数理解析研究所講究録
381巻 1980年 116-132

組合せ論理回路の幅と段数について

京都大学　工学部　安浦寛人
矢島脩三

あらまし　組合せ論理回路の幅は，回路を並列計算モデルとして見たときの計算の並列度の測度であると考えられる．幅 w，段数 d の回路で計算できる関数は，段数 $O(w \log d)$ の回路で計算できることを示す．

1. はじめに

近年の集積回路技術の急速な進歩により，大規模な論理回路がLSIや超LSIとして実現できるようになり，算術演算や信号処理のような特に高速性が要求される分野において，従来ファームウェアやソフトウェアで行なっていた機能がより高速なハードウェアで置き換えられ始めている．このような動きにともない，ハードウェア向きの高速並列アルゴリズムの研究が重要性を増してきた．特に，組合せ論理回路は，ハードウェアの基本回路であるとともに，並列計算機構のモデルと

しても理論的に重要であり，様々な研究がなされている〔SAVA 76〕〔YASUY7909〕．従来の研究では，組合せ回路の複雑さを測る尺度として，主に回路の素子数や段数が使われている．並列計算のモデルとして見ると素子数は総計算量，段数は計算時間に対応すると考えられる．一方，ペブルゲーム等の並列計算モデルや実際の並列計算システムでは，計算の多重度あるいは並列度を表わす尺度について研究されている．これは，並列計算系のプロセサの数に対応付けられる量である．本稿では，組合せ論理回路の並列度を表わす尺度として，回路の幅を考える．

組合せ回路の幅とは，組合せ回路を同期的に情報を処理して流す管と見たとき，並列に最も大量の情報が流れる部分の太さ（線の数）である．すべての2変数論理関数を実現する素子を基本素子として使うとき，任意の論理関数は幅2の回路で計算できる．さらに，段数d，幅wの回路で計算できる関数は，段数$O(w \log_2 d)$の回路でも計算できることを示す．これは，幅が段数に較べて十分小さい回路は，再構成して段数を減らせることを表わしている．UngerのCIT実現（Combinational Iterative Tree）はこの結果の特殊な場合と見なすことができる〔UNGE7704〕．

組合せ回路の素子数や段数とチューリング機械の時間や空

間との関係は，計算の理論の中で重要な問題を含んでいる〔BORO7712〕．本稿では，前述の結果を利用して，$T(n)$時間限定$S(n)$テープ限定決定性チューリング機械が，$O(S(n)\log T(n))$の段数の組合せ回路でシミュレートできることを示す．この結果は，Borodinの$O(S^2(n))$段のシミュレーションの一つの改善となっている．

2．諸定義

$K=\{0,1\}$とする．すべての2変数論理関数の集合$B=\{f \mid f: K^2 \to K\}$を基底とする．基底B上の組合せ回路Nを次のように定義する．Nはラベルのついた閉路を含まない有向グラフで，各節点の入次数は0または2である．入次数0の節点は，入力点と呼ばれ，K上の変数$x_1, x_2, \cdots, x_n$または定数0，1のいずれかがラベルとして付けられる．入次数が2の節点は，計算点と呼ばれ，B中の関数がラベルとして付けられる．各節点vに対して，その節点の出力となるべき関数$res(v): K^n \to K$を次のように対応させる．

$$res(v)=\begin{cases} v\text{のラベル} & (v\text{が入力点のとき}) \\ g(res(v_1), res(v_2)) & (v\text{が計算点のとき}) \end{cases}$$

ここに$g \in B$はvのラベルであり，v_1，v_2はそれぞれvの入力枝を出力枝とする節点である．n変数論理関数$f: K^n \to K$

を回路Nが計算するとは，ある節点vがあって，$f = res(v)$となることである．このvを特に出力点と呼ぶ．本稿では，すべての節点から出力点に至る経路が存在する回路のみを考える．

