

Title	VLSI回路の面積と計算時間の複雑さに関する研究
Author(s)	和田, 幸一
Citation	大阪大学, 1983, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/659
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

VLSI回路の面積と計算時間の
複雑さに関する研究

和田 幸一

工41

36

6061

VLSI回路の面積と計算時間の
複雑さに関する研究

昭和58年2月

和田 幸一

内 容 梗 概

本論文は、筆者が大阪大学大学院基礎工学研究科（物理系専攻）の学生として、都倉研究室において行った研究のうち、VLSIモデルにおけるVLSI回路の面積と計算時間の複雑さに関する研究をまとめたものである。

半導体技術、集積回路技術の飛躍的な発展により、回路のVLSI化が可能になり、従来の論理設計ではあまり考慮されなかったVLSI回路特有の尺度が用いられるようになった。VLSI回路においては、回路中に含まれている素子と配線や入出力端子等を含めた全体の面積と、回路の性能を表した計算時間が重要な評価尺度である。また、回路面積 A と計算時間 T の間にはトレードオフがあるので、面積時間積($AT^\alpha: \alpha \geq 0$)はバランスのとれたVLSI回路の評価尺度であると考えられる。

本論文の第1章では、VLSI回路の面積時間複雑度に関する研究の現状、その工学上の意義、本研究の新しい諸結果について概説している。

第2章では、本論文で用いるVLSI回路を面積と計算時間で評価するVLSIモデルと諸定義について述べている。

第3章では、一般の n 入力 m 出力論理関数のクラスに対して、その関数を実現する回路の面積時間積($AT^\alpha: \alpha \geq 1$)の一つの下界を導出し、復号化関数、符号化関数、対称関数など基本的に重要な関数がここで得られた下界を達成することが示される。第4章では、入出力端子は回路の境界上におく（境界配置の仮定と呼ぶ）という仮定を付加えたVLSIモデルのもとで、第3章で取り扱った関数のクラスに対して、面積時間積の下界が導出される。このとき、面積時間積の下界が境界配置の仮定に真に依

存することが示される。また、この下界をほぼ達成する関数が存在することが証明される。

第5章と第6章では、 AT^d に対して $d=0$ の場合、すなわち、回路の面積を評価する手法が議論されている。第5章では、順序回路による構成を許した場合、第6章では、構成を組合せ回路に限定した場合のそれぞれについて、回路をVLSIに埋め込むとき必要となる面積の下界を導出する手法が示され、その結果を利用して、復号化関数、乗算、ローディングなどを実現する回路の面積下界が示される。また、この下界はいずれも最良のものである。

第7章の結論では、本研究で得られた主な結果をまとめ、今後に残された問題について述べている。

関連発表論文

- [1] 和田, 萩原, 都倉: "VLSI回路の面積時間複雑度", 信学技報, EC81-2 (1981-04).
- [2] 和田, 萩原, 都倉: "VLSI回路における面積時間のトレードオフについて", 信学技報, EC81-16 (1981-06).
- [3] 和田, 萩原, 都倉: " n 変数論理関数の面積時間複雑度", 信学論(D), J64-D, 8, pp.676-681 (昭56-08).
- [4] K.Wada, K.Hagihara and N.Tokura: "Area-Time Bounds for VLSI Circuits," Proceedings 19th Annual Allerton Conference on Communication, Control, and Computing, p.234 (September-October 1981).
- [5] 和田, 萩原, 都倉: "面積時間積を用いた論理関数の複雑さについて", 信学技報, AL81-77 (1981-11).
- [6] 和田, 萩原, 都倉: "境界配置条件を課したVLSIモデルにおける面積複雑度について", 信学技報, AL81-96 (1982-01).
- [7] 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), J65-D, 4, pp.478-485 (昭57-04).

[8] K.Wada, K.Hagihara and N.Tokura : "Area and Time Complexities of VLSI Computations," Proceedings the 7th IBM Symposium on Mathematical Foundations of Computer Science, Mathematical Theory of Computations, IBM Japan, pp. 1-77 (June 1982).

[9] 和田, 萩原, 都倉 : " VLSIモデルにおける組合せ論理回路の面積下界について ", 信学技報, AL82-30, (1982-09).

[10] K.Hagihara, K.Wada and N.Tokura : "Effects of Practical Assumption Area Complexity in VLSI Computation," to appear in Lecture Notes in Computer Science, Springer-Verlag.

VLSI回路の面積と計算時間の 複雑さに関する研究

目 次

1.	緒論	1
2.	VLSIモデルと諸定義	9
3.	面積時間複雑度(1)	
	——境界配置の仮定を要請しない場合——	17
3.1.	面積時間積 AT^α の下界	17
3.2.	面積時間積 AT^α の上界	26
3.2.1.	復号化関数, 符号化関数	28
3.2.2.	一方向次元繰り返し回路 (CUODIC)	44
3.2.3.	優先順位付符号化関数	52
3.2.4.	対称関数族	61
4.	面積時間複雑度(2)	
	——境界配置の仮定を要請する場合——	70
4.1.	回路に対する面積時間積の下界	71
4.2.	関数に対する面積時間積の下界	79
4.3.	関数に対する面積時間積の上界	85
5.	面積複雑度(1) ——順序回路の場合——	93
5.1.	回路に対する面積の下界	93
5.2.	関数に対する面積の下界	101
5.3.	関数に対する面積の上界	108
6.	面積複雑度(2) ——組合せ回路の場合——	112
6.1.	入出力位置の指定と回路面積の関係	112
6.2.	乗算, 除算, ソーティングの面積下界	123
6.2.1.	乗算, 除算	124
6.2.2.	ソーティング	129

6.3. 回路を凸領域に埋め込む場合の面積下界	132
6.3.1. シフト関数, ソーティング	132
6.3.2. 復号化関数, 符号化関数	135
7. 結論	142
謝辞	145
文献	146
付録	150

1. 緒 論

半導体技術、集積回路技術の飛躍的な進歩により、数十万素子以上を一つのチップに集積する超大規模集積回路 (Very Large Scale Integrated Circuit: VLSI) が実現できるようになってきている。このような現状に応じて、VLSIに適した回路の設計手法、およびVLSI向けハードウェアアルゴリズムの開発に関する研究がさかんに行なわれている⁽³⁾。

VLSIを設計する場合、与えられた機能仕様を満たすだけでなく、費用、計算速度、電力、体積、テストの容易さなどいくつかの評価項目に対して最適な設計が望まれている。しかしながら、これらの評価項目にはトレードオフがありすべての項目を最適にすることは一般には困難である。VLSI設計時に、どの評価項目を用いるかはそれ自体重要な問題であり、また、それぞれの評価項目に対して定量的に解析できる理論的モデルの確立が重要となる。

Thompson, Brent, Kungは、VLSIを評価するために、回路の面積 A と計算時間 T を評価するVLSIの数学的モデルを提案し、評価尺度として面積時間積 AT^α ($0 \leq \alpha \leq 2$)を用いている⁽¹⁾⁽²⁾。 AT^α というときの α は面積に対して計算時間をどの程度重く見積るかの重みである。

従来の論理設計においては素子数が評価尺度として用いられていた。VLSIにおいては、回路の規模は回路内部に含まれる素子数だけでなく、素子間の配線や入出力端子にも依存する。面積は歩留り、ひいては費用などに直接的に影響するため、一つの重要な評価項目になり得る。また、回路の計算時間は、その回路の性能を抽象化したものであり、重要な評価基準である。一般に、面積

Aと計算時間Tの間には、一方を減らそうとすると一方が増えるというトレードオフがある。従って、それらの積 AT^α は、回路の総合的な評価となり工学的にバランスの良い評価基準と考えられる。また、AとTで回路を評価する限り、積の形 AT^α で評価すれば充分であることが Lipton, Sedgewick によって証明されている⁽¹²⁾。すなわち、AとTの評価関数 $u(A, T)$ が、

(i) $u(A, T)$ は A, T に関して単調である。

すなわち、 $A \leq A', T \leq T'$ ならば、

$$u(A, T) \leq u(A', T')$$

(ii) AとTの測定単位を変えても比例定数が変わるだけである。すなわち、

$$u(kA, lT) = g(k, l)u(A, T)$$

となる g が存在する。

という2つの条件を満たすものとするとき、

$$u(A, T) = \alpha_0 A^{\alpha_1} T^{\alpha_2} \quad (\alpha_0, \alpha_1, \alpha_2 \text{ はある定数})$$

の形しかありえないことが示されている。

以下では、Thompson, Brent, Kung に始まる VLSI の面積時間複雑度に関する研究の概説を行う。

1979年に Thompson は VLSI を面積と計算時間で評価するための数学的モデルを提案し、離散フーリエ変換 (DFT)、ソーティングを実現する回路の面積時間積 AT^α ($0 \leq \alpha \leq 2$) の上下界を示した⁽²⁰⁾。Thompson の VLSI モデルは、次のようなものである。配線は単位幅 (例えば、最小加工線幅 λ ⁽¹³⁾) をもつ。配線は高々2本が一点で交差できる。各素子は方眼上の一つのマス目の中におく。配線は水平、垂直方向におかれ、一つの方眼のマス目内には同方向の配線は2つ以上おくことはできない。Thompson は、回路をグラフとしてとらえ、このモデルのもとで、グラフを埋め込むためには、そのグラフの最小

2分割幅⁽²⁰⁾の2乗に比例する面積が必要であることを示した。この結果と DFT やソーティングを実現するために、入力から出力へ伝達しなければならない情報量を求めることにより、これらの関数を実現する回路の面積時間積の下界を得ている。また、この下界をほぼ達成する DFT, ソーティングを実現する回路構成を示している。

1979年に Brent, Kung は Thompson と類似の VLSI モデルを提案し、乗算器の面積時間積 AT^2 が少なくとも入力ビット数の2乗に比例すること、その下界をほぼ達成する回路構成、面積時間積をオーダ的に小さくする加算器の回路構成をそれぞれ示した⁽²⁾。Brent, Kung のモデルは回路を凸平面領域に埋め込むとし、凸領域の性質乗算を計算するために伝達しなければならない情報量を用いて下界を証明している。加算器については、1980年に Johnson によって Brent, Kung の構成がオーダ的に最良であることが証明された⁽⁸⁾。また、乗算器については、Brent, Kung の示した下界を達成する構成が1980年に Preparata によって示された⁽¹⁶⁾。1981年に Savage は、Thompson, Brent, Kung の両方の VLSI モデルの上で、 n 次正方行列の積を実現する回路の面積時間積 AT^2 がいずれの場合も、少なくとも n^4 に比例することを証明した⁽¹⁸⁾。また、この下界も漸近的に最良である⁽¹⁰⁾。

1981年に、和田、萩原、都倉は、一般の m 入力 m 出力論理関数に対する面積時間積 AT^α ($\alpha \geq 1$) の下界を示した⁽²³⁾。この下界は Thompson の結果に対して相補的なものであり、いくつかの基本的な関数についてこの下界を達成する回路構成が示されている。

Brent, Kung は、乗算器の面積時間積の下界を導出する過程で、乗算器の面積の下界を求めている⁽¹⁾。面積の評価は、 AT^α において $\alpha = 0$ としたものである。面積

下界は、計算時間を考慮しないとき、ある関数を実現するためにどれ程面積が必要になるかを表し基本的に重要である。1982年に、和田、萩原、都倉は n 入力 m 出力論理関数を実現する回路の面積下界を評価する手法を示し、その面積下界を達成する関数が存在することを示している⁽²⁹⁾。

Thompson, Brent, KungによるVLSIモデルが提案されて以来、そのモデルのもとでいろいろな関数を実現する回路の面積時間複雑度が議論される一方、より現実的な仮定を付加した変形モデルの上で、その仮定が面積などにどのように影響を与えるかについても研究されている。そのうちの一つは入出力端子は回路の境界上におく（この仮定を境界配置の仮定と呼ぶ）というものである。

Thompson, Brent, Kungのモデルでは、入出力端子は回路内の任意の場所におくことができる。ところが、VLSIでは、各種機能をブロックとして実現し、それらのブロックを接続することによりさらに大きな機能を実現する。この場合、各ブロック間の通信はブロックの境界を通して行われることが多いので、境界配置はより現実的な仮定と考えられる。Brent, Kungは、 n 個の葉節点をもつ完全二分木を葉節点と根が凸領域の境界上にあることを要請した場合、少なくとも $n \log n$ に比例する面積が必要となることを証明した⁽³⁾。また、 n 入力 m 出力論理関数を実現する回路の面積時間積の下界が境界配置の仮定に真に依存すること⁽²⁷⁾、境界配置の仮定を付加したVLSIモデルのもとで、組合せ回路、ソーティングなどを実現する回路の複雑さが議論されている⁽⁴⁾⁽³¹⁾⁽³⁴⁾。

本論文は、関連発表論文[1]～[10]として公表したVLSIの面積時間複雑度および面積複雑度に関する研究をまとめたものである。

第2章の本論文で用いるVLSIモデルと諸定義に続き、第3章、第4章では、一般の自明でない n 入力 m 出力論理関数(以下、 (n, m) -関数と略す)を実現する回路の面積時間複雑度の上下界が述べられている。

まず、第3章では、Brent, Kung類似のVLSIモデルに基づき、境界配置の仮定を要請しない場合、 (n, m) -関数族に対して、面積時間積 AT^α ($\alpha \neq 1$) が少なくとも、 $\max(n, m) [\max(\log N, \log M)]^{\alpha-1}$ に比例することが証明される。ここで、 N は N_i を i 番目 ($1 \leq i \leq m$) の出力変数が真に依存する入力数としたときの N_1, \dots, N_m の最大値であり、 M は M_j を j 番目 ($1 \leq j \leq n$) の入力変数が真に依存する出力数としたときの M_1, \dots, M_n の最大値である。

Thompsonらの面積時間積の下界はDFT, 乗算に対しては良い下界となるが、復号化関数, 符号化関数などに対しては弱い下界になる。本論文の結果は比較的構造の単純な関数に対しては良い下界を与える。

第3章の後半部では、上記の下界を達成する関数が存在することが示されている。例えば、 $(n, 2^n)$ -復号化関数を実現する回路の面積時間積 AT^α ($\alpha \neq 1$) は、上記の結果から少なくとも $2^n \cdot n^{\alpha-1}$ に比例するが、同じオーダーで実際に構成できる。以下、符号化関数, 優先順位付符号化関数 (priority encode function), 対称関数族など基本的かつ重要な関数に対して下界を達成するための回路構成を示す。

第4章では、第3章で取り扱った (n, m) -関数を実現する回路の面積時間積が境界配置の仮定に真に依存することが示される。すなわち、 (n, m) -関数族に対して、それらを実現する回路の面積時間積 AT , AT^α ($\alpha \geq 2$) がそれぞれ、少なくとも

$\max(n, m)$, $\max(n, m) \cdot \max(\log^\alpha N / \log \log N, \log^\alpha M / \log \log M)$

 に比例することが証明される。このとき、境界配置の仮定により、 AT^α ($\alpha \geq 2$) の下界が $\max(\log N / \log \log N, \log M / \log \log M)$ だけ上がる。また、この下界をほぼ達成する関数が存在することを、比較関数、復号化関数などを例として示す。このとき、第3章で得られた回路構成を単に境界配置になるように配置しなおしただけではここで得られた下界とはギャップがあるので、そのギャップをうめるために回路が再構成される。

一般に、回路面積と計算時間にはトレードオフがあり回路の面積は計算時間に依存する。また、計算時間は入力の与え方に関係する。例えば、 n 入力論理積関数は入力を並列に与える n 個の葉節点をもつ完全二分木の回路で実現すると $\log n$ 時間で計算できる。この回路は n 個の入力端子をもつので、 n に比例する面積を必要とする。また、境界配置の仮定を要請すると $n \log n$ に比例する面積を必要とする⁽³⁾。ところが、高速の計算時間を必要としなければ、直列に入力する回路で定数面積で実現できる。このように、 n 入力論理積関数や2つの n ビット2進数加算に対してはいずれも定数面積で実現できる。一方、2つの n ビット2進数乗算は、どのような入力の与え方をしても(計算時間を考慮しなくても)、回路は n に比例する面積が必要となる⁽¹⁾。理論的には、ある関数を実現するために、回路面積はどの程度必要となるかが重要となる。

第5章では、 (n, m) -関数を実現する回路の面積下界を導出する手法について述べ、その結果を用いて、加算、復号化関数、符号化関数などを実現する回路の面積下界が示される。例えば、 $(n, 2^n)$ -復号化関数、 $(n, \log n)$ -符号化関数に対しては、それぞれ、 n , $\log n$ に比例し

た回路面積が必要となる。また、これらの関数については、ここで求めた下界のオーダーで実現できることを示す。ここで示される実現は、いずれも境界配置の仮定を満たしている。従って、計算時間を考慮せず、面積だけに着目した場合は、境界配置の仮定を要請しても下界は変化しない。

第3章～第5章では、組合せ回路で実現される関数についても順序回路として実現することを許して、広実現方式内での面積時間複雑度および面積複雑度を議論するが、第6章では、同じVLSIモデルのもとで、実現を組合せ回路に限定したときの面積複雑度を考察する。

第6章では、境界配置の仮定を付加したモデルのもとで、 (n, m) -関数を実現する組合せ回路を埋め込むのに必要な面積を評価する手法が述べられている。その結果を用いると、乗算、除算、リーディングを組合せ回路で実現するとき、入力数の2乗に比例した面積が必要となることが示される。Lipton, Tarjan は平面グラフの分割定理 (Planar Separator Theorem) を用いて、2つの n ビット2進数乗算を実現するのに n^2 に比例した面積が必要となることを示しているが⁽¹¹⁾、本論文の手法を用いても同様の結果が示される。

ここで得られる面積下界は、Brent, Kung らが仮定しているように、回路が埋め込まれる領域は凸領域であることを必要としない。すなわち、乗算、リーディングなどに対する面積下界は境界配置の仮定さえ満たせば成立する。また、ここでは、凸領域と境界配置の仮定の関係について議論され、乗算、除算、リーディングについては、どちらか一方の仮定を満たせば同様の面積下界が成立することが示される。さらに、リーディング、復号化関数などを実現する組合せ回路に対して、両方の仮定を

おいたとき、従来の結果とは異なる面積下界が得られる
ことが示される。

2. VLSIモデルと諸定義

本章では、面積時間複雑度および面積複雑度を議論するためのVLSIモデルと諸定義をまとめている。

VLSIモデル

(A-1) 回路はゲート、1ビット記憶素子、入出力端子と導線から構成される。ゲート、1ビット記憶素子、入出力端子をPE (Processing Element) と呼ぶ。

(A-2) 回路は凸平面領域 R に埋め込まれる。

現在の集積回路はウェハの上に平面的に構成される。 R はチップの全体でもよいし、その一部でもよい。回路を凸領域に埋め込むという仮定は、第4章の下界の導出に必要となる。第3章、第5章では凸領域である必要はなく閉領域であればよい。また、第6章ではこの仮定をおく場合とおかない場合でどのようになるかについて議論する。

(A-3) 導線は最小幅 $\lambda (>0)$ をもつ。また、導線間の間隔も λ 以上と仮定する。各PEは少なくとも λ^2 の面積をもつ。

(A-4) PE同志、およびPEと導線は重ならない。 R 中のどの点においても、高々 $\nu (\geq 2)$ 本の導線が交差しうる。

(A-5) 1ビットのデータを導線を通して転送するの

に $\tau (> 0)$ 時間必要とする。PE の遅延時間も τ に含める。また、 τ は導線の長さによらず一定とする。

この一定遅延は、導線の長さに見合ったドライバを用いることによって達成される。そのためドライバの面積の増大は導線の占める面積に比べてゆるやかである⁽⁴⁾。

(A-6) 各入力データは回路に一度だけ与えられる。

すなわち、回路の外部に入力値を保持する記憶を想定しない。その値を複数回必要とするなら、内部で記憶しなければならない。

(A-7) 入力をどの入力端子にどの時刻に印加するか (入力のわりあて)、どの出力端子にどの時刻に出てくる信号をどの出力値とみなすか (出力のわりあて) は、入力のデータによらず固定されている。

この仮定は固定プログラム方式と呼ばれることがある。入力データの値によって、これら入出力のわりあてが変る回路は除外して考える。

(A-8) 各 PE は高々 $d (\geq 2)$ の入力線と高々 $d' (\geq 2)$ の出力線をもつ。ここで、 d, d' はある定数とする。また、入力端子の入力線数と出力端子の出力線数は 0 とする。

