと見られている．例えば，並列計算における時間量のクラス NC^k をNC^{k+1} から分離する問題は未解決であり，さらには，NC^1 を NP から分離できるか否かさえも分かっていない[6]．しかしながら，「より難しい問題には，より多くの並列計算時間が必要となる」ことは直観的に明らかであり，このことを示唆する如何に強い証拠を見つけるかが計算量理論の分野の重要な課題になっている．

Kirchherr [7] は，プロセッサ数を n^j に制限した CRCW PRAM で$\log^i n$ 時間で受理できる言語のクラスは，プロセッサ数 $n^{96j+104}$ の CRCW PRAM で $\log^{i+8} n$ 時間で受理できる言語のクラスより真に小さいことを証明した．文献3) において，この結果は大幅に改善され，計算時間を定数倍に増やすだけで（プロセッサ数を一切増加させせなくても），CRCW PRAM の能力は真に上昇することが示された．また，同じ文献で，ある定数 c, d に対し，素子数 $Z(n)$, 段数 $T(n)$ の対数時間一様な論理回路族で受理される言語のクラスは，素子数 $(Z(n))^c$, 段数 $dT(n)$ で受理される言語のクラスより真に小さいことが示されている．また，岩本ら[5]は，計算量を厳密に測定できる TM を並列計算用に拡張した並列乱アクセス TM を提案している．並列乱アクセス TM は，PRAM とほぼ同等の能力をもつことが証明され，また，並列乱アクセス TM の計算量の時間的階層性も示されている[5]．これらは，並列計算における時間量の階層性を強く示唆する結果であるが，プロセッサ数や素子数に制限を加えており，「プロセッサや素子を多項式の範囲で幾らでも使っても良い」とする一般的な仮定の下では成立しない．

本稿では，プロセッサ数に関する一般的な仮定の下で，CRCW PRAM が受理する言語のクラスの時間的階層性を示す．$t(n)$ を次の条件を満たす関数とする．(i) 動作数が $(t(n))^{\Theta(1)}$ で押えられる TM が存在する．(ii) $t(n)$ の逆関数が $O\left(\frac{\log n}{\log\log n}\right)$ で押えられる．このとき，多項式の範囲内で多数のプロセッサをもつ如何なるCRCW PRAM でも，受理するのに $\Theta(t^{-1}(n))$ 時間を必要かつ十分とする言語が存在することを示す．

CRCW PRAM の時間量に関する研究として，文献4) は CRCW PRAM には計算時間の一般的下限は存在しないこと，つまり，CRCW PRAM の並列計算時間のクラスには，いくらでも小さいものがありうることを示した．また，文献1),9) は，$\Theta\left(\frac{\log n}{\log\log n}\right)$ が並列計算時間の上下限となる言語を示している．本稿では，定数と $\frac{\log n}{\log\log n}$ の間で，並列時間計算量の稠密な階層性を示す．

計算量の階層性の証明で最も一般的に用いられる手法は対角線論法である．例えば，逐次計算量の時間的階層性は，対角線論法で証明されているが[2]，「$t(n)$ 時間の如何なるTMも模倣できる」というTMを作っているため，模倣のための $\log t(n)$ 時間のオーバーヘッドが必要となり，階層性の稠密さを損なう原因になっている．一方，並列計算量の階層性を対角線論法で証明すると，逐次計算と同様のオーバーヘッドが生じることに加え，少ないプロセッサで多くのプロセッサを模倣することのないように，プロセッサ数を制限せざるを得ない[3),5),7]．しかし，本稿のように，$\frac{\log n}{\log\log n}$ よりも小さい並列時間量を考えた場合は，多項式の範囲で幾らでも多くのプロセッサを持つ CRCW PRAM でも，$t^{-1}(n)$ が並列計算時間の上下限となる言語を具体的に示すことができる．したがって，対角線論法における模倣のためのオーバーヘッドは生じ

平成9年6月20日受付
* 情報工学専攻 博士後期課程
** 九州芸術工科大学
*** 情報工学専攻 (現京都大学)