組合せ回路Nの計算点に対し，次のようにレベルを定義する．

(1) 出力点のレベルを1とする．

(2) 各計算点vのレベルは，vの出力枝を入力枝とする各節点のレベルの最大値に1を加えたものとする．（図1参照）

回路Nの素子数 $size(N)$ とは回路中の計算点の数であり，段数 $depth(N)$ とはレベルの最大値である．図1の回路は，素子数11，段数6である．

論理関数fの計算量 $C(f)$ および計算時間 $D(f)$ をそれぞれfを計算する回路の素子数，段数の最小値によって定義する．

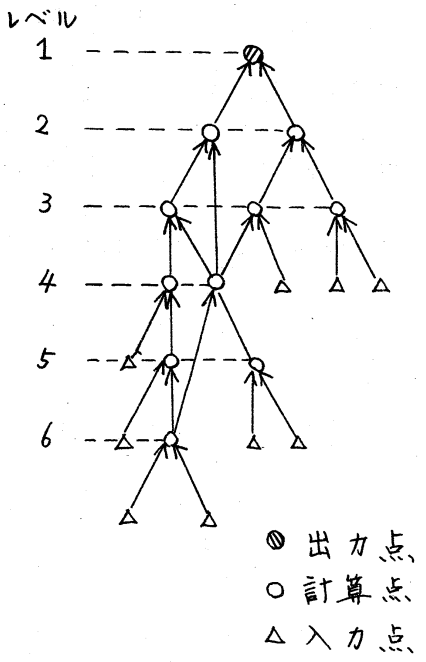

図1．組合せ回路N

回路Nのレベルiにおける幅とは，iより小さなレベルの計算点への入力枝を出力枝とするレベルがi以上の計算点の数である．回路Nの幅 $width(N)$

とは，各レベルにおける幅のうち最大のものの値である．

〔例〕図1の回路において各レベルにおける幅は

レベル	1	2	3	4	5	6
幅	1	2	4	2	3	1

となるから，回路の幅は4となる．

3．幅の性質

本節では，幅の基本的な性質について述べる．

〔性質1〕 任意の論理関数は，幅2の回路で計算できる．

（証明） f の積和形表現を考える．各積項は明らかに幅1の回路で計算できる．これらの積項の論理和を図2のような回路で計算する．図中斜線を入れた計算点はレベルを合わせるために入れた恒等関数を実現する計算点である． （証明終）

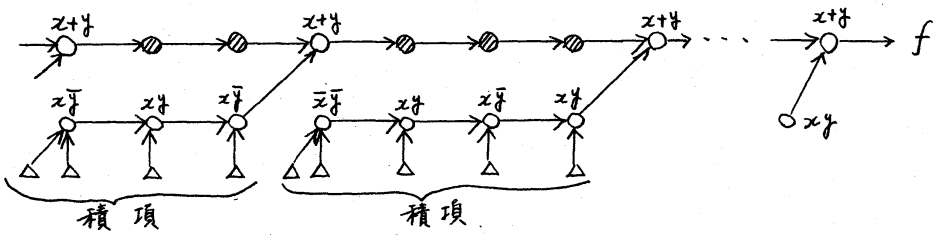

図2 幅2の回路

〔性質2〕 基底B上で幅1の回路では計算できない論理関数が存在する．

（証明） 3変数関数のうち $xy+yz+zx$ および $xyz+\bar{x}\bar{y}\bar{z}$ とこ

れらにNPN等価な関数（計16個）は幅1の回路では計算できないことが容易に示せる．$n-3$個の変数を適当に固定したとき，これら16個の3変数関数のうちの1つとなるようなn変数関数（$n \geqq 3$）は，幅1の回路では計算できない．

（証明終）

回路の幅と段数，素子数の間には次のような関係が成立する．Nを論理関数fを計算する回路とする．$C(f)$，$D(f)$の定義および前述の性質より，

$$\begin{cases} size(N) \geqq C(f) \\ depth(N) \geqq D(f) \\ width(N) \geqq 1 \quad \text{または} \quad 2 \end{cases}$$