(A-9) 入出力端子は R の境界上におかれる。

この仮定を境界配置の仮定と呼ぶ。

この VLSI モデルは，入出力線数の制限 (A-8)，境界配置の仮定 (A-9) を除いて，Brent, Kung のモデルと基本的に同じである⁽²⁾。また，Chazelle, Monier⁽⁴⁾，安浦，矢島⁽³⁴⁾ のモデルは境界配置の仮定をおいている。

VLSI 回路 C 内の各 PE はそれぞれ識別子をもつとする。また， C 内の入力端子の集合，出力端子の集合をそれぞれ $I(C)$ ， $O(C)$ と表す。

回路 C に対する回路グラフ $G(C) = (V, E)$ は次の条件を満たすラベル付有向グラフである。

(1) $G(C)$ の節点は C の各 PE に対応し，PE の識別子がラベルとしてつけられている。

(2) $G(C)$ の有向辺は C の各導線に対応し，辺の向きは C でその導線を流れるデータの方に一致する。

$G(C)$ において， $I(C)$ ， $O(C)$ に対応する節点をそれぞれ入力節点，出力節点と呼ぶ。

回路 C に対して， C を埋め込む領域 R の面積を $A(C)$ と表し， C に入力を印加しはじめたから出力が出終わるまでの時間を $T(C)$ と表す。 C が明らかなる場合は，それぞれ A ， T と表す。また，回路グラフ $G(C)$ の面積というときは， $A(C)$ を意味するものとし， $\text{area}(G(C))$ と表す。 $G(C)$ の部分グラフ G' に対しても同様に面積 $\text{area}(G')$ を定義する。すなわち， G' の節点を PE，有向辺を導線と考え，VLSI モデルのもとで埋め込んだときの面積とする。

以下では，ことわりのない限り， λ^2 を単位面積とし， τ を単位時間とする。

次に、本論文で用いる記法、取り扱う関数を定義する。

記法

f, g を非負整数から非負数への関数とする。ある正数 c , 非負整数 m_0 が存在して、すべての $m \geq m_0$ に対して、 $f(m) \geq cg(m)$ が成立つとき、 $f(m) = \Omega(g(m))$ と表し、 $f(m)$ の下界 (lower bound) は $g(m)$ であるという。ある正数 c' , 非負整数 m'_0 が存在して、すべての $m \geq m'_0$ に対して、 $f(m) \leq c'g(m)$ が成立つとき、 $f(m) = O(g(m))$ と表し、 $f(m)$ の上界 (upper bound) は $g(m)$ であるという。また、 $f(m) = \Omega(g(m))$, かつ、 $f(m) = O(g(m))$ のとき、 $f(m) = \Theta(g(m))$ と表す。

p を任意の実数とする。 $p \leq p_c < p+1$ を満たす整数 p_c を $\lceil p \rceil$ と表す。また、 $p-1 < p_f \leq p$ を満たす整数 p_f を $\lfloor p \rfloor$ と表す。

$\log x$ は $\log_2 x$ を表すものとする。すなわち、対数の底が 2 であるとき、対数の底を省略する。また、 $\log^2 x$ は、 $(\log x)^2$ を意味するものとする。

変数系列 $Z = (z_1, \dots, z_k)$ に対して、 $(z_{i_1}, \dots, z_{i_j})$ ($1 \leq i_1 < \dots < i_j \leq k$) を Z の部分系列という。また、 $S(Z)$ で変数の集合 $\{z_1, \dots, z_k\}$ を表す。

Z の部分系列 Z_1, Z_2 に対して、 $S(Z_1) \cap S(Z_2) = \emptyset$, かつ $S(Z_1) \cup S(Z_2) = S(Z)$ が成立つとき、 Z_2 を $Z - Z_1$ で表す。

取り扱う関数

$\{0, 1\}^m$ で 0 と 1 の m 組の集合を表す。まず, m 入力 m 出力論理関数 $f: \{0, 1\}^m \rightarrow \{0, 1\}^m$ (以下, (m, m) -関数と略す) のクラスを定義する。

(m, m) -関数を

$$(y_1, \dots, y_m) = f(x_1, \dots, x_m) = (f_1(x_1, \dots, x_m), \dots, f_m(x_1, \dots, x_m))$$

と表す。ここで $f_i: \{0, 1\}^m \rightarrow \{0, 1\}$ ($1 \leq i \leq m$) とする。

$(m, 1)$ -関数を $h(x_1, \dots, x_m)$ とする。変数 x_i に対して

$$h(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_m) \neq h(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_m)$$

となる $m-1$ 組 $(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_m)$ が存在するとき, x_i は h に対して冗長でないという。

(m, m) -関数 f に対して, N_f, M_f を次のように定義する。

N_{f_i} は f_i に対して冗長でない入力変数の個数とし,

$$N_f = \max(N_{f_1}, \dots, N_{f_m})$$

と定義する。

M_j は x_j が冗長でない出力関数の個数とし,

$$M_f = \max(M_1, \dots, M_m)$$

と定義する.

関数のクラス $F(n, m)$ は,

(i) 各 i ($1 \leq i \leq m$) に対して $N_{fi} > 0$,

(ii) 各 j ($1 \leq j \leq m$) に対して $M_j > 0$

を満たす (n, m) -関数 f の集合とする.

$F(n, m)$ に属する関数 f のうち, $N_f = N$, かつ $M_f = M$ となる (n, m) -関数全体を $F(n, N, m, M)$ と定義する.

$F(n, N, m, M)$ に属する関数 f を (n, N, m, M) -関数という. また, (n, m) -関数 f を実現する回路を C_f と表す.

以下, 本論文で議論する具体的な関数を定義する.

(1) $(n, 2^n)$ -復号化関数

$(n, 2^n)$ -復号化関数は, 入力変数を 2 進数 $x_{n-1} \cdots x_0$ とみなし, その値が i ($0 \leq i \leq 2^n - 1$) のとき, 出力変数 y_i が 1 となり, 他の出力変数 y_j ($j \neq i$) が 0 となる関数である.

(2) $(n, \lceil \log n \rceil)$ -符号化関数

$(n, \lceil \log n \rceil)$ -符号化関数は, 入力変数 x_0, \dots, x_{n-1} のうち一つだけの変数が 1 になり, その添字の 2 進数表現 $y_{\lceil \log n \rceil - 1} \cdots y_0$ を出力する関数である.

(3) $(n, \lceil \log n \rceil)$ -優先順位付符号化関数

$(n, \lceil \log n \rceil)$ -優先順位付符号化関数 (priority encode function) は, 入力変数 x_0, \dots, x_{n-1} のなかで 1 になる変数のうち, 最大の添字の 2 進数表現 $y_{\lceil \log n \rceil - 1} \cdots y_0$

を出力する関数である。

(4) $(n, \lceil \log(n+1) \rceil)$ -重み関数

$(n, \lceil \log(n+1) \rceil)$ -重み関数は、入力変数 x_0, \dots, x_{n-1} に対し、変数の値が 1 に等しいものの個数の 2 進数表現 $y_{\lceil \log(n+1) \rceil - 1} \dots y_0$ を出力する関数である。

(5) $(n, 1)$ -対称関数族

系列 $(1, \dots, n)$ の置換を $(p(1), \dots, p(n))$ で表す。 $(n, 1)$ -関数 h が、任意の $(a_1, \dots, a_n) \in \{0, 1\}^n$ 、任意の置換 $(p(1), \dots, p(n))$ に対して、

$$h(a_1, \dots, a_n) = h(a_{p(1)}, \dots, a_{p(n)})$$

が成立つとき、 h は対称関数という。すべての変数が冗長でない $(n, 1)$ -対称関数の集合を S_n で表す。

(6) 2 つの n ビット 2 進数の比較

2 つの n ビット 2 進数 $x_{n-1} \dots x_0$, $w_{n-1} \dots w_0$ の値をそれぞれ p , q とし、

$p > q$ のとき、

$$(y_1, y_2) = (1, 0),$$

$p = q$ のとき、

$$(y_1, y_2) = (0, 0)$$

(または、 $(1, 1)$)

$p < q$ のとき、

$$(y_1, y_2) = (0, 1)$$

を出力する関数である。この関数は $(2n, 2)$ -関数である。

(7) k 個の n ビット 2進数の加算

k 個の n ビット 2進数 $x_{(k-1)i} \cdots x_{0i}$ ($1 \leq i \leq k$) の和の 2進数表現 $y_{m+\lceil \log k \rceil - 1} \cdots y_0$ を出力する関数である。この関数は $(kn, m + \lceil \log k \rceil)$ -関数である。

(8) 2つの n ビット 2進数の乗算

2つの n ビット 2進数 $x_{n-1} \cdots x_0, w_{n-1} \cdots w_0$ の積の 2進数表現 $y_{2n-1} \cdots y_0$ を出力する関数である。この関数は $(2n, 2n)$ -関数である。

(9) $2n$ ビット 2進数の n ビット 2進数による除算

$2n$ ビット 2進数 $x_{2n-1} \cdots x_0$ を n ビット 2進数 $w_{n-1} \cdots w_0$ で割った商の 2進数表現 $y_{2n-1} \cdots y_0$ を出力する関数である。この関数は $(3n, 2n)$ -関数である。

(10) k ビット 2進数 n 個のリーディング

n 個の k ビット 2進数 $x_{(k-1)i} \cdots x_{0i}$ ($1 \leq i \leq n$) を昇順 (または降順) に並べかえて, j 番目に小さい (または大きい) ものを $y_{(k-1)j} \cdots y_{0j}$ ($1 \leq j \leq n$) として出力する関数である。この関数は, (kn, kn) -関数であり, $k \geq \lceil \log n \rceil$ と仮定する。

3. 面積時間複雑度 (1)

——境界配置の仮定を要請しない場合——

本章では、境界配置の仮定を要請しない VLSI モデルのもとで $F(n, N, m, M)$ に属する関数を実現する回路の面積時間複雑度を議論する。

3.1. 面積時間積 AT^{α} の下界

仮定 (A-1), (A-3) ~ (A-8) を満足する VLSI 回路を C とする。 C に対する回路仕様 $S_C = (f, \sigma_{\pm}, \sigma_o)$ を次のように定める。

(1) f は $F(m, m)$ に属する関数であり

$$(y_1, \dots, y_m) = f(x_1, \dots, x_n)$$

とする。

(2) σ_{\pm} は各入力変数 x_i が C のどの入力端子にどの時刻に与えられるかを指定する。 x_i が入力端子 p に、時刻 t に与えられるとき、 $\sigma_{\pm}(x_i) = (p, t)$ と表す。

(3) σ_o は各出力変数 y_j が C のどの出力端子からどの時刻に得られるかを指定する。 y_j が出力端子 q に、時刻 t に得られるとき、 $\sigma_o(y_j) = (q, t)$ と表す。

σ_{\pm} , σ_o をそれぞれ入力指定, 出力指定という。また、仮定 (A-7) によって入力指定, 出力指定は関数になる。

回路仕様 $(f, \sigma_{\pm}, \sigma_o)$ を満足する回路の集合を、 $C[f, \sigma_{\pm}, \sigma_o]$ で表し、 $C[f]$ を f を実現する回路全体の集合を表す。すなわち、

$$C[f] = \bigcup_{\sigma_{\pm}, \sigma_o} C[f, \sigma_{\pm}, \sigma_o].$$

回路 C の回路グラフを $G(C)$ とする。本章で対象とする

回路は組合せ回路とは限らない。組合せ回路として実現できる関数でも、順序回路として実現することも許して広い実現方式内で考える。従って、 C は記憶素子を含む場合もあり、またループをもつ場合もある。このとき、 $G(C)$ にはサイクルが含まれる。

[補題 3.1] f を $F(n, N, m, M)$ に属する関数とし、 $C \in C[f]$ とする。このとき、

$$T = \Omega(\max(\lceil \log_{d+1} N \rceil, \lceil \log_{d'+1} M \rceil))$$
 が成立つ。

(証明) C の回路仕様を $S_C = (f, \sigma_x, \sigma_y)$ とする。一般性を失うことなく、 $\sigma_x(x_i)$ の時間成分の最小値を 0、 $\sigma_y(y_j)$ の時間成分の最大値を T と仮定できる。 C と S_C から、次のようなグラフ $E(C, S_C)$ を構成する。

$E(C, S_C) = (U, H)$ はラベル付有向グラフで、 U の各節点は PE の名前と時刻の対をラベルとしてもつ。 U は最初、 C の入力端子に対応する節点を含み、各節点は入力指定 σ_x によるラベルがつけられる。 H の初期値は空集合である。以下、次の規則に従って、 $E(C, S_C)$ を構成してゆく。

(R-1) $G(C)$ において、節点 v に隣接するすべての節点を u_1, \dots, u_j とし、 u_1, \dots, u_j, v のラベルをそれぞれ a_1, \dots, a_j, b とする。 U に $(a_i, t), \dots, (a_j, t)$ なるラベルをもつ節点が含まれているならば U にラベル $(b, t+1)$ をもつ節点を加える。また、 $(a_1, t), \dots, (a_j, t)$ なるラベルをもつ節点から、ラベル $(b, t+1)$ をもつ節点への有向辺を H に加える(図 3.1(a)).

また、ラベル (a_k, t) をもつ節点が C において記憶素子に対応するならば、ラベル $(a_k, t+1)$ をもつ節点を U に、ラベル (a_k, t) をもつ節点からラベル $(a_k, t+1)$ をもつ節点への有向辺を H にそれぞれ加える (図3.1 (b)).

この規則は、 C において時刻 t での a_1, \dots, a_j のデータを用いて、時刻 $t+1$ での各の出力が決まることに対応している。また、 a_k が記憶素子のとき、 a_k の t から $t+1$ への状態遷移に対応している。

f の計算は、時刻 T に終了するので T より大きい時刻成分をもつラベルをもつ節点を U に含める必要はない。

上の規則 (R-1) とこの除外規則を繰り返して用いて、それ以上新しい節点が U に含められなくなれば $E(C, S_c)$ の完成である。($G(C)$ の節点数を n とすると、 $E(C, S_c)$ は高々 $n(T+1)$ 個の節点しかもちえないので、この作業は有限のステップで終る。)

注1) 上の $E(C, S_c)$ の構成では、遅延素子について特に説明しなかったが、1単位時間 (τ 時間) 遅延素子は $G(C)$ において、その素子を一つの有向辺に置き換えて考えればよい (図3.2)。

注2) C がフリックフロップなどの遅延素子を含む場合、入力印加開始前にそれらをリセットすることがある。その場合、遅延素子 α の初期値は決まっているから、入力端子と同様に $(\alpha, 0)$ を U に含まれているとして先の規則を適用してゆく。

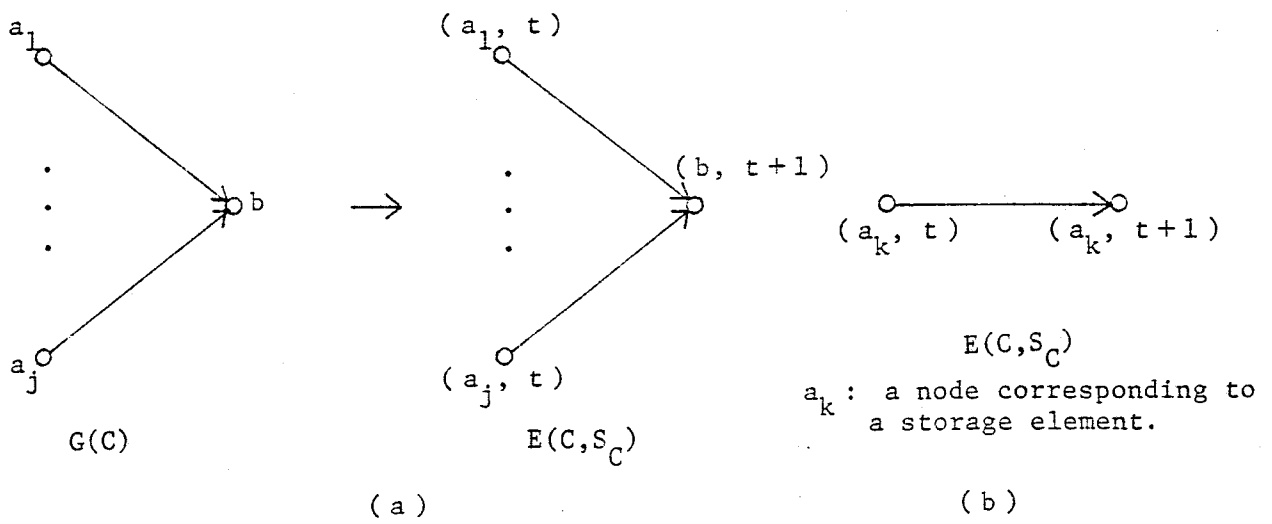


図3.1 $G(C)$ から $E(C, S_C)$ への変換.

Fig.3.1 Transformation rule from $G(C)$ to $E(C, S_C)$.

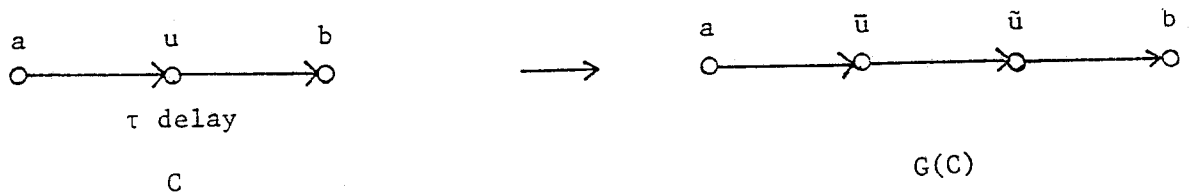


図3.2 遅延素子を含む場合.

Fig.3.2 A unit-time delay is expressed as an edge in $G(C)$.