ず，稠密な階層が得られる．

以下，節 2. でモデルの定義と本稿の主定理を述べる．定理の証明は，節 3. で与える．

2. モデルと結果

Σ を記号の有限集合とする．本稿では，CRCW PRAM は Σ^N から $\{0,1\}$ への写像を計算する機械と定義する．CRCW PRAM は，プロセッサ$P_0, P_1, P_2, \ldots$ からなり，それらすべては入力の長さに関係なく同じプログラムを持つ．プログラムでは，有限個の局所変数 $x, y, \ldots$, 共有の配列 $M[0], M[1], M[2], \ldots$, 及び，以下の命令を使うことができる．

$x \leftarrow$ 定数（非負整数）
$x \leftarrow$ プロセッサ番号
$x \leftarrow y \circ z,\quad \circ \in \{+, -, \vee\}$ (y と z は整数)
$M[x] \leftarrow y$
$y \leftarrow M[x]$
GOTO *label* if $x = y$
HALT

同時書き込み規則は，いわゆる PRIORITY である．つまり，複数のプロセッサが共有配列の一つのセルに同時に書き込みを試みた場合は，プロセッサ番号が最も小さいプロセッサが書き込みに成功する．サイズ N の入力は，文字列$a_0 a_1 \cdots a_{N-1}$ である．最初，各 $a_i \in \Sigma$ は $M[i]$ に格納されている．プロセッサ数は N の多項式に制限されている．全プロセッサは同時に計算を開始し，計算を終了する際には同時に HALT 命令を実行しなければならない．計算結果は，$M[0]$ に 0 または 1 として与えられる．

長さ n の全ての入力に対して $(t(n))^{\Theta(1)}$ 時間の動作をする TM が存在し，かつ，$t(n)$ が単調増加関数であるとき，$t(n)$ は「良性な関数」であるという．関数 $t(n)$ の逆関数 $t^{-1}(N)$ を，$t(n) \le N$ を満たす最大の整数 n と定義する．

定理1. $t(n)$ は良性な関数で，その逆関数 $t^{-1}(n)$ が $O\left(\frac{\log n}{\log\log n}\right)$ で押えられるとする．このとき，多項式の範囲内で多くのプロセッサをもつ如何なる CRCW PRAM でも，受理するのに $\Theta(t^{-1}(n))$ 時間が必要かつ十分な言語が存在する．(証明は節 3. で与える.)

関数 $t^{-1}(n)$ としては，$c(\log\log n)^k$, $c(\log\log\log n)^k$, $\ldots$, $c(\log^* n)^k$ や逆アッカーマン関数などがあり，一般に扱う微小な関数の全てに対して定理は成立する．また，$t_1(n)$ と $t_2(n)$ が共に良性な関数ならば，時間量 $t_1^{-1}(t_2^{-1}(n))$ に対しても定理は成立する．

3. 定理 1 の証明

3.1 証明の概要

以下では，CRCW PRAM の入力長を N で表す．n は TM の入力長を表すために用いる．

本証明では，受理するのに $\Theta(t^{-1}(N))$ 時間が必要十分である言語L を示す．証明の大雑把な流れは，文献4)と同じであるが，受理時間を $\Theta(t^{-1}(N))$ で押えるために幾つかの新しいアイデアが必要となる．

直観的には，L は次のような構造をもつ．

$$L = \{\ \sigma = \sigma_1\sigma_2 : |\sigma| = N, |\sigma_2| = (\log N)^{\psi(N)},\ \text{かつ}\ \sigma_2\ \text{は奇数個の1を含む}\},$$

σ_1, σ_2 の細かい定義については，後の節で述べる．

m 変数のパリティ関数を計算する段数 k の回路には，入力線数無制限の素子が少なくとも $\Omega(2^{m^{\frac{1}{4k}}})$ 個は含まれていることが知られる1),9)．そこで，$m = (\log N)^{\psi(N)}$ とすれば，$\Omega(2^{(\log N)^{\frac{\psi(N)}{4k}}})$ 個の素子が必要となる．これが，N の多項式であるためには，k は $k \ge \psi(N)/4$ を満たさなくてはならない．上記言語 L において，σ_2 には，奇数個の 1 が含まれていなくてはならず，しかも，$|\sigma_2| = (\log N)^{\psi(N)}$ である．つまり，L を認識するには段数 $\Omega(\psi(N))$ が必要になる．論理回路と CRCW PRAM の模倣の関係8)から，CRCW PRAMでも L を受理するには $\Omega(\psi(N))$ 時間が必要となる．$|\sigma_2| = (\log N)^{\psi(N)} < N$ でなくてはならないので，定理は $\psi(N) \ne O\left(\frac{\log N}{\log\log N}\right)$ なる $\psi(N)$ に対しては成立しない．($\psi(N) = \Theta\left(\frac{\log N}{\log\log N}\right)$ を上下限とする言語は，文献1),9) において既に示されている．)