が成立する．幅の定義より，各レベルの素子の数は，$width(N)$を越えないから，

$$depth(N) \cdot width(N) \geqq size(N)$$

が成立する．また，N中のすべての節点から出力点に至る経路が存在する回路のみを考えており，1つのレベル毎に幅は高々半分にしかなり得ないことより，$width(N)$の幅がレベル1において幅1となるためには，

$$depth(N) \geqq \log_2(width(N)) + 1$$

が成立しなければならない．レベルiの幅を$width(N)$とする．

i 以上のレベルを持つ計算点の数は定義より少なくとも $width(N)$ 以上である．一方，すべての節点から出力点へ至る経路が存在し，各節点の入次数は2であるから，$i-1$ 以下のレベルの計算点の数は，$width(N)-1$ 以上となる．よって，

$$size(N) \geq 2\,width(N) - 1$$

が成立する．

4. 段数削減定理

本節では，段数と幅の関係に注目し，段数は大きくても幅が小さな回路で計算できる関数は，幅が大きく段数が小さな回路でも計算できることを示す．この結果は，幅が小さく段数が大きな回路の構成法が知られている関数に対し，段数の小さな高速回路の構成法を与えることになり，実用上も有用な手法である．

〔定理1〕（段数削減定理） 論理関数 f が段数 d，幅 w の組合せ回路で計算できるとき，f を計算する段数 $(w+\lceil \log_2 w \rceil + 1)\lceil \log_2 d \rceil + 1$ 以下の回路を構成することができる．

（証明）f を計算する段数 d，幅 w の回路 N_0 を考える．N_0 を $\lceil \frac{d}{2} \rceil$ 以下のレベルの計算点からなる部分 N_1 と $\lceil \frac{d}{2} \rceil + 1$ 以上のレベルの計算点からなる部分 N_1' にわける．レベル $\lceil \frac{d}{2} \rceil + 1$ における幅を p とすると，$p \leq w$ である．N_1 のコピーを 2^p

個作り，N_1' からの入力線（p本）に 2^p とおりの K^p のパターンをそれぞれ与える．N_1' の計算結果に従って，N_1' のコピーのうちの1つの出力を選択することによって f が計算できる．（図3参照）．このようにして作った回路を $\tilde{N}_0$ とすると，

$$depth(\tilde{N}_0) = \lfloor \frac{d}{2} \rfloor + \lceil \log_2 p \rceil + p + 1$$
$$\leq \lfloor \frac{d}{2} \rfloor + \lceil \log_2 w \rceil + w + 1$$

が成立する．N_1, N_1' はそれぞれ幅 w 以下の回路であるから，N_0 に施したのと同様の操作を N_1, N_1' に施すことができる．この操作を次々に $\lceil \log_2 d \rceil$ 回繰り返して施すことにより，段数 $(w + \lceil \log_2 w \rceil + 1)\lceil \log_2 d \rceil + 1$ 以下の回路が得られる．(証明終)