例として、 n ビット加算器 ($n=2$ とする) C_{add} について $E(C_{add}, S_{C_{add}})$ を構成する。図3.3 (a) に2ビット加算器 C_{add} を示す。ここで、 FA_i ($i=0, 1$) は全加算器、 A_i , B_i ($i=0, 1$) は記憶をもつ入力端子、 D はキャリ入力端子、 F_i ($i=0, 1, 2$) は出力端子をそれぞれ表す。 C_{add} に対応する回路グラフ $G(C_{add})$ 、および回路仕様 $S_{C_{add}}$ を図3.3 (b) に示す。図3.3 (c) には $E(C_{add}, S_{C_{add}})$ を示す。

さて、この $E(C, S_c)$ は次の性質をもつ。

(p-1) $G(C)$ にどのようなサイクルが含まれていても $E(C, S_c)$ にはサイクルは含まれない。

仮定(A-5)で、導線を単位時間でデータが転送されるとして、有向辺を一つ通るごとに時刻成分が1増すことによる。

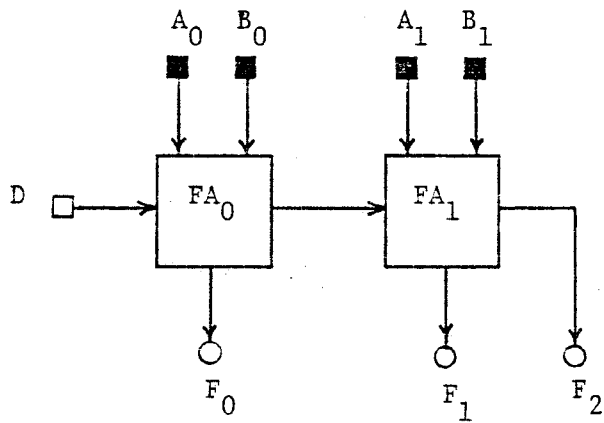
(p-2) $E(C, S_c)$ の各節点の入次数、出次数はそれぞれ $d+1$, $d'+1$ 以下である。

仮定(A-8)により、 $G(C)$ の各節点の入次数、出次数はそれぞれ d , d' 以下であり、 C に記憶素子が含まれる場合、入次数、出次数が1だけ増す。

f は $F(n, N, m, M)$ に属する関数ゆえ、 $E(C, S_c)$ には、入力端子 p_1, \dots, p_N をラベルにもつ N 個の異なる節点から N 本の道をもつ出力端子 q をラベルにもつような節点が存在する。性質 (p-1), (p-2) よりこれらの道の長さは少なくとも $\lceil \log_{d+1} N \rceil$ である。従って、

$$T \geq \lceil \log_{d+1} N \rceil \quad (3.1)$$

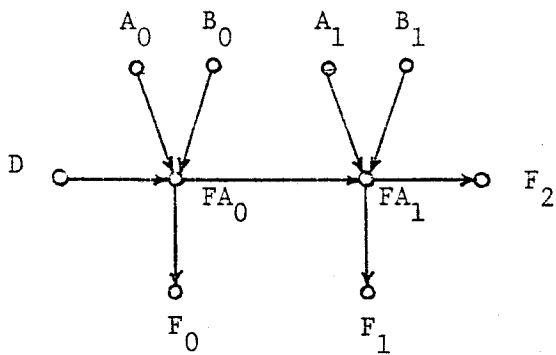
が成立つ。入力端子と出力端子の関係を入れかえると同様にして、



■ : input port with memory
 □ : carry input port
 ○ : output port

FA_i : full adder (i = 0, 1)

(a) 2-bit adder C_{add}



$$\begin{aligned}
 S_{C_{add}} &= (y_2 \cdot 2^2 + y_1 \cdot 2 + y_0 \\
 &= (x_1 + w_1) 2 \\
 &\quad + (x_0 + w_0) + v ;
 \end{aligned}$$

$$S_I(x_i) = (A_i, 0) \quad (i=0,1),$$

$$S_I(w_i) = (B_i, 0) \quad (i=0,1),$$

$$S_I(v) = (D, 0) ;$$

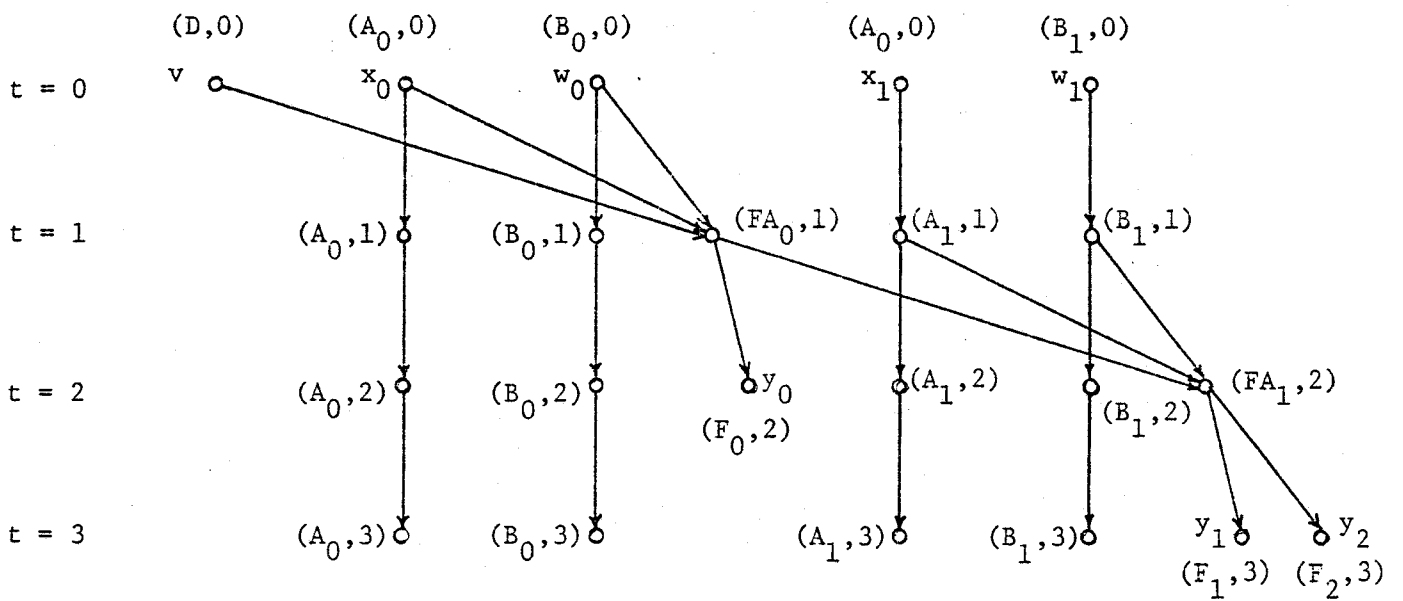
$$S_O(y_0) = (F_0, 2),$$

$$S_O(y_i) = (F_i, 3) \quad (i=1,2)$$

(b) Circuit graph $G(C_{add})$ and the circuit specification $S_{C_{add}}$.

図3.3 $E(C, S_C)$ の例.

Fig.3.3 An example of $E(C, S_C)$.



(c) $E(C_{\text{add}}, S_{C_{\text{add}}})$

図3.3 $E(C, S_c)$ の例 (続).

Fig.3.3 An example of $E(C, S_c)$ (continued).

$$T \geq \lceil \log_{d+1} M \rceil \quad (3.2)$$

が成立つ。(3.1), (3.2)より次式が成立する。

$$T = \Omega(\max(\lceil \log_{d+1} N \rceil, \lceil \log_{d+1} M \rceil)) \quad \blacksquare$$

補題3.1より, $F(n, N, m, M)$ に属する関数を実現する回路の面積時間積の下界は次の定理により示される。

[定理3.1] f を $F(n, N, m, M)$ に属する関数とし, $C \in C[f]$ とする。このとき

$$AT^\alpha = \Omega(\max(n, m) \lceil \max(\lceil \log_{d+1} N \rceil, \lceil \log_{d+1} M \rceil) \rceil^{\alpha-1})$$

($\alpha \geq 1$)が成立つ。

(証明) (n, N, m, M) -関数の定義より, C は n 個の入力値を読み込まなければ計算は終了できないので,

$$T \geq \lceil n/A \rceil$$

が成立つ。同様に, C は m 個の値を出力しなければならないので,

$$T \geq \lceil m/A \rceil$$

である。これらのことから

$$AT \geq \max(n, m) \quad (3.3)$$

が成立つ。補題3.1と(3.3)から $\alpha \geq 1$ のとき

$$AT^\alpha = AT \cdot T^{\alpha-1} \\ = \Omega(\max(n, m) [\max(\lceil \log_{d+1} N \rceil, \lceil \log_{d+1} M \rceil)]^{\alpha-1})$$

が成立つ。■

定理3.1は、 (n, N, m, M) -関数を実現する任意の回路について成立つ。従って、例えば、パイプライン方式の入出力を行う回路についても成立つ。また、定理3.1は入出力の与え方には何の制限も置いていない。しかしながら、すべての入力を同時に与えなければならぬ、あるいは、すべての出力を同時に得なければならぬ場合がある。このような入出力条件により回路の面積に制限がおかれているとき、定理3.1は次のように拡張できる。

[定理3.2] f を $F(n, N, m, M)$ に属する関数とし、 $C \in C[f]$ とする。問題の性質上、 C の面積下界が決まっており、 $A = \Omega(B)$ とすると、

$$AT^\alpha = \Omega(\max[B \cdot T_{N,M}^\alpha, n T_{N,M}^{\alpha-1}, m T_{N,M}^{\alpha-1}])$$

($\alpha \geq 1$) が成立つ。ここで、 $T_{N,M} = \max(\lceil \log_{d+1} N \rceil, \lceil \log_{d+1} M \rceil)$ である。■

3.2. 面積時間積 AT^α の上界

ここでは、いくつかの (n, N, m, M) -関数を実現する回路の面積時間複雑度の上界を議論し、定理 3.1, 3.2 の下界を達成する (n, N, m, M) -関数が存在することを示す。

表 3.1 に、本節で取り扱う関数の面積時間複雑度の上下界をまとめて示す。各関数の面積時間複雑度の下界はいずれも定理 3.2 から得られる。表 3.1 に示される (n, N, m, M) -関数の回路構成は計算時間を $O(\log NM)$ で実現している。本節で取り扱う (n, N, m, M) -関数については、計算時間 $O(\log NM)$ で実現する回路が定理 3.2 の下界を達成する有望な候補になる。従って、計算時間を $O(\log NM)$ に保存したまま、回路面積をできるだけ小さくすることが重要な問題になる。そのような構成は、関数の帰納的な構造に着目し、帰納的に面積の小さい構成を行い、さらにパイプライン入出力を用いて面積を削減することで達成される。

表 3.1 関数の面積時間複雑度.

関数 (入力数/出力数)	面積時間積 AT^α ($\alpha \geq 1$)
復号化関数 ($n/2^n$)	$\Omega(2^n \cdot n^{\alpha-1})$ $O(2^n \cdot n^{\alpha-1})$
符号化関数 ($n/\lceil \log n \rceil$)	$\Omega(n \cdot \log^{\alpha-1} n)$ $O(n \cdot \log^{\alpha-1} n)$
論理積, 論理和 排他的論理和 ($n/1$)	$\Omega(n \cdot \log^{\alpha-1} n)$ $O(n \cdot \log^{\alpha-1} n)$
2つのnビット2進数 比較 ($2n/2$)	$\Omega(n \cdot \log^{\alpha-1} n)$ $O(n \cdot \log^{\alpha-1} n)$
優先順位付符号化 関数 ($n/\lceil \log n \rceil$)	$\Omega(n \cdot \log^\alpha n)$ $O(n \cdot \log^\alpha n)$
重み関数 ($n/\lceil \log(n+1) \rceil$)	$\Omega(n \cdot \log^{\alpha-1} n)$ $O(n \cdot \log^{\alpha-1} n)$
対称関数 $S^n_{(a_1, \dots, a_k)}$ ($n/1$)	$\Omega(n \cdot \log^{\alpha-1} n)$ $O(n \cdot \log^{\alpha-1} n)$ ($k = O(n/\log^3 n)$) $O(n \cdot \log^\alpha n)$ (any k)

3.2.1. 復号化関数, 符号化関数

$(n, 2^n)$ -復号化関数を実現する回路を $(n, 2^n)$ -復号器 ($(n, 2^n)$ -decoder) という。また, $(n, \lceil \log n \rceil)$ -符号化関数を実現する回路を $(n, \lceil \log n \rceil)$ -符号器 ($(n, \lceil \log n \rceil)$ -encoder) という。いずれの関数もそれらの帰納的構造に基づいて構成することにより, それぞれ $AT^\alpha = O(2^n \cdot n^\alpha)$, $AT^\alpha = O(n \log^\alpha n)$ で実現できる。次に入出力を適当に分割しパイプライン入出力を用いることにより, 定理3.1の下界を達成できる。まず, $(n, 2^n)$ -復号器について示す。

[補題3.2] $(n, 2^n)$ -復号器は, $A = O(2^n)$, $T = O(n)$ で実現できる。

(証明) $(k, 2^k)$ -復号器を $D(k)$ で表す。 $D(k)$ の各出力は定義より次のように表される。

$$y_0 = \bar{x}_0 \bar{x}_1 \cdots \bar{x}_{k-2} \bar{x}_{k-1},$$

$$y_1 = x_0 \bar{x}_1 \cdots \bar{x}_{k-2} \bar{x}_{k-1},$$

⋮

$$y_{2^{k-2}} = \bar{x}_0 x_1 \cdots x_{k-2} x_{k-1},$$

$$y_{2^{k-1}} = x_0 x_1 \cdots x_{k-2} x_{k-1}.$$

$a_i \in \{0, 1\}$, $x_i^0 = \bar{x}_i$, $x_i^1 = x_i$ ($0 \leq i \leq k-1$) とし,

$$y_{a_0 \dots a_{k-1}}(d) = x_0^{a_0} \cdots x_{k-1}^{a_{k-1}} d$$

と定義する。 $D(k, d)$ を x_0, \dots, x_{k-1}, d を入力とし, $y_{0 \dots 0}(d), \dots, y_{1 \dots 1}(d)$ を出力とする回路とする。復号化関数の定義により $D(k) = D(k, 1)$ である。

$D(k, d)$ は次の関係

$$y_{a_0 \dots a_{k-1}}(d) = y_{a_0 \dots a_{k-2}}(x_{k-1}^{a_{k-1}} d)$$

によって、図 3.4 のように 2 つの $D(k-1, d)$ によって構成される。この回路を図 3.5 のように配置すると、

$$\begin{aligned} A &= O(2^k), \\ T &= O(k) \end{aligned}$$

である。† 従って、 $(n, 2^n)$ -復号器は $A = O(2^n)$, $T = O(n)$ で実現できる。■

補題 3.2 の $(n, 2^n)$ -復号器の構成では定理 3.1 の下界を達成しない。この下界を達成するためには、計算時間を $O(n)$ に保存したまま、面積を $O(2^n/n)$ で実現しなければならない。 $(n, 2^n)$ -復号器の出力は 2^n 個あるので、全出力を同時に出力すると、仮定 (A-3) より $\Omega(2^n)$ となる。以下の構成では、 $O(2^n/n)$ の面積にするために、 2^n 個の出力を n 組に分割し、各組の $2^n/n$ 個の出力を順次パイプラインを用いて出力する。

† 付録 A-1 参照。

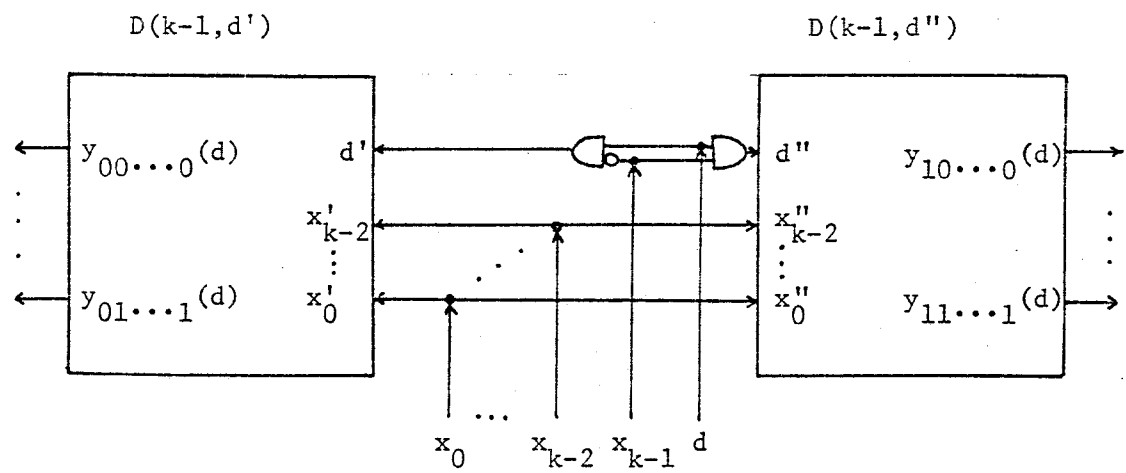


図3.4 $D(k, d)$ の帰納的な構成.

Fig.3.4 Recursive construction of $D(k, d)$.

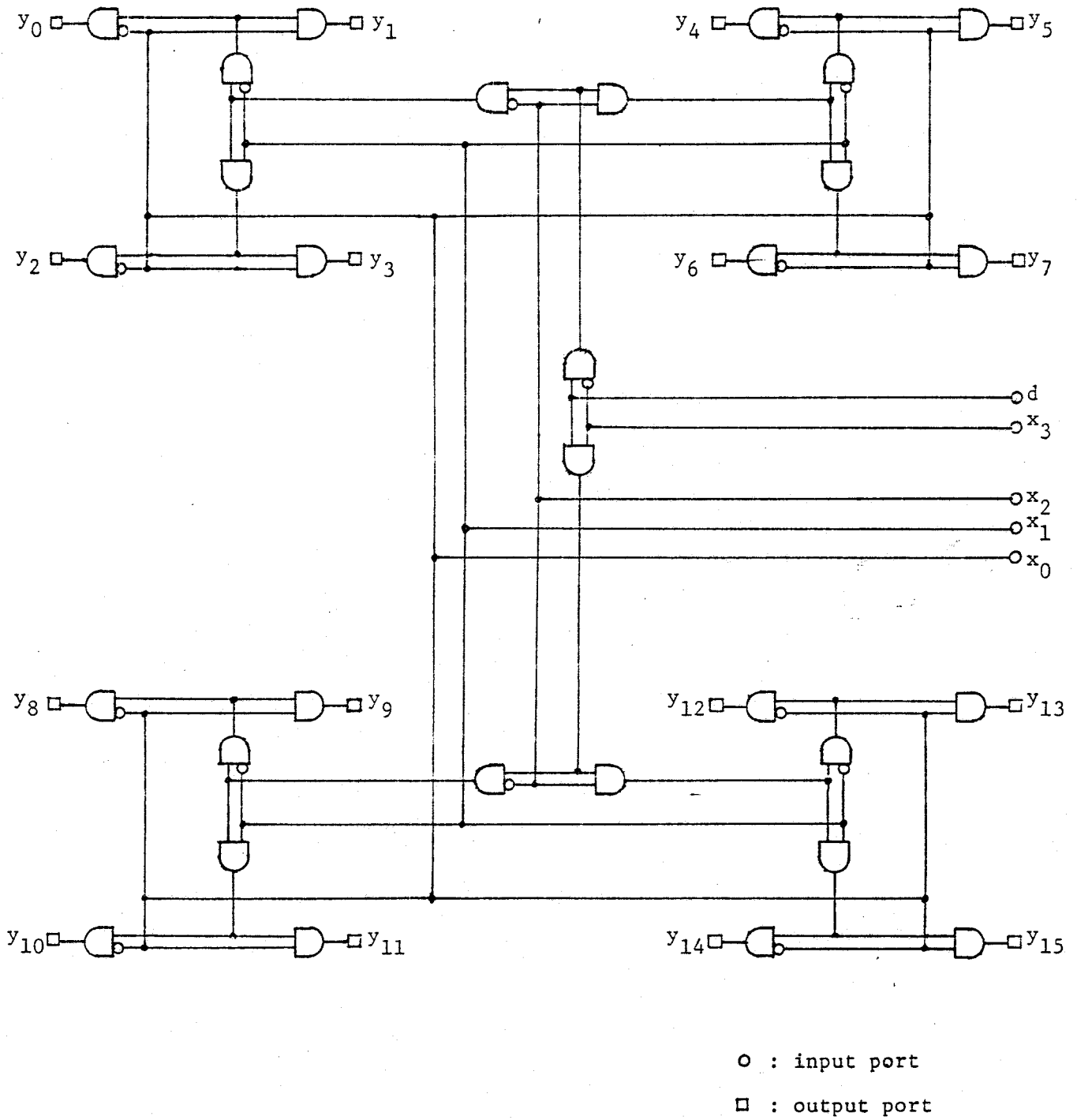


図 3.5 $D(k, d)$ の配置 ($k=4$).

Fig.3.5 Layout of $D(k, d)$ ($k=4$).

[定理 3.3] $(n, 2^n)$ -復号化関数に対して,

$$AT^\alpha = \theta(2^n \cdot n^{\alpha-1}) \quad (\alpha \neq 1)$$

が成立つ.

(証明) $w_1 + w_2 = n$ とする. $(n, 2^n)$ -復号器 $D(n)$ は, 次の関係式.

$$y_{a_0 \dots a_{n-1}}(1) = y_{a_0 \dots a_{w_2-1}}(x_{w_2}^{a_{w_2}} \dots x_{w_1+w_2-1}^{a_{w_1+w_2-1}}),$$

によって, $D(w_1, 1)$ と $D(w_2, d)$ から構成される. ここで, $x_{w_2}^{a_{w_2}} \dots x_{w_1+w_2-1}^{a_{w_1+w_2-1}}$ は $D(w_1, 1)$ の各出力に対応している.

$D(n)$ の 2^n 個の出力は, $D(w_1, 1)$ の 2^{w_1} 個の出力を, $D(w_2, d)$ の入力 d に順次読み込むことにより得られる.