関数 $\psi(N)$ を，関数 $t(n)$ を用いて以下のように定義する．$t(n)$ は良性な関数なので，長さ n の全ての入力に対して $(t(n))^{\Theta(1)}$ 時間の動作をする TM T が存在する．多テープ TM は，動作時間を多項式の範囲で増やせば，1 方向無限の 1 本のテープを持ち，テープ記号 0,1 の TM で模倣可能である 2)．そこで，T もそのような TM と仮定し，T の動作時間を $t'(n)$ とおく．以降の節で述べる技術的理由から$\psi(N)$ は，$10(t'(n))^{2c} \le N$ を満たす最大の整数 n と定義する．ただし，c は十分大きい整数であり，c の値は節 3.3.1, 3.4 で述べる．本証明では，多項式の範囲で多くのプロセッサをもつ如何なる CRCW PRAM でも，上記 L を受理するには $\Theta(\psi(N))$ 時間が必要十分であることを示す．また，$\psi(N)$ は $\Theta(t^{-1}(N))$ で押えられることを証明する．

3.2 主要な命題

$Q = \{s_1, s_2, \ldots, s_k\}$ を T の状態の有限集合とする．本証明では，集合 Σ 上の言語 L を構成する．ここで，Σ には，記号として$0, 1, B, \$, \sharp$ とそれらに添字 $^1, ^2, \ldots, ^k$

を付けたものや，印 $', ^\dagger, ^\ddagger, ^@$ を付けたものが含まれる．

σ が言語 L に属しているための最初の条件は，σ が同じ長さの 10 個の部分から構成されていることである．

$$\sigma = \alpha_1\alpha_2\alpha_3\alpha_4\beta\gamma\delta\lambda\rho\omega$$

各部分の最後の記号には，部分と部分の境界を表す印 $\ddagger$ が付いている．α_1 は，$\alpha_1 = \alpha_{11}\alpha_{12}$ のように二つの小部分から構成される．$\alpha_{12} = \sharp\sharp\cdots\sharp\sharp^\ddagger$ は，α_1 の長さを調整するための列で，$|\alpha_{11}| = N_1$ とおくと，$|\alpha_1| = (N_1)^c$ である．

σ の各部分については，β と γ を除いて，後の節3.3 で述べる．β は次の構造をもつ長さ$(N_1)^c$ の文字列である．

$$\beta = p_0p_1p_2\cdots p_{N_1-1}\sharp\sharp\cdots\sharp\sharp^\ddagger,$$

ただし，$p_i \in \{0,1\}$ である．β の最初の N_1 個の文字は任意の文字列である．γ は，

$$\gamma = q_0q_1q_2\cdots q_{h-1}\sharp\sharp\cdots\sharp\sharp^\ddagger$$

なる長さ $(N_1)^c$ の文字列であり，$h = (\log N_1)^n$ である．ただし，n は α_1 の定義の際に TM を用いて定義する．(節3.1 で述べたように，$\psi(N)$ の値を n としている．) γ の先頭からh番目までの文字列に対する条件は，次の通りである．まず，β を「関数 f を定義している文字列」と見なす．つまり，$i = \mathrm{bin}(x_{\log N_1}\cdots x_2x_1)$ として，γ の i 番目の文字は，$f(x_1, x_2, \ldots, x_{\log N_1}) = p_i$ なる関数を定義していると見なす．ただし，$\mathrm{bin}(x_{\log N_1}\cdots x_2x_1)$ は 2 進数$x_1x_2\cdots x_{\log N_1}$ を 10 進数で表現したものである．(例えば$\mathrm{bin}(1100) = 3$.) γ に対する条件は，

$$f^n(q_0, q_1, \ldots, q_{h-1}) = 1,$$

である．β をうまく選べば，f^n はパリティ関数になることに注意．δ, λ, ρ は，f を高速に計算するために用いる．ω は，各部分の長さが $(N_1)^c$ であるか否かを短時間で判定するために用いる．これらは，節 3.3.1, 3.3.2 にて述べる．