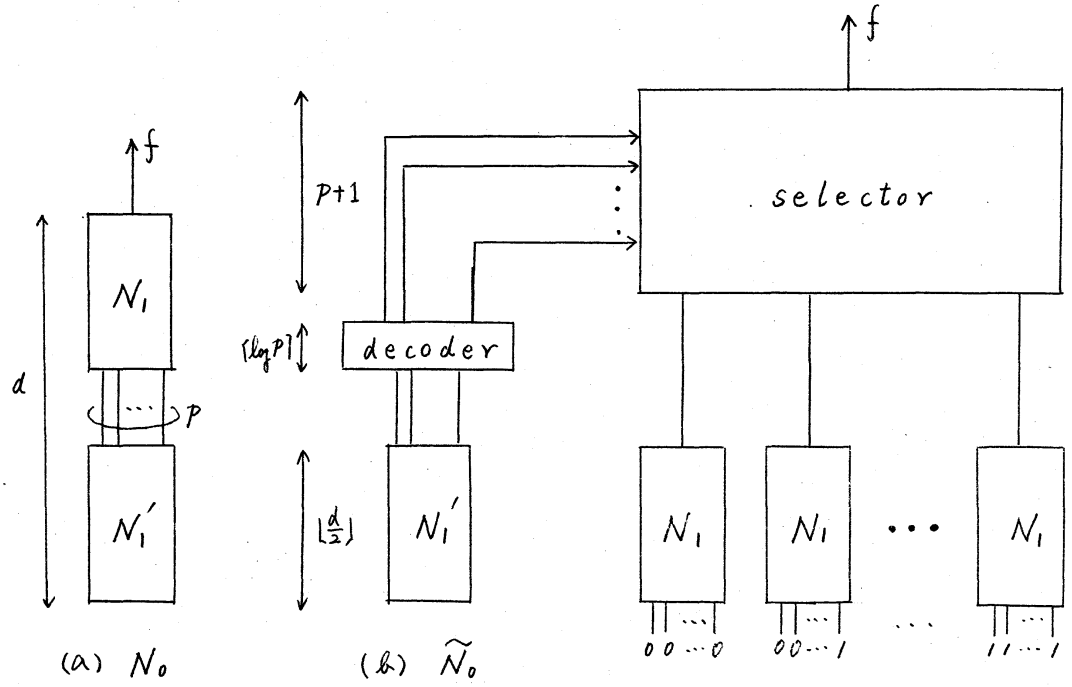

図3　段数削減操作

定理1は，段数が幅に対して十分に大きな回路で計算できる関数は，段数のより小さな回路で計算できることを示している．このとき，回路の幅と素子数は，重複して使えるコピーを考慮して幅 $O(2^w d)$，素子数 $(2^{2w} w d)$ となる．次に，逆に幅が段数に対して比較的大きな回路で計算できる関数を，より小さな幅の回路で計算する場合を考える．

〔定理2〕 論理関数 f が段数 d の回路で計算できるとき，f を計算する幅 d 以下，段数 2^{d-1} 以下の回路が構成できる．

（証明） f を計算する段数 d の回路を N とする．N から段数 d でしかも各計算点の出次数が1である f を計算する回路 N_0 は容易に作れる．N_0 は木となる．N_0 の左部分木を N_1，右部分木を N_2 とする（図4(a)）．N_1, N_2 は段数 $d-1$ 以下の回路である．今，$d-1$ まで定理が成立したとする．すなわち，N_1, N_2 はそれぞれ幅 $d-1$，段数 2^{d-2} 以下の回路 $\tilde{N}_1$, $\tilde{N}_2$ で実現できたとする．この $\tilde{N}_1$ と $\tilde{N}_2$ を使って，図4(b)のように $\tilde{N}_0$ を構成する．すなわち，$\tilde{N}_2$ の出力を $depth(\tilde{N}_1)-1$ 段の恒等関数の計算点を通して出力点に入れる．このようにすると，明らかに，

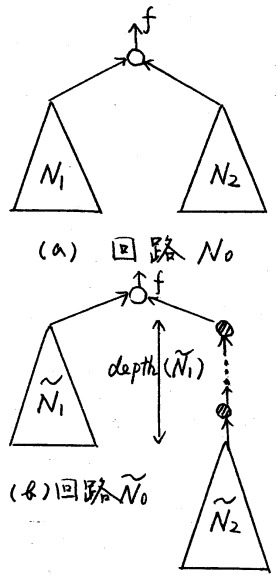

図4 幅の削減

$$width(\tilde{N}_0) = \max(width(\tilde{N}_1)+1, width(\tilde{N}_2))$$

となる．仮定より，$width(\tilde{N}_1) \leqq d-1$，$width(\tilde{N}_2) \leqq d-1$ であるから，$width(\tilde{N}_0) \leqq d$．段数についても，

$$depth(\tilde{N}_0) = depth(\tilde{N}_1) + depth(N_2) \leqq 2 \cdot 2^{d-2} = 2^{d-1}.$$

よって，d においても定理は成立する．$d=1$ のときは明らかに定理は成立するから，帰納的に証明された．　（証明終）

前節で述べたように，すべての論理関数は幅2で実現できるが，その回路の段数と素子数は一般には変数の数の指数関数のオーダとなる．定理2は，変数の数の対数のオーダの段数で実現できる関数は，同じオーダの幅かつ多項式オーダの段数の回路でも計算できることを示している．定理1と定理2は，段数と幅の間のトレードオフであると考えられる．

5．関数列の複雑さ

論理関数の複雑さを，入力変数の数との関係において調べるために，我々は論理関数列を導入した〔YASUY7909〕．論理関数列（以後，関数列と省略）は，各自然数 n に対し，n 変数関数をちょうど1つずつ含むような関数の無限集合である．関数列 $\{f_n\}$ は，列中の各 n 変数関数 $f_n(x_1, x_2, \cdots, x_n)$ の出力を1にする入力 $(x_1, x_2, \cdots, x_n)$ をK上の系列 $x_1 x_2 \cdots x_n$ と見なすことにより，K上の形式言語（系列の集合）と1対1に