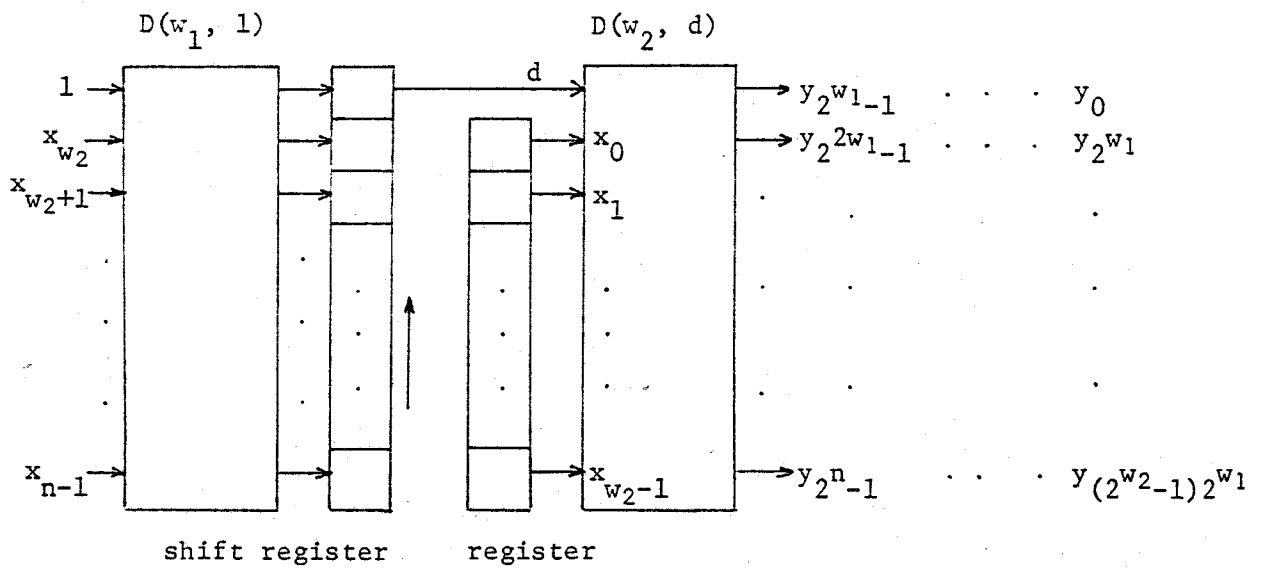
$D(n)$ の構成を図 3.6 に示す. $D(w_1, 1)$ の出力はシフトレジスタに, $D(w_2, d)$ への入力はレジスタにそれぞれ貯えらる. $D(w_1, 1)$ の各結果は 1 単位時間おくれで $D(w_2, d)$ へ入力される.

$D(n)$ の各出力の得られる時刻, および, $D(n)$ の計算時間 T は次のようになる.

$D(w_1, 1)$, $D(w_2, d)$ の計算時間をそれぞれ t_1 , t_2 とする. $D(w_1, 1)$ の最初の結果を $D(w_2, d)$ へ入力してから, 2^{w_1} 個の出力を得るまで $(t_1 + t_2 + 1)$ 単位時間かかる. そのときの 2^{w_1} 個の出力は, $D(n)$ の

$$y_0, y_{2^{w_1}}, \dots, y_{(2^{w_2-1})2^{w_1}}$$

に対応する. シフトレジスタにおかれた 2^{w_1} 個の $D(w_1, 1)$ の出力は 1 単位時間おくれで $D(w_2, d)$ へ入力されるので, 時刻 $(t_1 + t_2 + i + 1)$ ($1 \leq i \leq 2^{w_1} - 1$) に得られる $D(w_2, d)$



(The outputs of $D(w_1, 1)$ are obtained on the edge of the $D(w_1, 1)$.)

図3.6 下界を達成する $D(n)$ の構成.

Fig.3.6 Construction of $D(n)$ to attain the lower bound.

の i 番目の出力は $D(n)$ の

$$y_i, y_{i+2^{w_1}}, \dots, y_{i+(2^{w_1-1})2^{w_2}}$$

に対応する。 $D(n)$ は時刻 $(t_1 + t_2 + 2^{w_2})$ に計算を終了するので、

$$T \leq t_1 + t_2 + 2^{w_2} \quad (3.4)$$

が成立つ。

次に、 $D(n)$ の面積を解析する。図 3.7 に $D(n)$ の配置を示す ($n=6, w_1=2, w_2=4$)。 $D(w_2, d)$ の配置は、補題 3.2 の配置 (図 3.5) と同じである。 $D(w_1, 1)$ の配置は、その出力がシフトレジスタに貯えなければならないので、出力が $D(w_1, 1)$ の境界上に得られるように配置する。このとき、 $D(w_1, 1)$ の面積は $O(2^{w_1} \cdot w_1^2)$ となる[†]。一方、 $D(w_2, d)$ の面積は $O(2^{w_2})$ である。従って、 $D(n)$ の面積 A は、

$$A = O(2^{w_1} \cdot w_1^2 + 2^{w_2}) \quad (3.5)$$

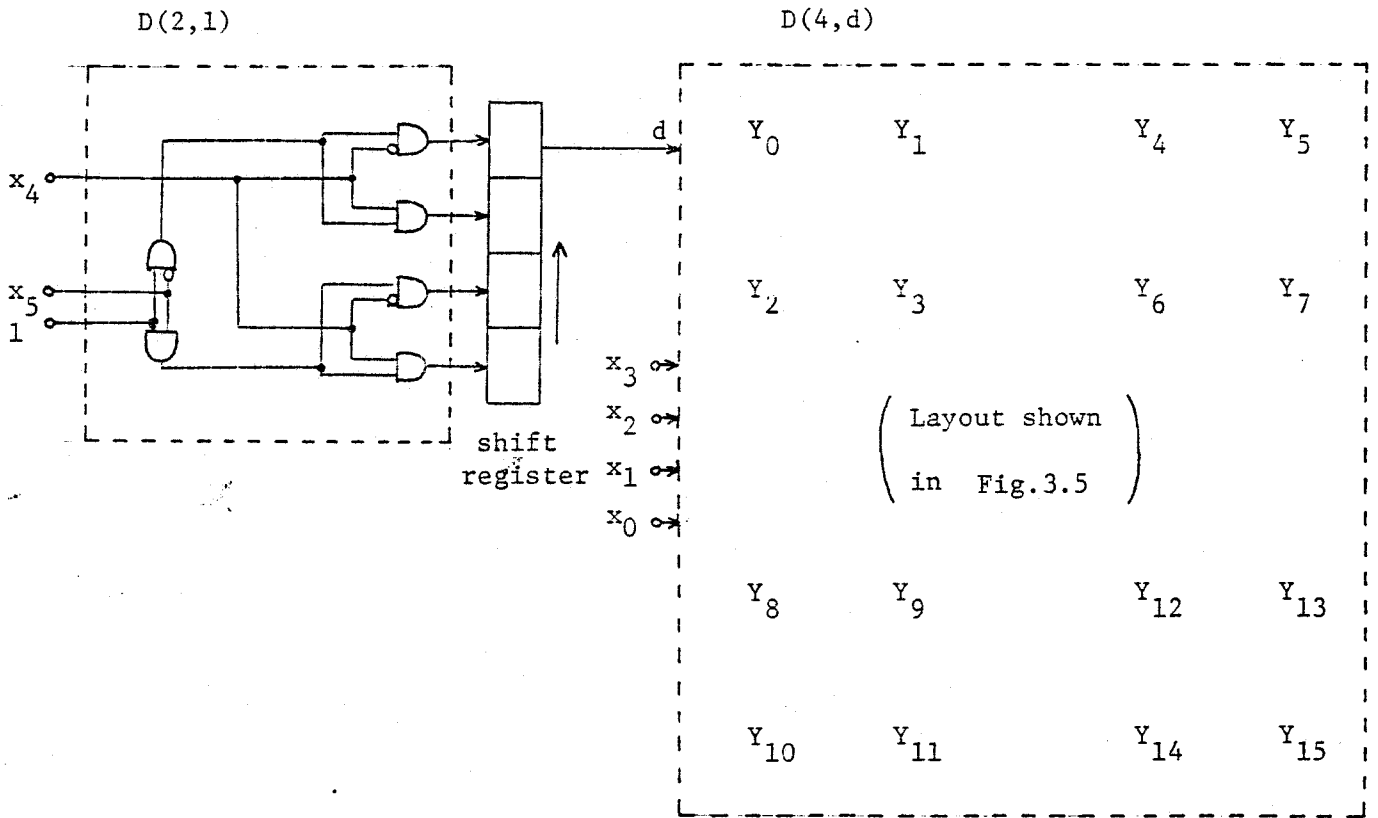
となる。また、 $D(n)$ の計算時間は (3.4) と補題 3.2 により

$$T = O(2^{w_1} + w_2) \quad (3.6)$$

が成立つ。

$$w_1 = \lceil \log n T \rceil, w_2 = n - \lceil \log n T \rceil \text{ と設定すれば, (3.5)}$$

[†] 付録 A-2 参照。



The outputs y_i , y_{i+4} , y_{i+8} and y_{i+12} are obtained from Y_i ($0 \leq i \leq 15$).

図3.7 図3.6 の $D(n)$ の配置 ($n=6$).

Fig.3.7 Layout of $D(n)$ ($n=6$).

(3.6) より,

$$\begin{aligned} A &= O(2^n/n), \\ T &= O(n) \end{aligned}$$

となる。従って、 $D(n)$ は $AT^\alpha = O(2^n \cdot n^{\alpha-1})$ ($\alpha \geq 0$)
で実現できる。また、下界は $(n, 2^n)$ -復号化関数は
 $F(n, n, 2^n, 2^n)$ に属する関数ゆえ、定理 3.1 より、
 $AT^\alpha = \Omega(2^n \cdot n^{\alpha-1})$ ($\alpha \geq 1$) が得られる。■

符号器も復号器と同様の構成法を用いて、定理3.1の下界を達成できる。

[補題3.3] $(n, \lceil \log n \rceil)$ -符号器は、 $A = O(n)$, $T = O(\log n)$ で構成できる。

(証明) 簡単のため $n = 2^k$ とする。 $(2^k, k)$ -符号器を $E(2^k)$ で表す。 $E(2^k)$ の各出力は次のように表される。

$$y_0 = x_1 + x_3 + \dots + x_{2i+1} + \dots + x_{2^{k-1}},$$

$$y_1 = x_2 + x_3 + \dots + x_{4i+2} + x_{4i+3} + \dots + x_{2^{k-1}},$$

⋮

$$y_{k-1} = x_{2^{k-1}+1} + x_{2^{k-1}+2} + \dots + x_{2^{k-1}}.$$

$E(2^k, e)$ を x_0, \dots, x_{2^k-1} を入力とし、 y_0, \dots, y_{k-1} と

$$e = x_0 + x_1 + \dots + x_{2^k-1}$$

を出力とする回路とする。符号化関数の帰納的性質から $E(2^k, e)$ は2つの $E(2^{k-1}, e)$ を用いて図3.8のように構成できる。図3.8の構成を図3.9のように配置することにより、 $E(2^k, e)$ に対して

$$\begin{aligned} A &= O(2^k), \\ T &= O(k) \end{aligned}$$

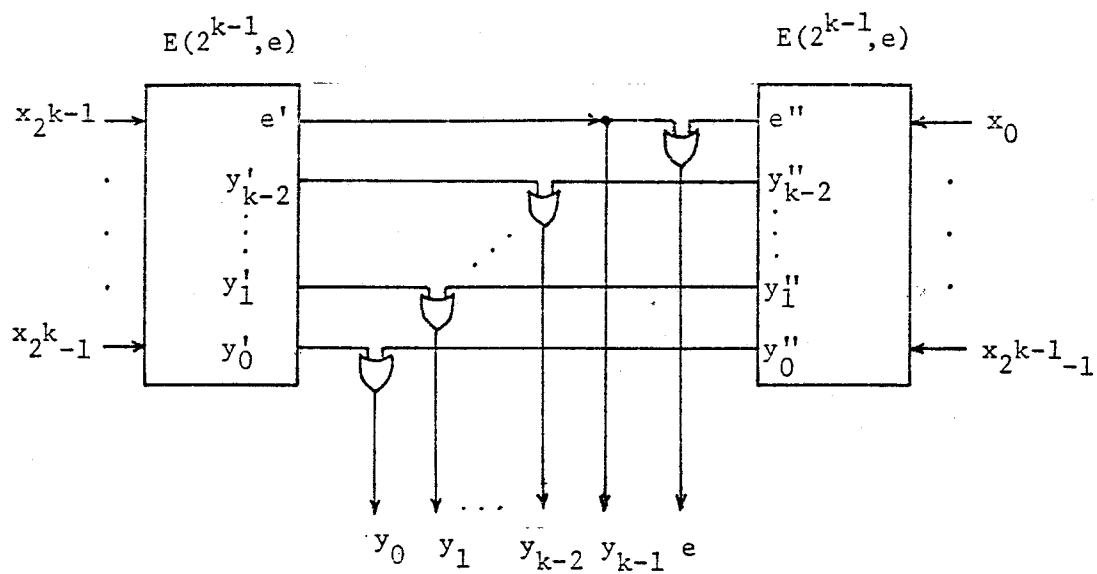


図3.8 $E(2^k, e)$ の帰納的な構成.

Fig.3.8 Recursive construction of $E(2^k, e)$.

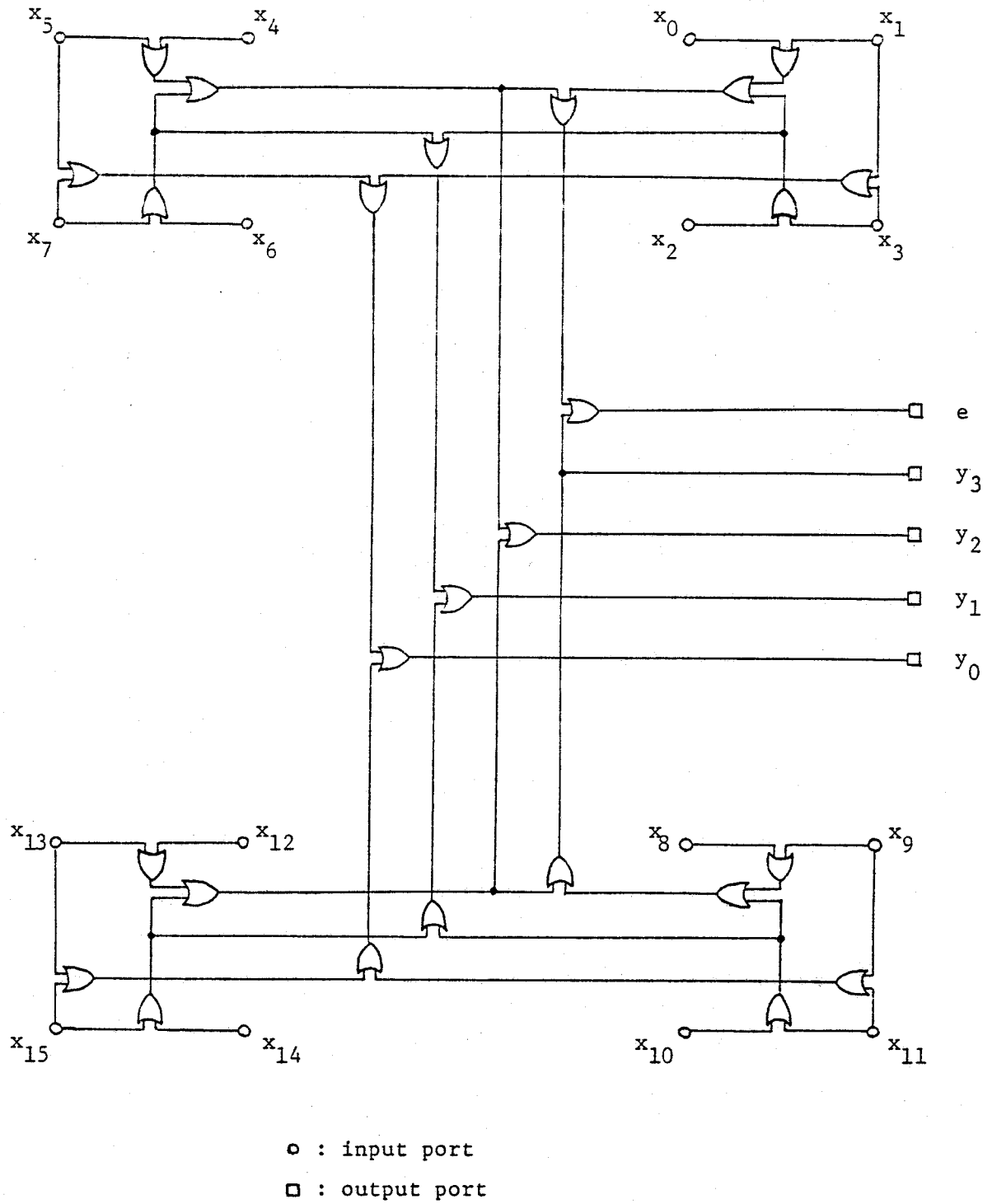


図3.9 $E(2^k, e)$ の配置 ($k=4$).

Fig.3.9 Layout of $E(2^k, e)$ ($k=4$).

が成立つ！従って、 $(n, \lceil \log n \rceil)$ -符号器は $A = O(n)$,
 $T = O(\log n)$ で実現できる。 ■

〔定理 3.4〕 $(n, \lceil \log n \rceil)$ -符号化関数に対して,
 $AT^\alpha = \theta(n \log^{\alpha-1} n) \quad (\alpha \neq 1)$
 が成立つ。

(証明) $(n, \lceil \log n \rceil)$ -符号化関数は $F(n-1, \lfloor (n-1)/2 \rfloor, \lceil \log n \rceil, \lceil \log n \rceil)$ に属する関数ゆえ、定理 3.1 によって下界が成立つ。

回路構成は次の考えに基づいている。 $(n, \lceil \log n \rceil)$ -符号器の入力, 出力をそれぞれ, $x_0, \dots, x_{n-1}, y_0, \dots, y_{\lceil \log n \rceil - 1}$ とする。 n 入力を次のように n/w 個の組に分割する^{††}。

x_0, x_1, \dots, x_{w-1}	第 1 組
$x_w, x_{w+1}, \dots, x_{2w-1}$	第 2 組
⋮	
$x_{n-w}, x_{n-w+1}, \dots, x_{n-1}$	第 n/w 組

符号化関数の定義によって, ちょうど 1 つの入力変数が 1 になるので, ちょうど 1 つの組が 1 になる入力変数を含み, 他の $(n/w - 1)$ の組の変数はすべて 0 である。

今, $x_{jw+k} = 1$ とする。すなわち, 第 $j+1$ 組の k 番目の入力変数が 1 になるとする。

j, k の 2 進数表現をそれぞれ $u_{\lceil \log n/w \rceil - 1} \dots u_0,$

† 付録 A-3 参照。

†† 簡単のため, n/w を 2 のべきとする。

$v_{\lceil \log w \rceil - 1} \cdots v_0$ とすると、符号化関数の性質から、

$$(y_{\lceil \log n \rceil - 1}, \dots, y_0) = (u_{\lceil \log n \rceil - 1}, \dots, u_0, v_{\lceil \log w \rceil - 1}, \dots, v_0)$$

が成立つ。従って、 j と長から $u_{\lceil \log n \rceil - 1} \cdots u_0$ と $v_{\lceil \log w \rceil - 1} \cdots v_0$ を計算できれば符号化関数を実現できる。

このことから、 $(n, \lceil \log n \rceil)$ -符号器 $E(n)$ は、補題 3.3 で示した $E(w, e)$ と $\lceil \log n \rceil$ ビット計数回路とから構成される。 $E(n)$ の構成を図 3.10 に示す。

図 3.10 において、 $\lceil \log n \rceil$ ビット計数回路 CTR は以下のように動作する。CTR に 1 が入力されているあいだ CTR は 1 ずつ増加し、 n とたゞ 0 が入力されるとそれ以後状態は変わらないものとする。また、CTR の初期値は 0 とする。

図 3.10 の回路に各組の入力変数を 1 単位時間おくれで入力する。すべての入力変数が 0 の組が入力されると、 $\bar{e} = 1$ となるので、CTR の値が 1 増加する。また、 $E(w, e)$ の出力は 0 になる。

$x_{jw+k} = 1$ を含む第 $j+1$ 組が入力されるとき、CTR の値は j であり、 $\bar{e} = 1$ が CTR に入力されるので以後 CTR の状態は変化せず j の値を保持している。このとき、 $E(w, e)$ の出力は長である。また、残りの組が入力されても、 $E(w, e)$ の出力値は変化しない。結局、 $E(n)$ の出力は

$$(u_{\lceil \log n \rceil - 1}, \dots, u_0, v_{\lceil \log w \rceil - 1}, \dots, v_0)$$

となるので、正しく符号化関数を実現できる。

$E(n)$ の面積 A 、計算時間 T は次のようになる。補題 3.3 より、 $E(w, e)$ は面積 $O(w)$ で実現でき、CTR は面積 $O(\log n/w)$ で実現できるので、⁽⁷⁾