命題 1. L は CRCW PRAM で $O(n)$ で認識できる．

ここで，$\psi(N) = n$ であることに注意．命題 1 の証明は，節 3.3 で与える．節 3.1 にて，$\psi(N)$ は，TM T の動作時間の関数 $t'(n)$ $(= (t(n))^{\Theta(1)})$ によって，$10(t'(n))^{2c} \leq N$ なる最大の n と定義された．つまり，$\psi(N)$ は，$t(n)$ の逆関数ではなく，$10(t'(n))^{2c}$ の逆関数になっている．そこで，次の命題が必要になる．

命題 2. $\psi(N)$ は $\Theta(t^{-1}(N))$ で押えられる．(証明は，節 3.4 で与える.)

命題 1,2 より，L は CRCW PRAM で $O(t^{-1}(N))$ で認識できる．下限の証明では，論理回路を用いるので，入力列は $\{0,1\}$ 上の文字列であるほうが都合が良い．そこで，L の各文字列の記号を，定数の長さの $\{0,1\}$ 上の文字列にマッピングして得られる言語を L^{B} とする．すると，次の命題 3 が成り立つ．

命題 3. L^{B} はCRCW PRAMで$O(t^{-1}(N))$ で認識できる．

下限の証明は，L^{B} の構造に違いはあるが，文献[4] と同様である．

命題 4 [4]**.** L^{B} が CRCW PRAM で$d(N_1)$ 時間で認識されるとすれば，$(\log N_1)^n$ 変数のパリティ関数は，素子数が N_1 の多項式，段数が $O(d(N_1))$, 各素子の入力線数が無制限の論理回路で計算できる．ただし，N_1 は σ に含まれるα_1 の長さである．

$d(N_1) \leq \log N_1$ だから，上記命題における $O(d(N_1))$ は $O(d(N))$ に置き換えても成立する．命題 2, 及び，節 3.1 と同じ計算により，次の命題が得られる．

命題 5 [4]**.** 如何なる論理回路でもL^{B} を認識するには $\Omega(t^{-1}(n))$ 段必要である．

論理回路による CRCW PRAM の模倣の結果[8]から，如何なる CRCW PRAM でも L^{B} を認識するには $\Omega(t^{-1}(n))$ 時間が必要である．

3.3 命題 1 の証明

本節では，$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta, \gamma, \delta, \lambda, \rho, \omega$ の条件を示し，さらに，それらの条件は微小な時間でチェックできることを示す．

3.3.1 $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ の構造

α_1 は，α_{11} と α_{12} からなる．α_{12} は，α_1 の長さを $(N_1)^c$ にするための列で，$\alpha_{12} = \sharp\sharp\cdots\sharp\sharp^\ddagger$ なる構造である．ただし，N_1 は α_{11} の長さである．α_{11} の i 番目のブロックは，$t'(n)$ 時間 TM の時刻 i における計算状況を表す．(**Fig. 1** 参照．最初のブロックは，時刻 1 における TM T の計算状況を表している．つまり，T のテープには入力列$i_1i_2i_3\cdots i_n$ と空白記号が書かれており，ヘッドは最初のマスの上にあり，状態は初期状態 s_1 である．**Fig. 1** の例では，ヘッドは次の時刻には 0 を書いて右へ動き，状態は s_7 になったことを表している.)

正確には，α_{11} の満たすべき条件は以下の通りである．

1. 記号 \$ は等間隔に現れ，最後の記号は \$ である．
2. j 番目のブロックの j 番目の記号には印 $\prime$ が付く．(つまり，印 $\prime$ は $t'(n) + 1$ ごとに等間隔に付けられている.) さらに，最後の記号に印 $\prime$ がつく．(つまり，各ブロックの長さ $t'(n)$ と，ブロック数は一致することになり，α_{11} の長さ N_1 は$(t_1(n))^2$ となる.)
3. 各ブロックは $\{0,1\}$ 上の文字列で始まり，B が続

$$\alpha_{11} = \overbrace{\underbrace{\overbrace{i_1{}^{1'} i_2 i_3 \cdots i_n}^{\text{input of TM}} B \cdots B\$}_{\text{block of length } t'(n)}\ \underbrace{0 i_2{}^{7'} i_3 \cdots i_n B \cdots B\$}_{\text{block of length } t'(n)} \cdots\cdots \underbrace{1^k 011 \cdots 01 B \cdots B\$'}_{\text{block of length } t'(n)}}^{N_1=(t'(n))^2}$$