対応付けができる．

高速並列アルゴリズムの研究においては，どのような関数列が段数の小さな回路で実現できるかが重要である．特に，$\{f_n\}$ 中の各 f_n が $O(\log n)$ の段数で実現できる関数列のクラスの特性化は実用上重要である．我々は，上記の言語と関数列との対応に基づいて，このクラスをチューリング機械における計算の複雑さのクラスとの関係から特性化しようと試みた〔YASUY8001〕．本稿では，定理1および定理2に基づいて，別の面からの特性化を試みる．

〔定義〕 関数列のクラスを次のように定義する．

(1) LOGDEPTH: $O(\log n)$ の段数の回路で計算できる関数列のクラス．

(2) LOG^{2}DEPTH: $O((\log n)^2)$ の段数の回路で計算できる関数列のクラス．

(3) CONSTWIDTH: n に依存しない定数以下の幅で，$P(n)$ 以下の素子数の回路で計算できる関数列のクラス．$P(n)$ は n の多項式．

(4) LOGWIDTH: $O(\log n)$ の幅で，$P(n)$ 以下の素子数の回路で計算できる関数列のクラス．$P(n)$ は n の多項式．

(5) R: K上の正規集合のクラス．

n は入力変数の数である．

〔定理3〕 $R \subsetneqq CONSTWIDTH \subseteqq LOGDEPTH \subseteqq LOGWIDTH \subseteqq LOG^2DEPTH$.

(証明) (1) $R \subsetneqq CONSTWIDTH$

R中の言語に対応する関数列は，Rを受理する有限オートマトンを一次元一方向展開した回路〔UNGER 7704〕によって計算できる．この回路の幅は有限オートマトンの状態数によって決まるからnには依存しない．真に含まれることは，$\{w\,w^R \mid w^R$ はwの反転$\}$ がRには含まれずCONSTWIDTHには入ることより示される．

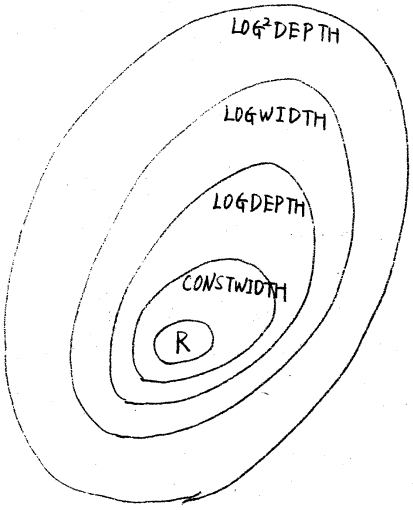

図5．クラス間の包含関係

(2) $CONSTWIDTH \subseteqq LOGDEPTH$ CONSTWIDTH中の任意の関数列$\{f_n\}$中の関数f_nに対し，幅w_0，段数$P(n)$の回路が存在する．定理1より，f_nに対して，段数$O(w_0 \log P(n)) = O(\log n)$の段数の回路が存在する．

(3) $LOGDEPTH \subseteqq LOGWIDTH$ LOGDEPTH中の任意の関数列$\{f_n\}$の中の関数f_nに対し，段数$c_1 \log n + c_2$以下の回路が存在する．定理2より，f_nに対し，幅$c_1 \log n + c_2$以下，段数$2^{c_2-1} n^{c_1}$の回路が存在するから，$\{f_n\} \in LOGWIDTH$である．

(4) $LOGWIDTH \subseteqq LOG^2DEPTH$ LOGWIDTH中の任意の関数列$\{f_n\}$中

の関数 f_n に対し，幅 $c_1 \log n + c_2$ 以下，段数 $p(n)$ 以下の回路が存在する．定理1より，f_n は段数 $O((\log n)^2)$ の回路で計算できる．（証明終）

上の証明において，(2)，(3)，(4)の関係について，等号が成立するか否かは未解決である．LOGDEPTHの中で，CONSTWIDTHに入ることが示されていない例としては，対称関数の列 $\{f_n \mid f_n = S^n_{\lceil \frac{n}{2} \rceil}\}$ やnビットの2進数の積のnビット目を求める関数等がある．LOGWIDTHの中でLOGDEPTHに入ることが示されていない関数列としては，出次数1の有向グラフの隣接行列を入力として，節点1から節点nへ至る経路が存在するか否かを判定する問題がある．LOG^{2}DEPTHに入っていて，LOGWIDTHに入るか否かがわかっていない関数列の例としては，nビットの整数の逆数の半nビットを求める関数がある．この関数列は，除算の高速化において非常に重要である〔SAVA76〕．