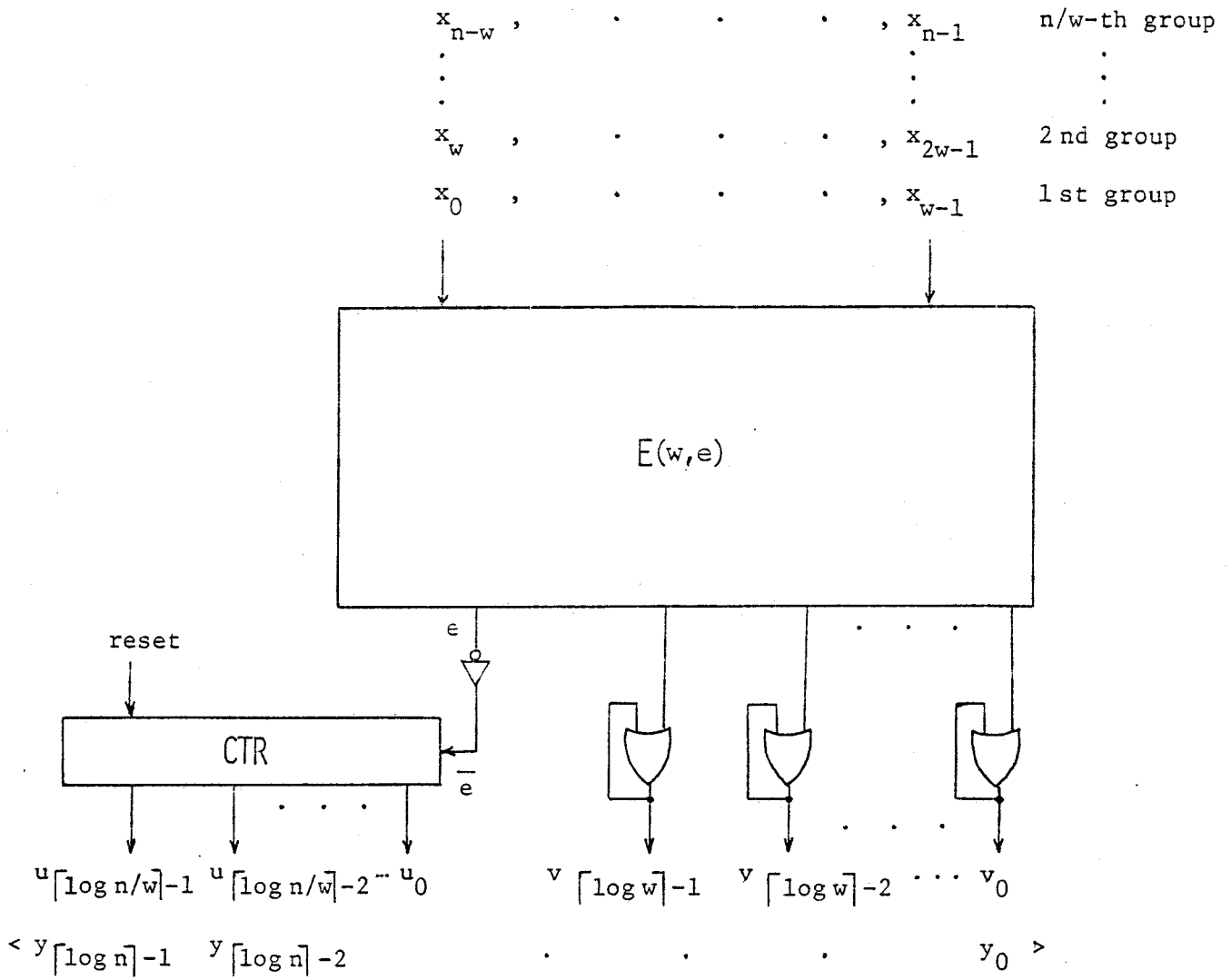


図3.10 下界を達成する $E(n)$ の構成.

Fig.3.10 Construction of $E(n)$ to attain the lower bound.

$$A = O(w + \log n/w) \quad (3.7)$$

が成立つ。また、計算時間は各組が1単位時間おくれで
入力されるので

$$T = O(n/w + \log w) \quad (3.8)$$

となる。

$w = \lceil n/\log n \rceil$ に設定すると、(3.7), (3.8)より

$$\begin{aligned} A &= O(n/\log n), \\ T &= O(\log n) \end{aligned}$$

となる。従って、 $(n, \lceil \log n \rceil)$ -符号器は

$$AT^\alpha = O(n \log^{\alpha-1} n) \quad (\alpha \geq 0)$$

で実現できる。下界の結果と合わせて

$$AT^\alpha = \Theta(n \log^{\alpha-1} n) \quad (\alpha \geq 1)$$

が成立つ。 ■

3.2.2. 一方向一次元繰り返し回路 (CUODIC)

n 入力論理積関数, n ビット比較, m ビット加算などは図3.11のような繰り返し回路によって実現される. このような回路は一方向一次元繰り返し回路 (Combinational Unilateral One-Dimensional Iterative Circuit : CUODIC と略す) と呼ばれる. CUODIC は, 同一基本回路 (セルという) の繰り返しにより実現でき, 配線も規則的なので特に VLSI に適した回路であり, 実用上重要な回路である. 本節では, CUODIC により実現される関数に対する面積時間複雑度を議論する.

Unger は順序回路を一次元に展開して得られる CUODIC を木構造に再構成することにより, 回路の段数を入力数の対数に比例するようにできることを示した⁽²⁾. Unger の手法の基本的な考えは次のとおりである.

CUODIC の各セルの機能を表した関数表の各欄は入力によって決まる変換と見なすことができる. I_a を入力 x_a によって決まる変換とする. このとき, I_a を要素とし, I_a の合成演算を積とする半群が得られる. 半群の積は結合則を満たすので, 半群の積を計算するセルを木状に構成することができ, 段数が入力数の対数に比例するようにできる.

Unger の木状回路の面積を削減するために, 2分木の H型配置を用いる⁽³⁾. H型配置を用いると葉数に比例した面積で埋め込める. ここで問題となるのは各セルからの出力 Z_{i_1}, \dots, Z_{i_r} をどのように取り扱うかであるが図3.12にその配置を示す. 状態の変換 I_{1-8} は根のセルから得られる. 各セルの状態を決定するために, 初期状態 S_0 が根のセルから入力され, IS セルは状態の変換と現在の状態から得られる情報を伝達してゆく. 葉のセル IZ

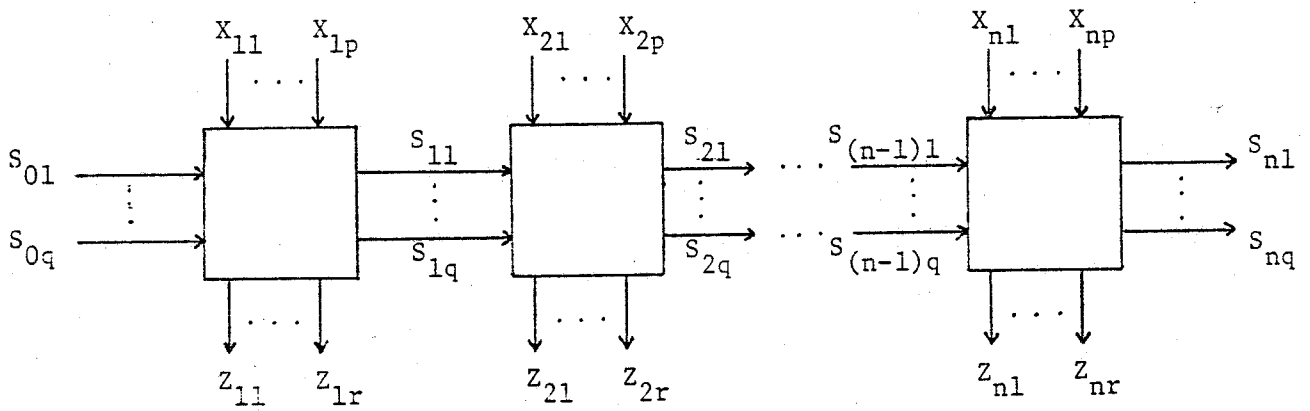


圖 3.11 一次元一方向繰り返し回路 (CUODIC).

Fig. 3.11 CUODIC.

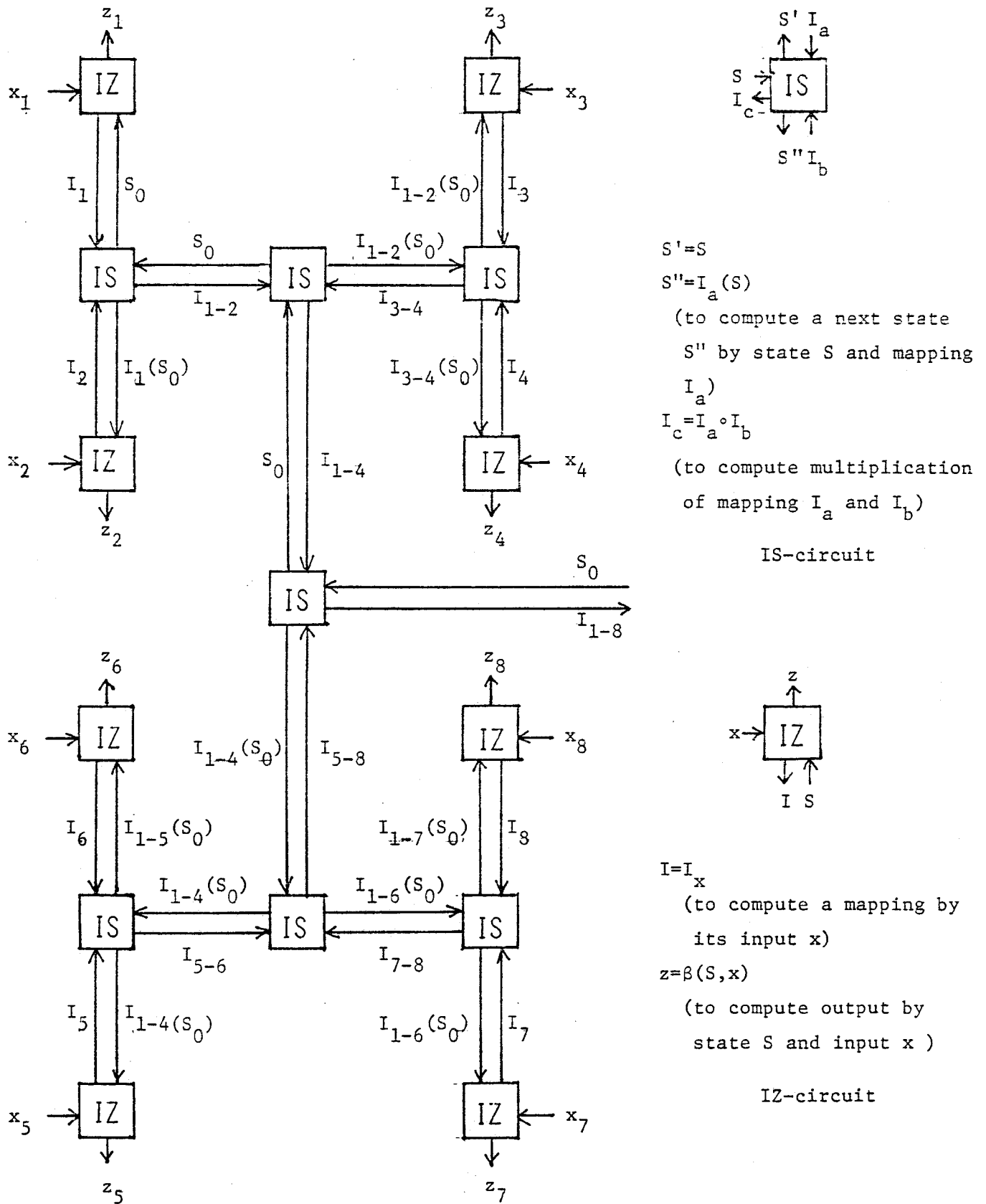


図 3.12 CUODIC に対する H 型配置

($n=8, p=q=r=1$)

Fig.3.12 An H-shape layout for a CUODIC ($n=8$).

(For simplicity, the case that $p=q=r=1$ is depicted)

は入力とその状態から出力を決定する。

このように構成することにより、

$$A = O(n c(n)),$$
$$T = O(\tau(n) \log n)$$

で実現できる。ここで $c(n)$, $\tau(n)$ はそれぞれ IS セル (または IZ セル) の面積および計算時間であり、一般の場合、いずれも n に依存する。しかしながら、2つの n ビット 2 進数の加算、2つの n ビット 2 進数の比較などは、 $c(n)$, $\tau(n)$ は n に依存せず定数になる。従って、2つの n ビット加算については

$$AT^\alpha = O(n \log^\alpha n) \quad (\alpha \geq 0)$$

で実現できる。一方、加算の面積時間積の下界は、

$$AT^\alpha = \Omega(n \log^\alpha n) \quad (\alpha > 1)$$

となることが示されている⁽⁸⁾。Brent-Kung はこの下界を達成する回路構成を示しているが⁽²⁾、図 3.12 と同様の回路によって下界を達成する構成が得られる。

2つの n ビット加算のように、各入力に対する出力が必要なとき、葉のセル IZ は状態に対する情報が必要になるので、図 3.12 の回路は情報が 2 方向に伝達する。従って、パイプライン入出力を用いることができない。ところが、 n 入力論理積関数、2つの n ビット 2 進数の比較のような関数は、状態変換と初期状態だけで結果が決定できる。このとき、情報の伝達は 1 方向になるのでパイ

イアライン入出力を用いることができ、復号器、符号器
のように入出力を分割することにより面積を削減できる
ので、定理3.1の下界を達成できる。

〔定理3.5〕 n 入力論理積（論理和、排他的論理和）
関数、2つの n ビット2進数の比較に対して、

$$AT^\alpha = \Theta(n \log^{\alpha-1} n) \quad (\alpha \geq 1)$$

が成立つ。

（証明） n 入力論理積（論理和、排他的論理和）関数
は $F(n, n, 1, 1)$ に属する関数であり、2つの n ビット
2進数の比較は $F(n, n, 2, 2)$ に属する関数ゆえ、定理3.1
により、いずれも

$$AT^\alpha = \Omega(n \log^{\alpha-1} n) \quad (\alpha \geq 1)$$

が成立つ。

n 入力論理積関数を計算するために図3.13のような回
路を構成する。 n 個の入力を w 個ずつの組に分け、各入
力を1単位時間ずつずらして、図3.13の回路に入力して
ゆき、最後のところで直列に論理積をとってゆく。

このようにすると、最初の入力を入れはじめて結果が
出るまでの時間 T は

$$T = O(n/w + \log w)$$

となる。また、回路の面積 A は、葉数 w の完全2分木を
H型に配置してあるので

$$A = O(w)$$

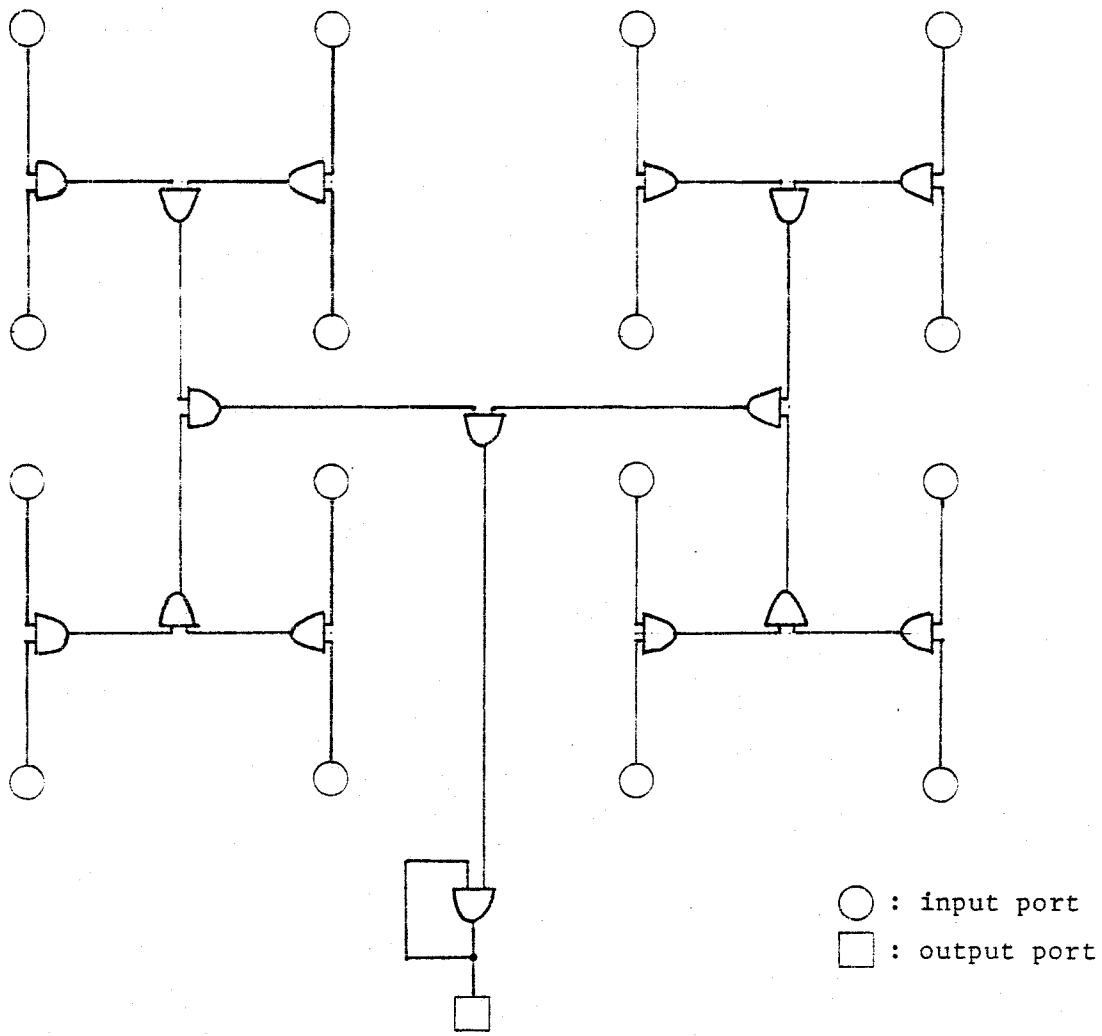


図3.13 論理積関数を実現する回路 ($w=16$).

Fig.3.13 Circuit to compute AND function ($w=16$).

である。 $w = \lceil n / \log n \rceil$ に設定すれば

$$AT^d = O(n \log^{d-1} n) \quad (d \geq 0)$$

が成立つ。ゆえに、 n 入力論理積関数に対して、

$$AT^d = \theta(n \log^{d-1} n) \quad (d \geq 1)$$

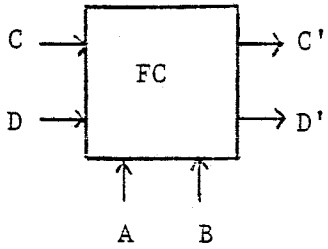
が成立つ。

論理和、排他的論理和に対しては、図3.13のANDゲートをそれぞれ、ORゲート、XORゲートにおきかえればよい。

2つの n ビット 2 進数の比較に対しては、ANDゲートのかわりに、図3.14(a)の基本回路(全比較器と呼ぶ)を用いる。全比較器は図3.14(b)の関数値表で示される4入力2出力回路である。ここで、00, 01, 10, 11をそれぞれ、0, 1, 2, 3と読み換え、2本の導線を4値の1本の導線とみなす。このとき図3.14(c)の真理値表を得る。こうして4入力2出力回路を新しく4値関数

$$f_{\text{ass}} : \{0, 1, 2, 3\}^2 \rightarrow \{0, 1, 2, 3\}$$

として定義する。このように定義すると f_{ass} は結合則を満たすことが容易に確かめられるので、図3.13の回路のANDゲートを f_{ass} を実現する回路(すなわち、全比較器でおきかえることにより2つの n ビット 2 進数の比較を実現できる。 ■

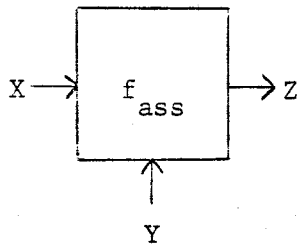


(a) full comparator

C	D	A	B	C'	D'	meaning
0	0	0	0	1	1	=
0	0	0	1	0	1	<
0	0	1	0	1	0	>
0	0	1	1	1	1	=
0	1	-	-	0	1	<
1	0	-	-	1	0	>
1	1	0	0	1	1	=
1	1	0	1	0	1	<
1	1	1	0	1	0	>
1	1	1	1	1	1	=

- : don't care

(b) function value table for full comparator



(c) truth value table for f_{ass}

X(CD)	Y(AB)	Z(C'D')
0	0	3
0	1	1
0	2	2
0	3	3
1	-	1
2	-	2
3	0	3
3	1	1
3	2	2
3	3	3

- : don't care

图3.14 全比较器与4值函数 f_{ass} .