Fig. 1　Structure of α_{11}

$$\alpha_2 = \overbrace{\underbrace{\overbrace{1^1 11 \cdots 1}^{|\alpha_1|=N_1} BB \cdots B\$}_{\text{block}}\ \underbrace{01^7 1 \cdots 1BB \cdots B\$}_{\text{block}} \cdots\cdots \underbrace{\overbrace{1^k 1 \cdots 1}^{\log N_1} 000 \cdots 0B\$}_{\text{block}} \sharp\sharp \cdots \sharp^{\ddagger}}^{(N_1)^c}$$

Fig. 2　Structure of α_2

き，最後は \$ である．各ブロックは，0 または 1 を少なくとも一つ含み，かつ，B も少なくとも一つ含む．また，\$ は最後の一つのみである．

4. 最初のブロックの最初の文字には，初期状態を表す添字 1 が付けられている．
5. 隣接するブロック（計算状況）間の関係は，TM T の遷移関数に従っている．
6. 最後のブロックの最初の文字には，最終状態を表す添字 k が付けられている．

本稿の α_1 は，文献4) の α_1 とは多少の構造が違いがある．しかし，ほぼ同様の方法で，α_1 が上記条件を満たしているか否かを定数時間で判定できる．

Fig. 2 のように，$\alpha_2, \alpha_3, \alpha_4$ は α_1 と同じ長さ，同じ構造をもつ．ただし，これらを定義する TM は，一進数で与えられた N_1 の値から $\log N_1, N_1 \log N_1, (N_1)^2$ を計算する TM とする．これらの TM の計算状況の列が長さ $(N_1)^c$ に収まるように，整数c は十分大きくとっておく．

$\alpha_1, \alpha_2, \alpha_3, \alpha_4$ のそれぞれの長さが $(N_1)^c$ であることをチェックするには，次節の ω を用いる．

σ の 10 個の部分のそれぞれの長さは $(N_1)^c$ となるので，σ 全体の長さは $N = 10(N_1)^c$ となる．$N_1 = (t'(n))^2$ だから，$N = 10(t'(n))^{2c}$ である．ψ は，N から n の値を求める関数なので，CRCW PRAM の入力が σ としての条件を満たしていれば，n の値は，α_1 の最も左にある B の位置から簡単に求めることができる．

3.3.2　ω の構造

ω を定義するため，まず，

$$\omega_1 = \underbrace{\sharp\sharp\sharp \cdots \sharp\$}_{N_1}$$

とする．ω_1 の j 番目の文字に添字 1 をつけた列を$\omega_1(j)$ とおく．このとき，$\omega_2 = \omega_1(1)\omega_1(2)\cdots\omega_1(N_1)$ とすれば，これは長さ $(N_1)^2$ の文字列となる．

今，長さ $(N_1)^i$ の文字列 ω_i が定義されているとする．ω_i の j 番目の文字に添字 i をつけた列を $\omega_i(j)$ とすれば，

$$\omega_{i+1} = \omega_i(1)\omega_i(2)\cdots\omega_i((N_1)^i)$$

は，長さ $(N_1)^{i+1}$ の文字列となる．長さ $(N_1)^c$ の文字列 ω_c の最後の記号に，印 $\ddagger$ を付けた文字列が ω である．ω の部分が，規則を満たしているか否かを判定する方法は，節 3.3.1 で与えた条件2 と同様である．(つまり，ω_{i+1} の最初と最後の文字に添字 i が付けられ，かつ，ω_{i+1} の中で添字 i が $(N_1)^i + 1$ ごとに現れることを確かめればよい.)