チューリング機械を回路でシミュレートする問題は，チューリング機械における時間計算量やテープ計算量の問題を回路の素子数や段数の議論に置き換えて解くために考えられた〔SAVA76〕．その後，多くの研究により，時間計算量やテープ計算量と素子数，段数の関係が明らかになり，計算の理論の一つの重要な分野となっている．ここでは，定理1を使って

チューリング機械をシミュレートする回路の段数の新しい上界を与える．ここで考えるチューリング機械は，2方向読取専用の入力テープと2方向で読書き可能な作業用テープを持つ決定性の受理器とする．

Schorr は $T(n)$ 時間限定 $S(n)$ テープ限定のチューリング機械が，$O(T(n)\log S(n))$ の素子数の回路でシミュレートできることを示した〔SCHN76〕．Borodin は $S(n)$ テープ限定チューリング機械（非決定性でもよい）が，$O(S^2(n))$ の段数の回路でシミュレートできることを示している〔BORO7712〕．

〔定理4〕 $T(n)$ 時間限定 $S(n)$ テープ限定チューリング機械で受理できるK上の言語に対応する関数列は，段数 $O(S(n)\log T(n))$，素子数 $O(S(n)T(n)2^{2S(n)})$ の組合せ回路で計算できる．ただし，$S(n)=\Omega(\log n)$，$T(n)=\Omega(n)$ であり，n は入力の長さとする．

（証明）チューリング機械の入力アルファベットを $K=\{0,1\}$，テープ記号の集合を Γ，有限制御部の状態数を Q，その遷移関数を δ とする．図6のような回路を，一方向に $T(n)$ 個結合すると，明らかにこのチューリング機械をシミュレートできる．δ を計算する回路は n に依存しないから，図6の回路の段数は，高々 $O(\max(n\log n), \log S(n))$ である．また，適当に恒等関数を入れてレベルを調整すると，その幅は $O(S(n)+$