Fig.3.14 Full comparator and the derived function f_{ass} .

3.2.3. 優先順位付符号化関数

優先順位付符号化関数を実現する回路を優先順位付符号器 (priority encoder) と呼ぶ。この関数は入力変数の添字が優先順位に対応し、最大の優先順位をもつ入力変数を選択し、その添字の2進数表現を出力する。 $(n, \lceil \log n \rceil)$ -優先順位付符号器は、 n 個の要求が同時に与えられ、そのうちで最大の優先順位をもつ要求を選択するという性質をもつ。従って、この回路の面積は仮定(A-3)より、 $A = \Omega(n)$ を満し、面積時間積については、定理3.2より

$$AT^\alpha = \Omega(n \log^\alpha n) \quad (\alpha \geq 0)$$

が成立つ。次の定理ではこの下界を達成する回路構成が存在することを示す。

[定理3.6] $(n, \lceil \log n \rceil)$ -優先順位付符号化関数に対して、

$$AT^\alpha = \Theta(n \log^\alpha n) \quad (\alpha \geq 0)$$

が成立つ。

(証明) $(n, \lceil \log n \rceil)$ -優先順位付符号化関数は、 $F(n, n, \lceil \log n \rceil, \lceil \log n \rceil)$ に属し、問題の性質上、 $A = \Omega(n)$ となるので、定理3.2より下界が成立つ。

回路構成は、Ungerの手法を利用する。 $n=8$ の場合について考える。各出力は次のように表される。

$$y_0 = x_7 + \bar{x}_6 x_5 + \bar{x}_6 \bar{x}_4 x_3 + \bar{x}_6 \bar{x}_4 \bar{x}_2 x_1 \quad (3.9)$$

$$y_1 = x_7 + x_6 + \bar{x}_5 \bar{x}_4 x_3 + \bar{x}_5 \bar{x}_4 x_2 \quad (3.10)$$

$$y_2 = x_7 + x_6 + x_5 + x_4 \quad (3.11)$$

最下位桁出力 y_0 は図 3.15 の回路により実現される。この回路は 2 つの通信シグナル a , b をもつ CUODIC である。 b は (3.9) 中のすべての積項を生成し、 a は偶数の添字をもつ入力変数の和項を生成する。図 3.15 のセルの関数値表を図 3.16 (a) に示す。この関数値表の各欄に対応する変換を α , β , γ , δ とし、その変換の積を図 3.16 (b) に示す。 α , β , γ , δ をそれぞれ 00, 01, 11, 10 と符号化すると、この変換の積を計算する回路 (I-回路と呼ぶ) が得られる。I-回路およびその関数値表を図 3.17 に示す。

この構成において、セルへの入力 $x_0 x_1$ は直接入力によって決まる変換を符号化する。入力 $x_0 x_1$ によって決まる変換を $t(x_0 x_1)$ と表すと、最下位桁出力 y_0 は

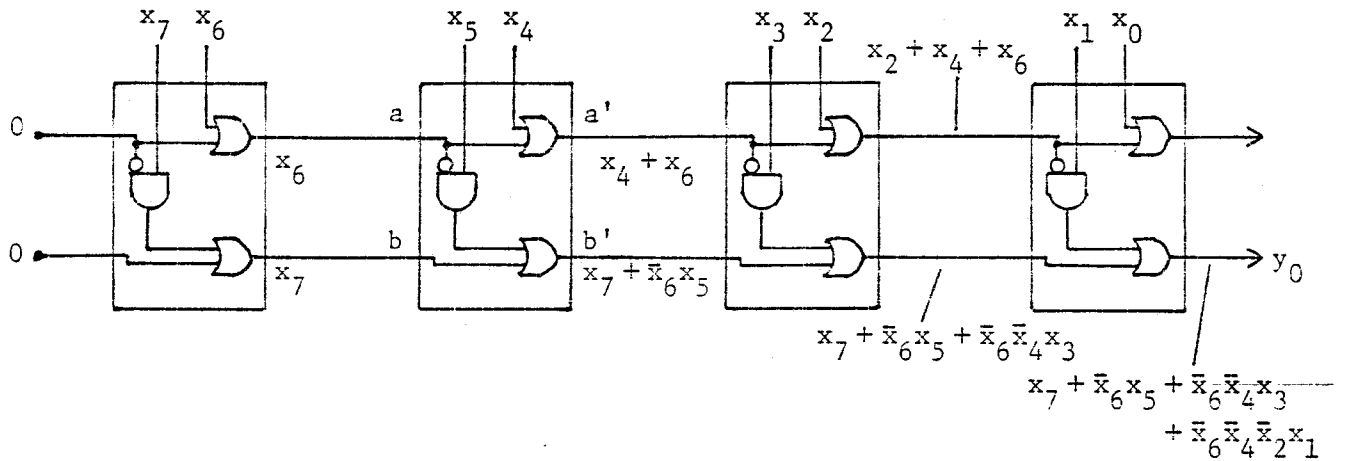
$$y_0 = [t(x_0 x_1) \cdot (t(x_2 x_3) \cdot (t(x_4 x_5) \cdot (t(x_6 x_7)) \cdot \Delta_0)))]_e$$

と表される。ここで Δ_0 は境界入力であり、 $[]_e$ は I 回路の e 出力を意味する。変換の積は結合則を満たすので

$$y_0 = [[t(x_0 x_1) \cdot t(x_2 x_3)] \cdot [t(x_4 x_5) \cdot t(x_6 x_7)] \cdot \Delta_0]_e$$

と表すことができる。さらに、 Δ_0 を 00 と符号化すると、

$$y_0 = [[t(x_0 x_1) \cdot t(x_2 x_3)] \cdot [t(x_4 x_5) \cdot t(x_6 x_7)]]_e$$



$$a' = x_e + a \quad x_e : \text{input with even index}$$

$$b' = x_o \bar{a} + b \quad x_o : \text{input with odd index}$$

図3.15 y_0 を実現する CUODIC ($n=8$).

Fig.3.15 CUODIC computing y_0 .

$\begin{matrix} x_o x_e \\ ab \end{matrix}$	00	01	11	10
00	00	10	11	01
01	01	11	11	01
11	11	11	11	11
10	10	10	10	10

	α	β	γ	δ
α	α	β	γ	δ
β	β	β	β	β
γ	γ	γ	γ	γ
δ	δ	γ	γ	δ

$$a' = x_e + a$$

$$b' = x_o \bar{a} + b$$

$$\alpha = 00, \beta = 01, \gamma = 11, \delta = 10$$

(a)

(b)

図3.16 図3.15の基本セルの関数値表と(a)の各欄の変換積表.

Fig.3.16 Function value table for the basic cell in Fig.3.15 and the product of the transformation of each column.

と表される。

このことから、 y_0 は I-回路を用いて、図 3.18 の木状回路により実現できる。

出力 y_1 の場合は、(3.10) を次のように変形する。

$$y_1 = (x_7 + x_6) + \overline{(x_5 + x_4)} \cdot (x_3 + x_2)$$

このように変形すると、図 3.17 の I-回路を利用できる。

すなわち、 $x_7 + x_6$ 、 $x_5 + x_4$ 、 $x_3 + x_2$ を I-回路への入力と見なすことにより、図 3.19 のように構成できる。

最上位桁出力 y_2 は、 x_4 から x_7 の論理和を計算すればよい。

一般の場合、各出力は図 3.20 のように I-回路を用いて実現される。最下位桁出力 y_0 は I-回路だけで構成される。第 i 桁出力 y_i ($1 \leq i \leq \lceil \log n \rceil - 2$) は

$$x_{n-1} + \dots + x_{n-2^i},$$

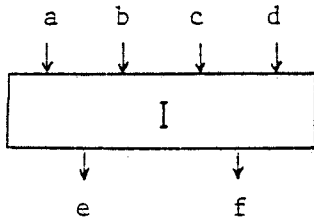
$$x_{n-2^{i-1}} + \dots + x_{n-2^{i+1}},$$

⋮

$$x_{2^{i-1}} + \dots + x_0.$$

を入力と見なし、I-回路の木状回路に入力することにより実現される。最上位桁出力 $y_{\lceil \log n \rceil - 1}$ は $x_{\lfloor n/2 \rfloor + 1}$ から x_{n-1} の論理和を計算すればよい。

図 3.20 の各木状回路を H 型に配置すれば、面積 $O(n)$ 計算時間 $O(\log n)$ で実現できる。しかしながら、回路全体の面積 A は少なくとも $\Omega(n \log n)$ となり、さらに n 入力を各木状回路に伝達するためには $\Omega(n \log n)$ 以



$$e = a + c \cdot \bar{b}$$

$$f = b + d$$

(a)

cd \ ab	00	01	11	10
00	00	01	11	10
01	01	01	01	01
11	11	11	11	11
10	10	11	11	10

(b)

ef

図3.17 I-回路とその関数値表.

Fig.3.17 I-circuit and its function value table.

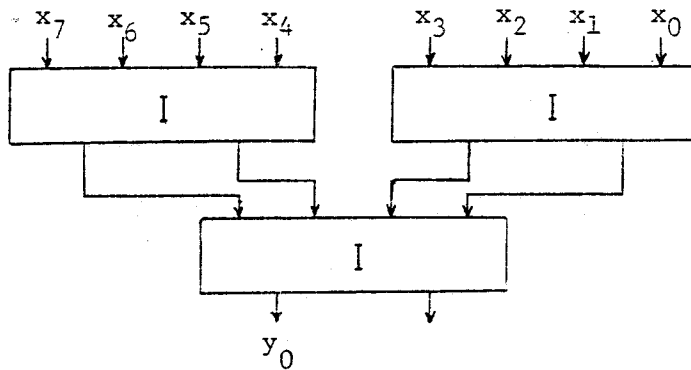


図3.18 I-回路による y_0 の実現.

Fig.3.18 Realization of y_0 by I-circuit.

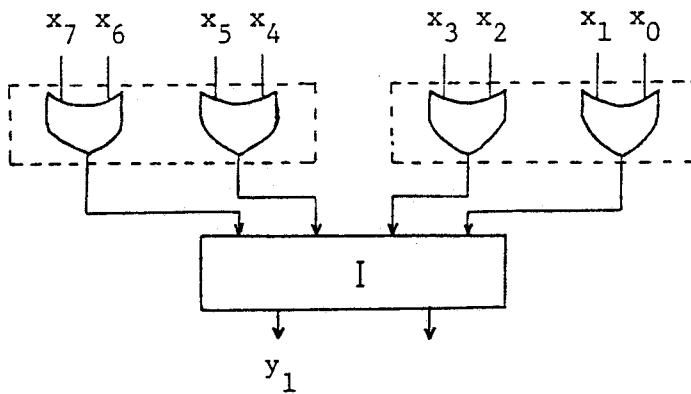


図3.19 I-回路による y_1 の実現.

Fig.3.19 Realization of y_1 by I-circuit.

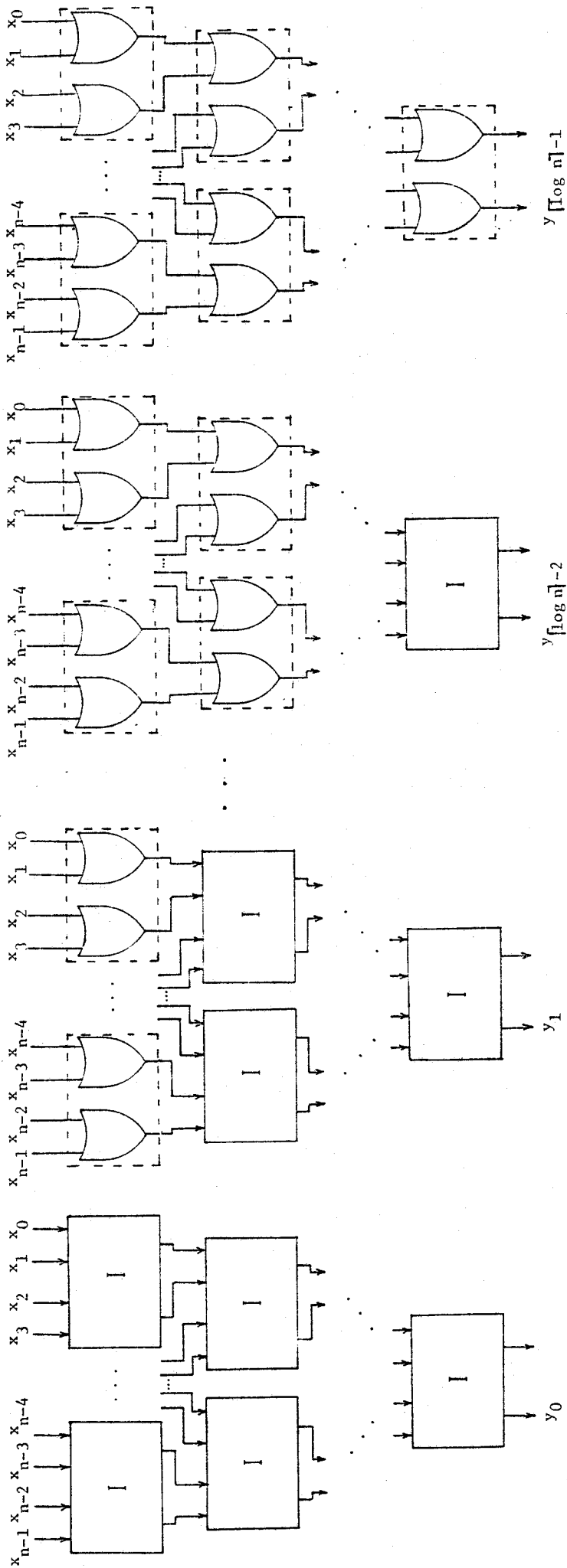


図3.20 $(n, \lfloor \log n \rfloor)$ -優先順位付符号器の構成.

Fig.3.20 Construction of $(n, \lfloor \log n \rfloor)$ -priority encoder.

上の面積を必要とする。従って、このままでは下界 $AT^\alpha = \Omega(n \log^\alpha n)$ ($\alpha \geq 0$) を達成できない。このギャップを埋めるために、図3.21のような回路を利用する。この回路はI-回路と2つのORゲート（O-回路と呼ぶ）を選択入力 λ によって、いずれの回路の出力をずるかを次のように切り換えることができる。

$\lambda = 0$ のとき、I-回路の出力、
 $\lambda = 1$ のとき、O-回路の出力。

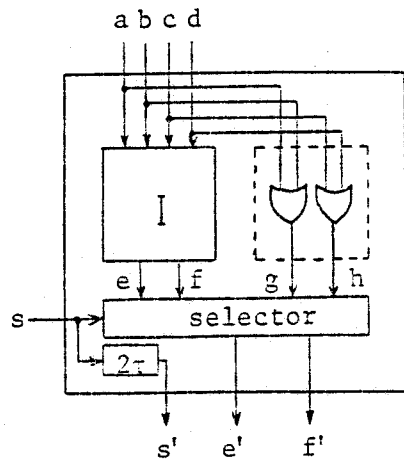
また、この回路をIO-回路と呼ぶ。選択入力は、次の段のIO-回路に2 τ 単位時間おくれで伝達される。ここではIO-回路の遅延時間とする。IO-回路を用いると、 $(n, \lceil \log n \rceil)$ -優先順位付符号器は図3.22のように構成される。ここで、出力 $y_0, \dots, y_{\lceil \log n \rceil - 1}$ は図3.22の回路の根のIO-回路の左側の出力から得られる。

図3.22の回路は、次のように動作する。

入力 x_0, \dots, x_{n-1} はレジスタに貯えられ、時刻0に与えられる。入力は τ 時間おきにレジスタから読み出され、選択入力は時刻 τ に $\lambda = 1$ として与えられる。このようにして y_0 は時刻 2τ に得られる[†]。時刻 τ には、図3.22の回路の葉節点に対応するIO-回路はO-回路に切り換えられ、 y_1 は時刻 3τ に得られる。時刻 3τ には、根のIO-回路がO-回路に切り換えられ、時刻 4τ に y_2 が得られる。

一般の場合、 y_0 が時刻 $\tau(\lceil \log n \rceil - 1)$ に得られ、以下で単位時間おくれで、 $y_1, \dots, y_{\lceil \log n \rceil - 1}$ が得られる。計算は $2\tau(\lceil \log n \rceil - 1)$ に終了するので、

[†] τ は導線の遅延を含む。



$$e'(t+\tau) = \bar{s}(t) \cdot e(t) + s(t) \cdot g(t)$$

$$f'(t+\tau) = \bar{s}(t) \cdot f(t) + s(t) \cdot h(t)$$

$$s'(t+2\tau) = s(t)$$

图 3.21 IO-回路.

Fig. 3.21 IO-circuit.

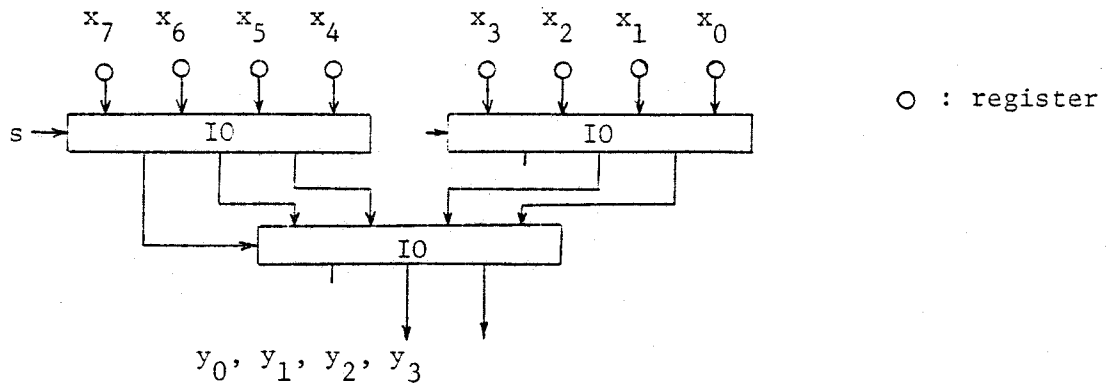


图 3.22 (8,3)-优先顺序付符号器.