$\sigma = \alpha_1\alpha_2\alpha_3\alpha_4\beta\gamma\delta\lambda\rho\omega$ の 10 個の部分の各境界を表す印 $\ddagger$ が等間隔に現れていることをチェックすることで，各部分の長さが $(N_1)^c$ であるか否かを判定できる．

3.3.3　δ, λ, ρ の構造

δ, λ, ρ は，β, γ で定義された$f^n(q_0, q_1, \ldots, q_{n-1})$ の値を $O(n)$ 時間で計算するために用いる．δ, λ, ρ は，長さが $(N_1)^c$ となるように後ろに記号$\sharp$ からなる長い列が付けられている．それ以外は，文献4) と同じである．f^n の計算方法も同様にできるので，ここでは省略する．

3.4　命題 2 の証明

命題 1 は，$10(t'(n))^{2c}$ の逆関数の時間で言語 L を認識できることを述べている．しかし，定理 1 の上限を証明するためには，$t(n)$ の逆関数の時間で認識できることを述べなくてはならない．そこで，ここでは$10(t'(n))^{2c}$ の逆関数 $\psi(N)$ が $\Theta(t^{-1}(N))$ で押えられることを述べる．

$t'(n)$ は，$(t(n))^{\Omega(1)}$ で押えられるので，ある正定数 n_1 より大きいすべての整数 n に対し，$t'(n) > (t(n))^{1/c_1}$ となるような定数 c_1 が存在する．従って，c の値として $c_1/2$ より大きい定数を選べば，$10(t'(n))^{2c} > 10(t(n))^{2c/c_1} > t(n)$ となる．故に，$10(t'(n))^{2c}$ の逆関数は，$t(n)$ の逆関数よりも成長が遅い．だから，ある正定数 N_1 より大きいすべての N に対して，$\psi(N) < t^{-1}(N)$ となるので，$\psi(N)$ は $O(t^{-1}(N))$ で押えられる．

$t'(n)$ は，$(t(n))^{O(1)}$ で押えられるので，ある正

定数 n_2 より大きいすべての整数 n に対し，$t'(n) < (t(n))^{c_2}$ となるような定数 c_2 が存在する．従って，$10(t'(n))^{2c} < 10(t(n))^{2cc_2}$ となる．ここで，g を x から $y = 10x^{2cc_2}$ を計算する関数とする．すると，$10(t'(n))^{2c} < g(t(n))$ なので，$10(t'(n))^{2c}$ の逆関数 $\psi(N)$ は $t^{-1}(g^{-1}(N))$ よりも成長が速い．だから，$\psi(N) > t^{-1}(g^{-1}(N)) = t^{-1}((N/10)^{1/2cc_2})$ となる．$t^{-1}(N)$ は $O(\frac{\log N}{\log\log N})$ で押えられるので (つまり，$t^{-1}(N) < \log N$ なので)，上記の式は $\psi(N) > t^{-1}((N/10)^{1/2cc_2}) > \frac{1}{2cc_2}t^{-1}(N/10) > \frac{1}{2cc_2}(t^{-1}(N) - \log 10)$ となる．従って，$\psi(N)$ は $\Omega(t^{-1}(N))$ で押えられる．

参考文献

1) J. Hastad, "Almost optimal lower bounds for small depth circuits," *in:* Proc. 18th ACM Symp. on Theory of Computing (1986) 6-20.
2) J.E. Hopcroft and J.D. Ullman, "Introduction to automata theory, languages and computation," Addison-Wesley, Reading, MA (1979).
3) K. Iwama and C. Iwamoto, "Parallel Complexity Hierarchies Based on PRAMs and DLOGTIME-Uniform Circuits," *in:* Proc. Eleventh Annual IEEE Conference on Computational Complexity, Philadelphia (1996) 24–32.
4) K. Iwama, C. Iwamoto, and M. Morshed, "Time lower bounds do not exist for CRCW PRAMs," *Theoretical Computer Science*, 155 (1996) 411–424.
5) 岩本宙造，岩間一雄，並列計算用に拡張した TM の時間計算量の階層，電子情報通信学会論文誌，J80-D-I 5 (1997) 421–427.
6) D. S. Johnson, A catalog of complexity classes, *in* "Handbook of Theoretical Computer Science," (J. van Leeuwen, ed.), Vol. A, 69-161, MIT Press, Amsterdam, 1990.
7) W.W. Kirchherr, "A hierarchy theorem for PRAM-based complexity classes," *in*: Proc. 8th Conference on Foundations of Software Technology and Theoretical Computer Science (Lecture Notes in Computer Science 338) (1988) 240–249.
8) L. Stockmeyer and U. Vishkin, "Simulation of parallel random access machines by circuits," *SIAM J. Comput.* 13 (1984) 409–422.
9) A. Yao, "Separating the polynomial-time hierarchy by oracles," *in:* Proc. 26th IEEE Symp. on Foundations of Computer Science (1985) 1-10.