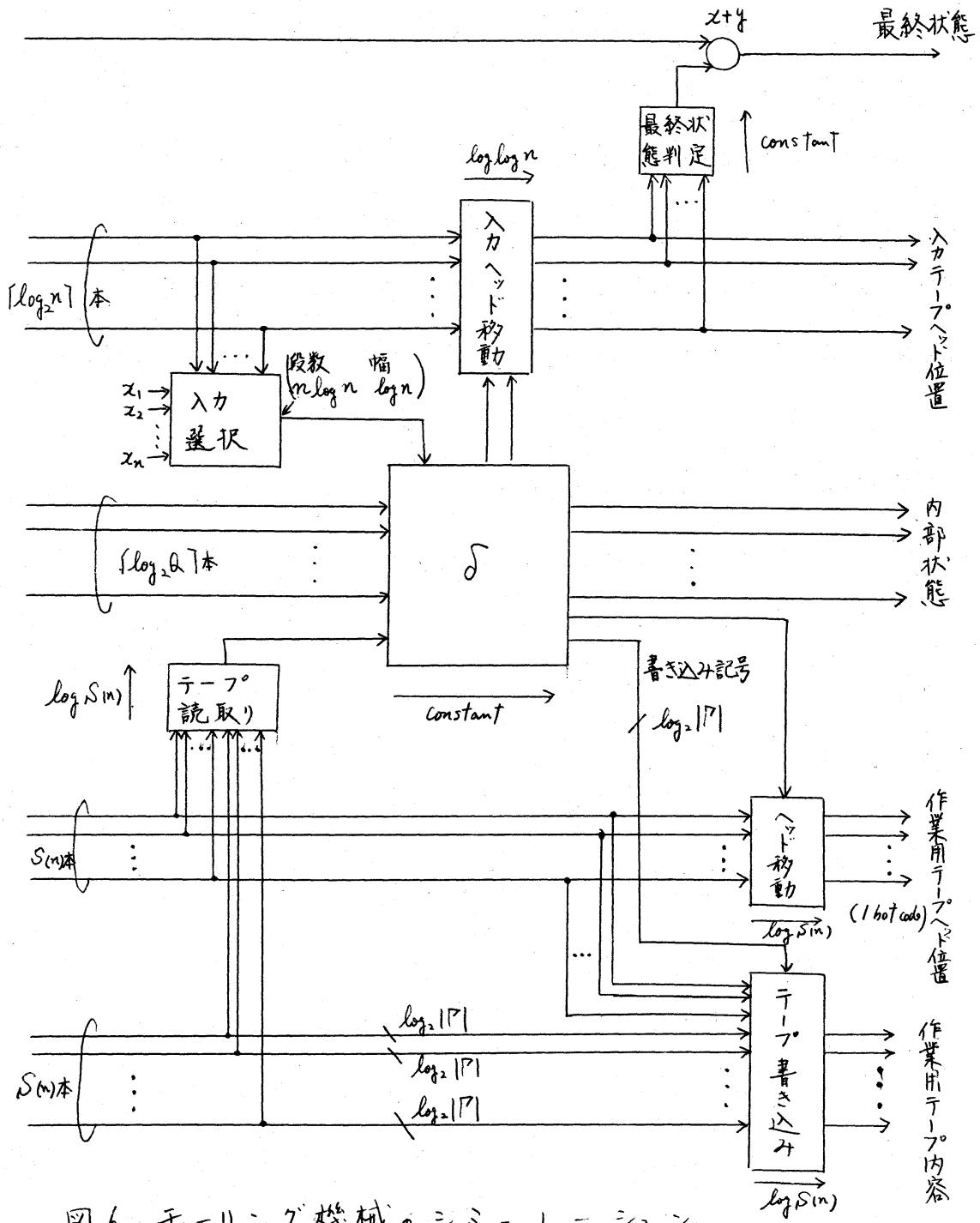

図6 チューリング機械のシミュレーション

$\log n$) となる. $S(n) = \Omega(n)$ であるから, これは $O(S(n))$ とできる. よって定理1を適用すると,

$$O(S(n) \log \{T(n) \times \max(n \log n, S(n))\})$$

となる. ここで, $T(n) \geq S(n)$, $T(n) = \Omega(n)$ であることを考慮すると, このシミュレーション回路は定理1の証明の手法で $O(S(n) \log T(n))$ の段数の回路に再構成できたことになる.

(証明終)

〔系〕 $\log n$ テープ限定決定性チューリング機械で受理できる言語に対応する関数列は, 段数 $O((\log n)^2)$, 素子数 $O(p(n))$ ($p(n)$ は多項式) の回路で計算できる.

$T(n)$ が多項式オーダのとき, $\log T(n) = O(S(n))$ が成立するから, この結果は, Borodin の結果 $O(S^2(n))$ の決定性の場合を真に強めている.

6. おわりに

並列計算の並列度の尺度として, 組合せ回路の幅を議論した. 幅は, 並列計算の中で同時に使われる資源の数に対応しており, 処理装置や記憶素子を対象に考えると, 特理的な制約から幅が定数や対数オーダでおさえられることが要請され

る場合も多いと思われる．今後，組合せ回路の幅とこのような上位レベルの並列度の関連の研究が課題である．

謝辞　御討論頂いた本学上林弥彦助教授はじめ矢島研究室の諸氏に感謝します．

文献

[BORO7712] A. Borodin, "On relating time and space to size and depth", SIAMJ. Comput. Vol.6, No.4, Dec. 1977.

[SAVA76] J. E. Savage, "The complexity of computing", Wiley-Interscience, 1976.

[SCHN76] C. P. Schnorr "The network complexity and the Turing machine complexity of finite functions", Acta Informatica, Vol.7, pp.95-107, 1976.

[UNGE7704] S. H. Unger, "Tree realization of Iterative Circuits", IEEE Trans. on Comput. Vol. C-26, No. 4, April 1977.

[YASUY7909] 安浦，矢島，"論理関数を実現するのに必要な論理回路の段数について"，電子通信学会論文誌 J62-D, No.9. 1979 9月．

[YASUY8001] 安浦，矢島，"論理関数の複雑さに関する完全問題について"，オートマトンと言語研究会資料 AL79-97, 1980 1月．