Fig. 3.22 (8,3)-priority encoder.

$$T = O(\log n)$$

が成立つ。回路の面積は、H型配置により

$$A = O(n)$$

である。従って、 $(n, \lceil \log n \rceil)$ -優先順位付符号器は

$$AT^\alpha = O(n \log^\alpha n) \quad (\alpha \geq 0)$$

で実現できる。 ■

3.2.4 対称関数族

ここでは、冗長でない n 入力対称関数族を実現する回路の面積時間複雑度を議論する。 n 入力対称関数は計算時間 $O(\log n)$ で実現できることが示されている⁽⁶³⁾。計算時間 $O(\log n)$ で実現するために、 $(n, \lceil \log(n+1) \rceil)$ -重み関数を用いる。

Muller-Preparata は、 $(n, \lceil \log(n+1) \rceil)$ -重み関数の帰納的構造を利用して、 $T = O(\log n)$ での実現を示している⁽⁶⁴⁾。この回路を、図 3.23 のように配置することによって $A = O(n)$ で埋め込める。† また、符号器と同様に入力を $\lceil \log n \rceil$ 組に分割することにより、 $A = O(n/\log n)$ に削減できる。この場合は、 $\lceil \log n \rceil$ 組の結果を $\lceil \log(n+1) \rceil$ ビット加算器を用いて累算してゆけばよい (図 3.24)。

$(n, \lceil \log(n+1) \rceil)$ -重み関数は、 $F(n, n, \lceil \log(n+1) \rceil, \lceil \log(n+1) \rceil)$ に属するので、定理 3.1 より、面積時間積の下界は、

$$AT^\alpha = \Omega(n \log^{\alpha-1} n) \quad (\alpha \geq 1)$$

となる。これらのことから次の補題が成立つ。

[補題 3.4] $(n, \lceil \log(n+1) \rceil)$ -重み関数に対して、
 $AT^\alpha = \Theta(n \log^{\alpha-1} n) \quad (\alpha \geq 1)$
が成立つ。 ■

† 付録 A-4 参照。

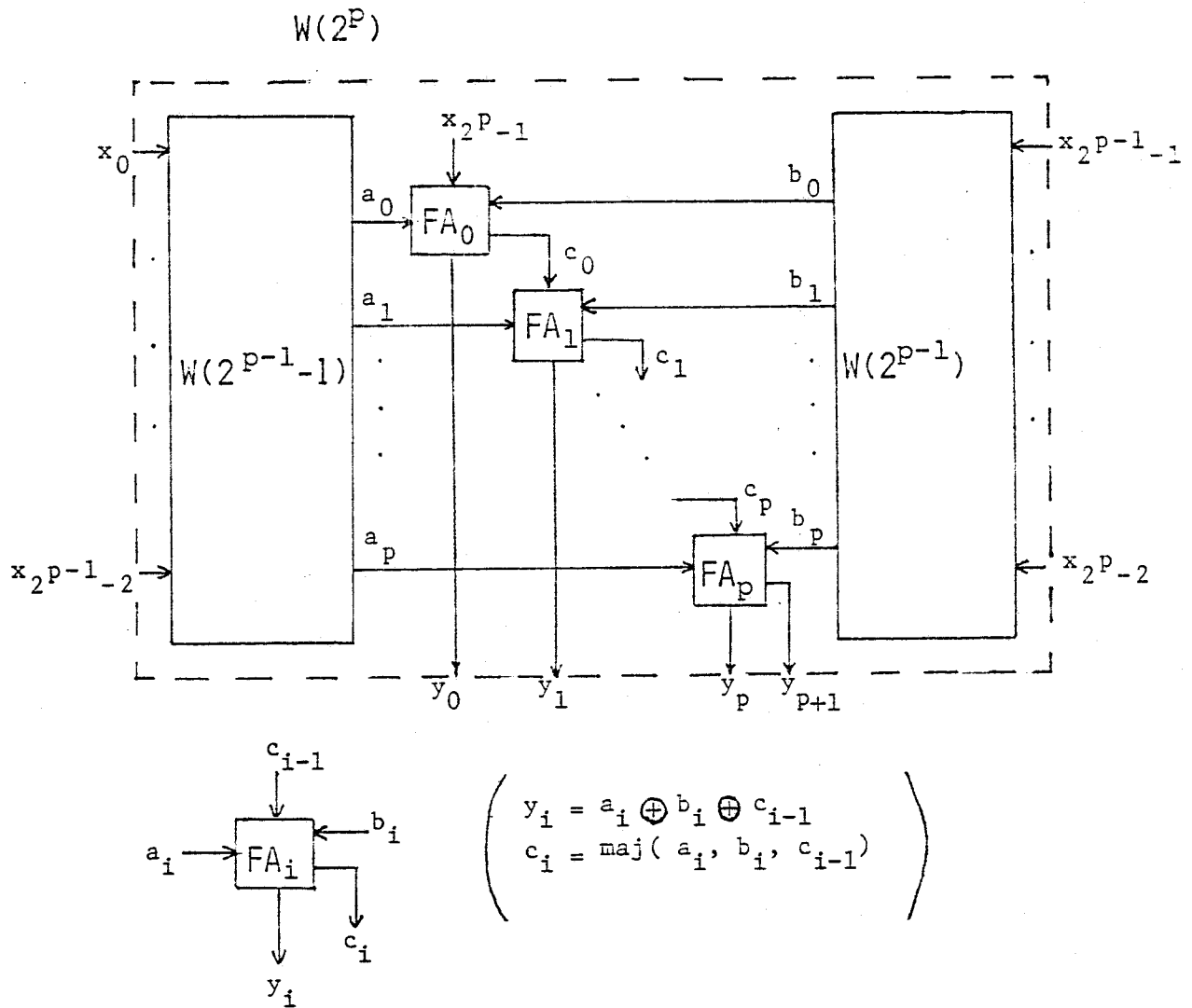


図3.23 $(n, \lceil \log(n+1) \rceil)$ -重み関数を実現する

Muller-Preparataの回路の配置 ($n=2^P$).

Fig.3.23 Layout of Muller-Preparata's circuit to compute $(n, \lceil \log(n+1) \rceil)$ -weight function $(W(2^P))$.

(For simplicity, $n = 2^P$)

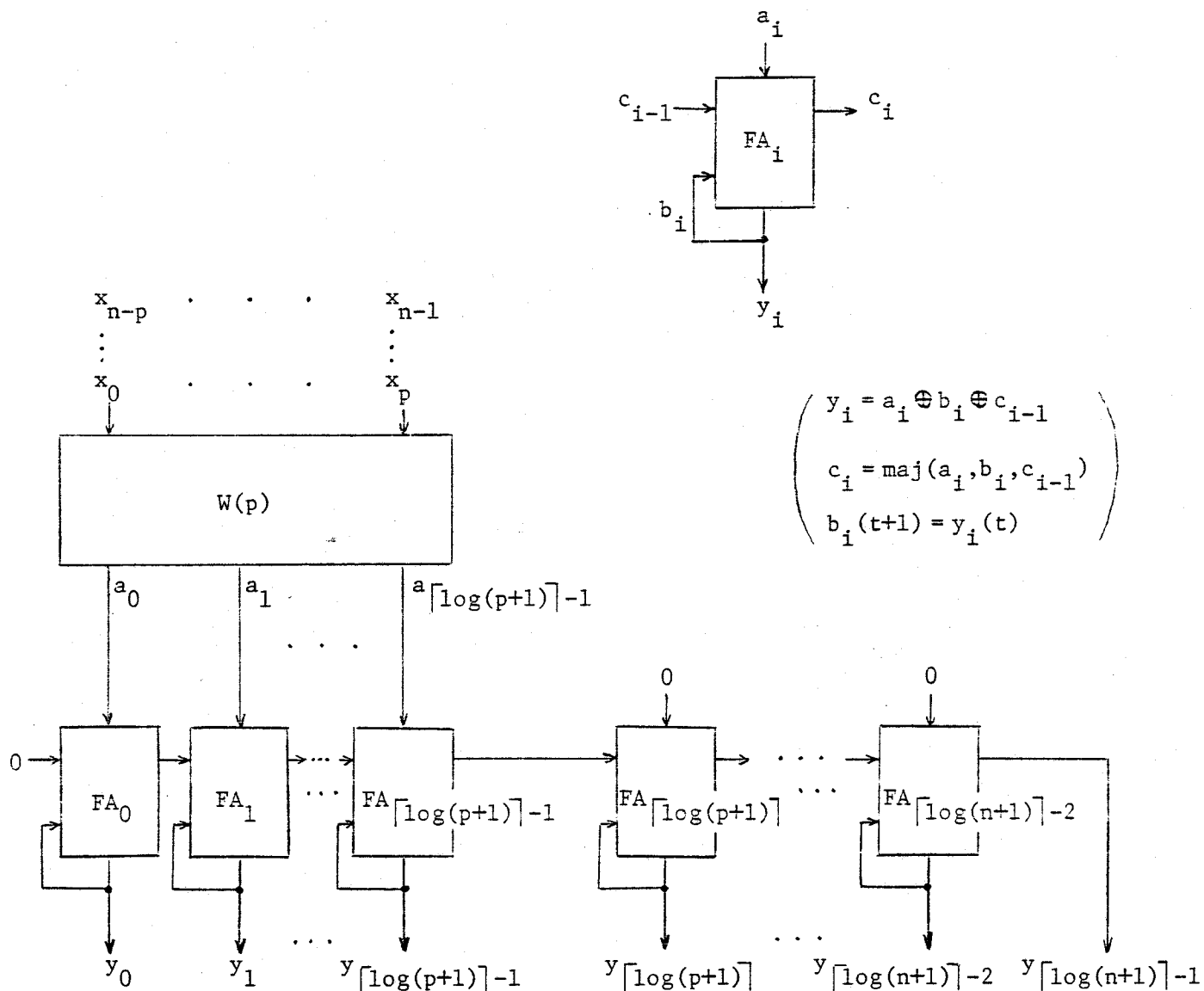


図3.24 下界を達成する $(n, \lceil \log(n+1) \rceil)$ -重み関数の構成
 $(p = \lceil n / \log n \rceil)$.

Fig.3.24 Construction of $(n, \lceil \log(n+1) \rceil)$ -weight function to attain the lower bound ($p = \lceil n / \log n \rceil$).

$(n, 1)$ -対称関数は、ある非負整数の組 (a_1, \dots, a_k) ($0 \leq a_1 < \dots < a_k \leq n$, $0 \leq k \leq n$) があって、入力変数のうちその値が 1 となるものの個数がある a_i に等しいとき 1 を出力し、そうでないとき 0 を出力する関数と考えられる。以降、 $(n, 1)$ -対称関数を $S_{(a_1, \dots, a_n)}^n$ と表す。また、 S_n に属する関数を a_i の個数で分類し、非負整数の組の個数が k である冗長でない $(n, 1)$ -対称関数の集合を $S_n(k)$ で表す。すなわち、

$$S_n = \bigcup_{0 \leq k \leq n+1} S_n(k).$$

[定理 3.7] $k = O(n/\log^3 n)$ とする。 $S_n(k)$ に属する関数に対して、

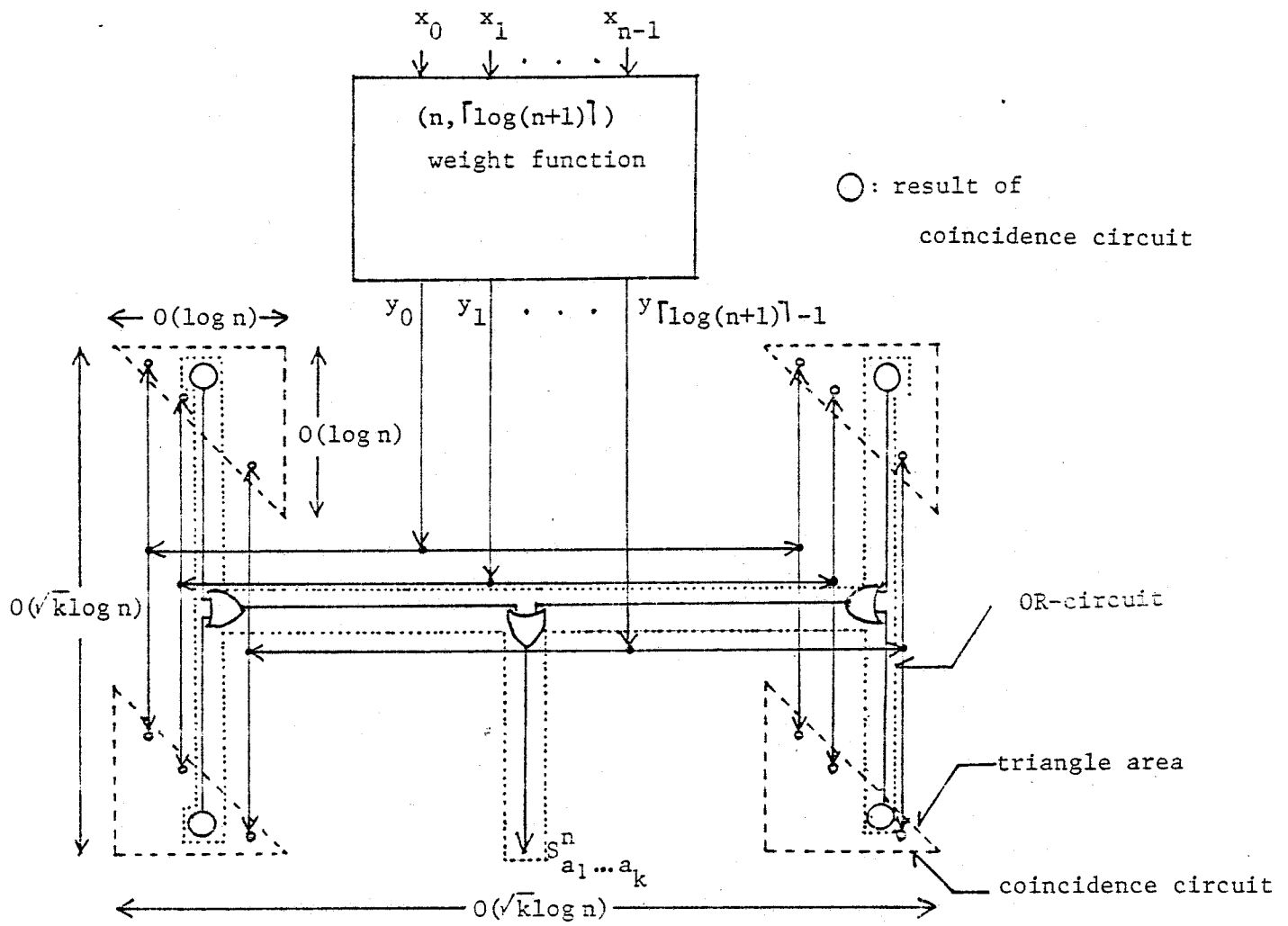
$$AT^\alpha = \Theta(n \log^{\alpha-1} n) \quad (\alpha \neq 1)$$

が成立つ。

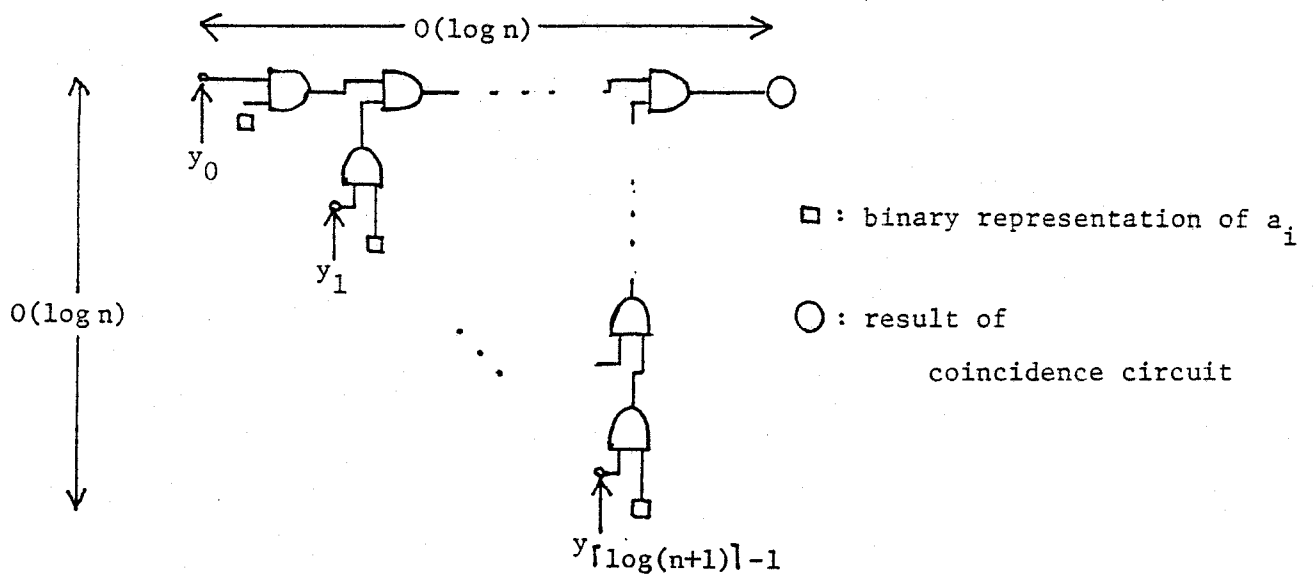
(証明) $S_n(k) \subseteq F(n, n, 1, 1)$ であるから下界が成立つ。

対称関数 $S_{(a_1, \dots, a_k)}^n$ ($k = O(n/\log^3 n)$) は、入力中の 1 の個数の 2 進表現と、 a_1, \dots, a_k の 2 進表現との一致をとり、これら k 個の一致結果の論理和をとることにより実現できる。図 3.25(a) のような回路を構成する。 n 入力中の 1 の個数の 2 進表現は $(n, \lceil \log(n+1) \rceil)$ -重み関数で得る。 $\lceil \log(n+1) \rceil$ 個の重み関数の出力は、図 3.25(a) のような配置で、それぞれ k 個のコピーを作る。そして、 a_1, \dots, a_k の 2 進表現それぞれとの一致をとり、一致結果の論理和をとる回路を重ねて求めればよい。

$(n, \lceil \log(n+1) \rceil)$ -重み関数は、補題 3.4 により、面積 $O(n/\log n)$ 、計算時間 $O(\log n)$ である。重み関数の出力の k 個のコピーを作る回路は、図 3.25(a) のように配置



(a)



(b)

图3.25 对称関数を實現する回路 ($k=4$)
Fig.3.25 Circuit to compute symmetric function ($k=4$).

すると面積 $O(k \log^2 n)$ となり[†]、計算時間 $O(\log k)$ である。 a_i の 2 進表現と一致をとる回路は $2 \lceil \log(m+1) \rceil$ 入力論理関数を計算すればよく、図 3.25 (a) の三角領域に配置でき (図 3.25 (b))、計算時間は $O(\log m)$ である。一致結果の論理和をとる回路の計算時間は $O(\log \log n)$ となる。以上のことから、 $S^m(a_1, \dots, a_k)$ を実現する回路の面積 A 、計算時間 T について、それぞれ

$$\begin{aligned} A &= O(m/\log n + k \log^2 n), \\ T &= O(\log n + \log k) \end{aligned}$$

が成立つ。 $k = O(m/\log^2 n)$ であるから

$$AT^\alpha = O(m \log^{\alpha-1} m) \quad (\alpha \geq 1)$$

である。 ■

定理 3.7 の構成は、 a_1, \dots, a_k を 2 進数として回路内に記憶している。この方法を用いる限り、任意の対称関数を $AT^\alpha = O(m \log^{\alpha-1} m)$ ($\alpha \geq 1$) で実現することはできない。実際、次の対称関数は $AT^\alpha = O(m \log^k n)$ でさえ実現できない。

$$S^m(a_1, \dots, a_k) = \begin{cases} 1 & (a_i \text{ は } m \text{ 以下の素数}) \\ 0 & (\text{その他}) \end{cases}$$

† 付録 A-5 参照。

素数定理により, n 以下の素数の数は $\Theta(n/\log n)$ であり, $k = \Theta(n/\log n)$ となるので,

$$AT^\alpha = O(n \log^\alpha n) \quad (\alpha \geq 1)$$

で実現できる。 S_n に属する任意の対称関数を

$$AT^\alpha = O(n \log^{\alpha-1} n) \quad (\alpha \geq 1)$$

で実現できるかどうかは未解決であるが, 次の定理により,

$$AT^\alpha = O(n \log^\alpha n) \quad (\alpha \geq 0)$$

では実現できる。

[定理 3.8] S_n に属する任意の対称関数は,

$$AT^\alpha = O(n \log^\alpha n) \quad (\alpha \geq 0)$$

で実現できる。

(証明) 図 3.26 のように回路を構成する。 $(n, \lceil \log(n+1) \rceil)$ -関数の出力を $(\lceil \log(n+1) \rceil, n+1)$ -復号器により復号する。このとき, 復号器の i 番目 ($0 \leq i \leq n$) に対応する出力は, もとの入力中の 1 の個数が i 個であることを示している。それらの出力の a_1, \dots, a_k に対応するものの論理和をとれば, S_{ca_1, \dots, a_k}^m を実現できる。論理和をとる回路は復号器に重ねるように配置すればよい。

$(\lceil \log(n+1) \rceil, n+1)$ -復号器は, 補題 3.2 により, 面積 $O(n)$, 計算時間 $O(\log n)$ で実現できる。従って, 図 3.26 の回路は

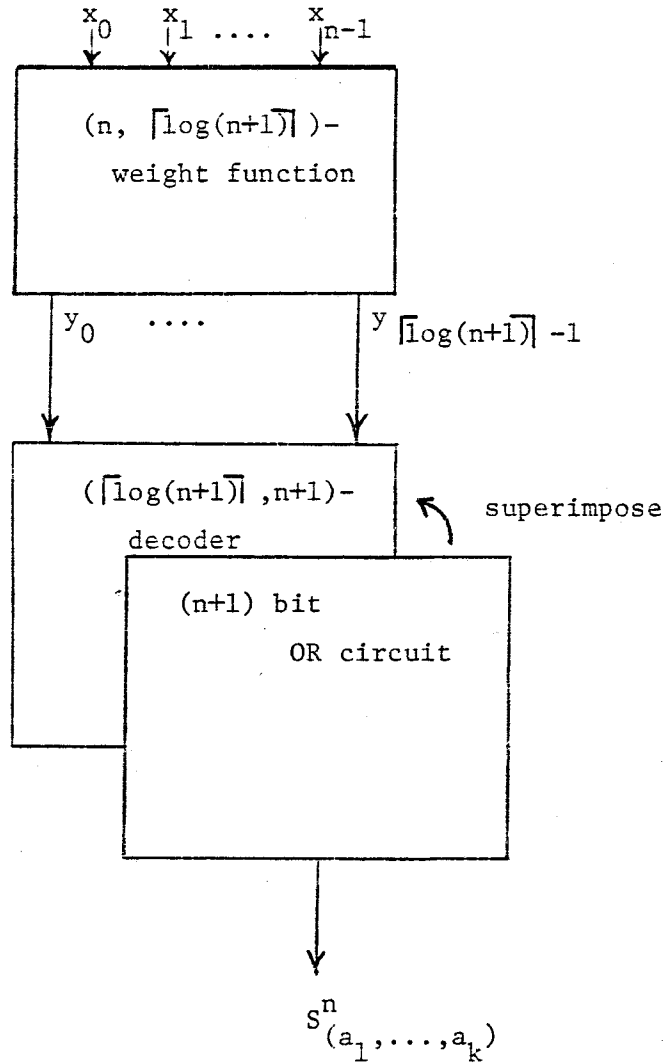


図3.26 $AT^\alpha = O(n \log^\alpha n)$ で対称関数を実現する構成.

Fig.3.26 Construction of symmetric function with $AT^\alpha = O(n \cdot \log^\alpha n)$.

$$A = O(n),$$

$$T = O(\log n)$$

が実現できる。 ■

4. 面積時間複雑度 (2)

——境界配置の仮定を要請した場合——

本章では、境界配置の仮定を付加した VLSI モデルのもとで、第3章で議論した $F(n, N, m, M)$ に属する関数を実現する回路の面積時間複雑度を考察する。

本章で考える回路は、VLSI モデルの仮定 (A-1) ~ (A-9) を満足するとし、回路はある凸平面領域に埋め込まれる。

凸平面領域に関する平面幾何学の結果として次の命題が知られている⁽¹⁾。

平面上の凸領域 R の周上の任意の2点を結ぶ線分を R の弦 (chord) と呼ぶ。 R の弦のうち長さ最大のものを R の直径 (diameter) と呼ぶ。 R の直径は $1 \geq$ とは限らない。

[命題4.1]⁽¹⁾ 面積 A , 周長 P , 直径 D の任意の凸領域において、 D に垂直な弦の長さを L とすると、

$$A \geq LP/2\pi \quad (4.1)$$

$$A \geq L^2/2 \quad (4.2)$$

が成立つ。ここで π は円周率である。 ■

4.1. 回路に対する面積時間積の下界

$F(n, m)$ に属する関数を実現する回路が埋め込まれる凸領域の周長と回路の計算時間の間には次の関係が成立する。

[補題4.1] f を $F(n, m)$ に属する関数とする。 C_f が埋め込まれる凸領域の周長を P とすると

$$P \cdot T(C_f) \geq \max(n, m)$$

が成立する。

(証明) C_f が持つ入力端子数, 出力端子数をそれぞれ, w, w' とする。境界配置の仮定および PE は互いに重ならないことから

$$P \geq w + w' \quad (4.3)$$

が成立する。このとき, n 入力を読み込むのに $\lceil n/w \rceil$ 単位時間, m 出力を得るのに $\lceil m/w' \rceil$ 単位時間少なくとも必要になるので,

$$T(C_f) \geq \max(\lceil n/w \rceil, \lceil m/w' \rceil) \quad (4.4)$$

でなければならぬ。(4.3), (4.4) より

$$P \cdot T(C_f) \geq \max(n, m)$$

が成立する。 ■

以下では, まず冗長でない $(n, 1)$ -関数 (すなわち,

$F(m, n, 1, 1)$ に属する関数) を実現する回路の面積時間積の下界について考察する。 $F(m, N, m, M)$ の場合については、その結果を拡張して得られる。

有向木の有向辺の向きをすべて逆向きにして得られる有向グラフを r -有向木という⁽⁶⁾。 r -有向木は各葉節点から根への有向道をもつ。 各節点の入次数が高々 r の r -有向木を r 分木と呼ぶ。

[定義 4.1] $h \in F(m, n, 1, 1)$ に属する関数とする。 C_h に対する回路グラフ $G(C_h)$ を張る木とは、 $G(C_h)$ の入力節点を葉節点とし、出力節点を根とする r -有向木である。 $G(C_h)$ の各節点の入次数は高々 d ゆえ、 $G(C_h)$ を張る木は d 分木になる。 $G(C_h)$ を張る木のうちで高さ[†] 最小の d 分木を $S[G(C_h)]$ と表す。 ここで、 $S[G(C_h)]$ は $G(C_h)$ に対して一意には定まらないがどれを選んでよい。 図 4.1 に $G(C_h)$ と $S[G(C_h)]$ の例を示す。 ■

C_h のもつ入力端子数を w ($1 \leq w \leq n$) とする。 C_h が h を実現することから、 $G(C_h)$ の各入力節点から出力節点への有向道が少なくとも一つ存在する。 従って、 $S[G(C_h)]$ は葉数 w の d 分木になる。 定義 4.1 より、次の補題が成立つ。

[補題 4.2] $h \in F(m, n, 1, 1)$ とする。 C_h のもつ入力端子数を w とし、 $S[G(C_h)]$ の高さを H とする。 このとき、

† r -有向木の高さとは、葉節点から根への有向道の長さの最大値である。

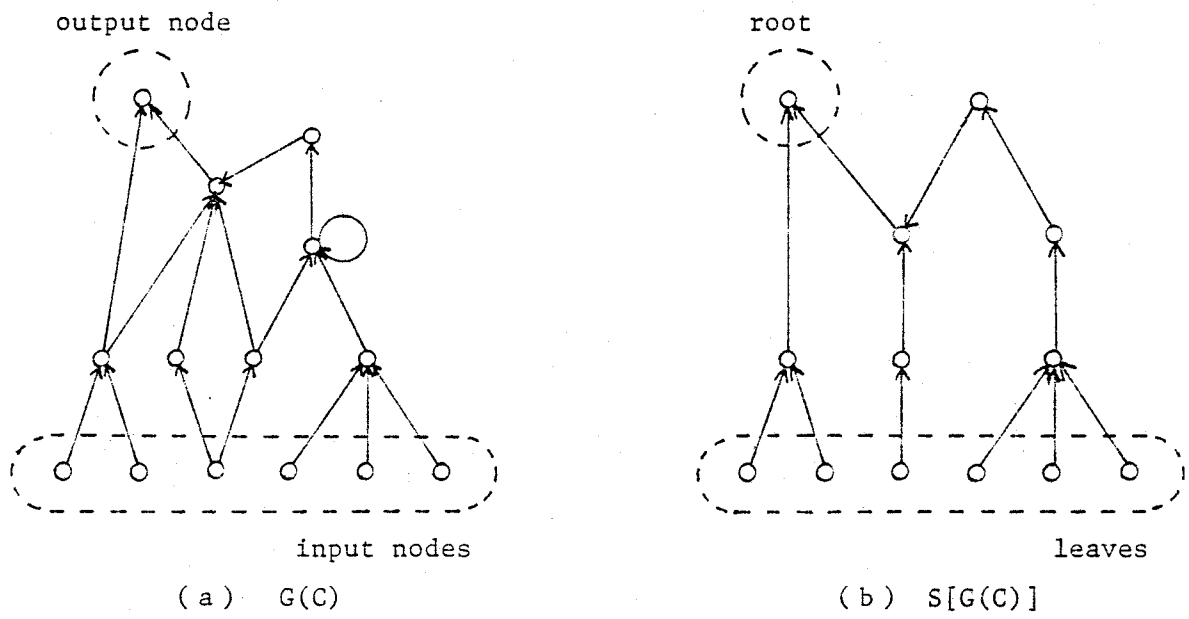


図4.1 $G(C)$ と $S[G(C)]$ の例.

Fig.4.1 An example of $G(C)$ and $S[G(C)]$.

$$A(C_h) \geq \text{area}(S[G(C_h)]),$$

$$T(C_h) \geq \max(H, \lceil n/w \rceil)$$

が成立つ。 ■

d 分木は、次に示す変換により2分木に変換される。
 d 分木の節点 v_1, \dots, v_k ($2 < k \leq d$)の子をきつとき、葉数長で高さ最小の2分木で置きかえる(図4.2)。
 この変換によって次の補題が成立つ。

[補題4.3] 高さ H_d の d 分木 G_d に対して、変換された2分木を G_2 とし、 G_2 の高さを H_2 とすると、

$$H_d \geq H_2 / (\lceil \log d \rceil + 1),$$

$$\text{area}(G_d) \geq \text{area}(G_2) / \gamma^2$$

が成立つ。ここで、 γ は定数である。

(証明) 高さに関しては変換から明らかである。面積に関しては次のようにして示すことができる。 G_d が領域 R に埋め込まれているとする。 G_d の k 次葉数長 ($2 < k \leq d$)の節点は、高さ最小で葉数長の2分木で置き換えられるが、この葉数長の2分木を図4.3のように配置する。図4.3の正方形の面積は高々 $9(2d+1)^2$ になる。

G_d の埋め込まれている領域 R を拡大して、拡大された G_d の各節点の領域に図4.3の2分木を埋込む。このとき、2分木は図4.3のように埋め込んであるので、根と葉節点は任意の場所から接続でき、 G_d の接続関係を保存できる。 G_d の変換される節点を埋め込むために必要な R の拡大率を γ とすると、 γ は d のみに依存するので定数となる。従って、

$$\text{area}(G_d) \geq \text{area}(G_2) / \gamma^2$$

が成立つ。 ■

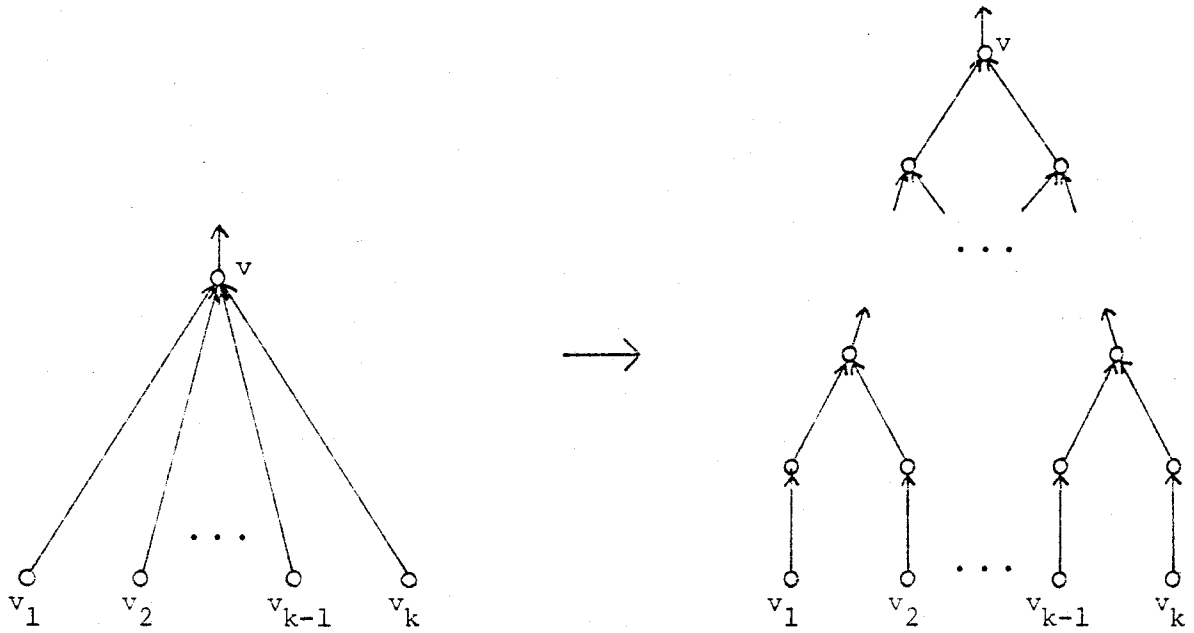


図4.2 d分木から2分木への変換

Fig.4.2 Transformation from d-ary tree to binary tree.

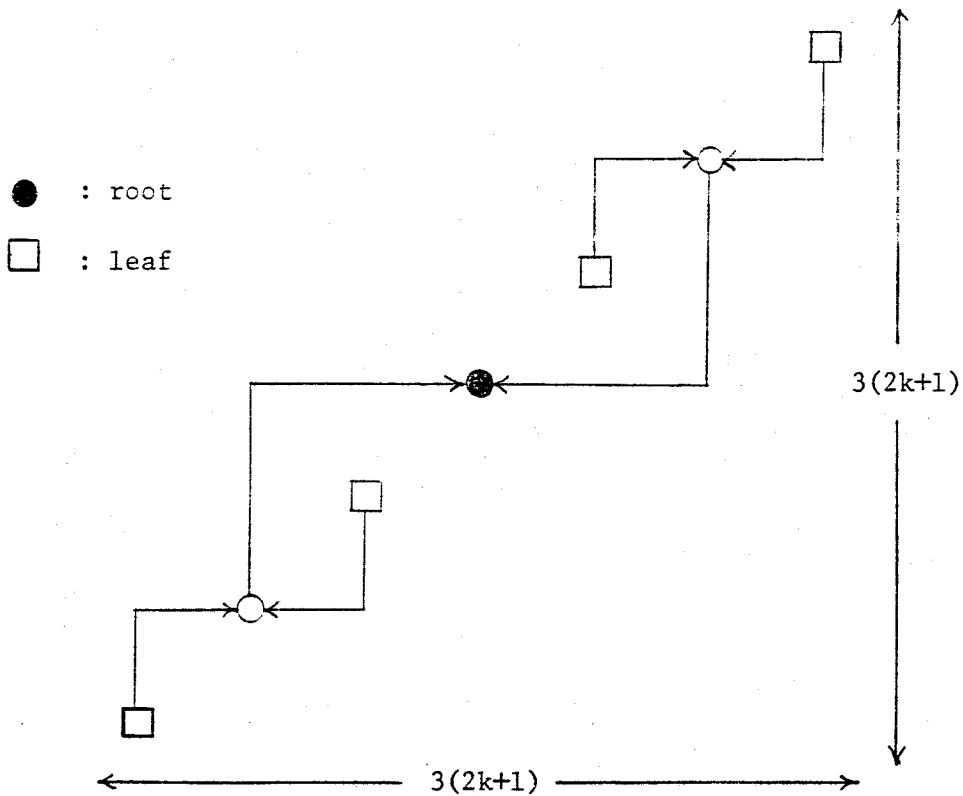


図4.3 変換された2分木の埋め込み (k=4).

Fig.4.3 Embedding of the transformed binary tree (k=4).

補題 4.2, 4.3 より, 回路 C_h の面積, 計算時間の下界は, $G(C_h)$ を張る木 $S[G(C_h)]$ を変換した二分木のそれぞれ面積, 高さを評価すればよい。

境界配置の仮定のもとで, 二分木を埋め込む面積に関して, 実効高度と呼ばれる二分木の複雑さを表す尺度と埋め込まれる領域の面積の関係が示されている⁽³⁴⁾。二分木 T の実効高度は次のように定義される。

T の各節点 v にラベルをつける。 v が葉節点ならば, ラベルを 0 とする。 v が葉節点でないとき, 2つの場合がある。 v がただ1つの子 v_1 をもつとき, v のラベルは v_1 のラベルに等しい。 v が2つの子 v_1, v_2 をもつとき, v のラベル ($\text{label}(v)$ と表す) は v_1, v_2 のラベルから次のように再帰的に定義される。

$$\text{label}(v) = \begin{cases} \text{label}(v_1) = \text{label}(v_2) \\ \text{then } \text{label}(v_1) + 1 \\ \text{else } \max(\text{label}(v_1), \text{label}(v_2)). \end{cases}$$

T の実効高度は, T の根のラベルで定義される。

定義より, 葉数 w の完全二分木の実効高度は高さに等しく $\lceil \log w \rceil$ である。一般に, 二分木の実効高度は, 二分木の有向辺を適当に短絡除去して得られる高さ最大の完全二分木の高さである⁽³⁵⁾。また, 葉数 w の二分木の実効高度を e とすると, $w > 1$ のとき,

$$1 \leq e \leq \lceil \log w \rceil$$

であり, $w = 1$ のときかつそのときに限り $e = 0$ となる。

[命題 4.2] ⁽³⁴⁾ 実効高度 e の 2 分木を凸領域 R に埋め込むとする。このとき、 R の任意の弦に対して、それに垂直な弦で長さ $|e/2|$ 以上のものが存在する。 ■

[定義 4.2] 高さ H 、実効高度 e をもつ 2 分木の集合を $B(H, e)$ とする。 $B(H, e)$ のなかで、葉節点数が最大となる 2 分木の葉数を $L(H, e)$ とする。 $B(H, e)$ のなかの 2 分木が w 個の葉節点をもつならば、

$$L(H, e) \geq w$$

が成立つ。葉数 w 、実効高度 e の 2 分木に対して、最小の高さをもつ木の高さを $H(w, e)$ と定義する。すなわち、

$$H(w, e) = \min \{ H \mid L(H, e) \geq w \}$$

とする。 ■

以上のことから、次の定理が成立つ。

[定理 4.1] $h \in F(n, m, 1, 1)$ とする。 C_h のもつ入力端子数を w とすると、

$$A(C_h) \cdot T(C_h)^\alpha = O(n \cdot \max(e, 1) \cdot \max[H(w, e), \lceil n/w \rceil]^{d-1})$$

($\alpha \geq 1$) が成立つ。ここで、 e は $S[G(C_h)]$ を変換した 2 分木の实効高度である。

(証明) $S[G(C_h)]$ を変換した 2 分木を G_2 とし、 G_2 を埋め込む凸領域 R の周長を P_2 とする。また、 R の直径に垂直な弦で命題 4.2 を満たすものが存在するので、その長さ

を L_2 とする。証明は、実効高度 e が $1 \leq e \leq \lceil \log w \rceil$ の場合と $e=0$ の場合に分けて行う。

(1) $1 \leq e \leq \lceil \log w \rceil$ の場合

命題 4.1, 4.2, 補題 4.1, 4.2, 4.3 より,

$$\begin{aligned}
 A(C_h) \cdot T(C_h) &\geq (\text{area}(E_2)/r^2) T(C_h) && \text{(補題 4.2, 4.3)} \\
 &\geq (L_2 P_2 / 2\pi r^2) T(C_h) && \text{(命題 4.1 (4.1))} \\
 &\geq L_2 \cdot n / 2\pi r^2 && \text{(補題 4.1)} \\
 &\geq n \cdot e / 4\pi r^2 && \text{(4.5)} \\
 &&& \text{(命題 4.2)}
 \end{aligned}$$

が成立つ。補題 4.2, 4.3, $H(w, e)$ の定義より,

$$\begin{aligned}
 T(C_h) &\geq \max(H, \lceil n/w \rceil) / (\lceil \log d \rceil + 1) \\
 &\geq \max(H(w, e), \lceil n/w \rceil) / (\lceil \log d \rceil + 1) \quad (4.6)
 \end{aligned}$$

が成立つ。(4.5), (4.6) より, $\alpha \geq 1$ のとき,

$$A(C_h) \cdot T(C_h)^\alpha = A(C_h) \cdot T(C_h) \cdot T(C_h)^{\alpha-1}$$

と変形すれば, 定理の式が成立つ。

(2) $e=0$ の場合は, $w=1$ のときに限られる。このとき, n 入力を読み込むのに n 単位時間かかるので,

$$A(C_h) \cdot T(C_h) \geq n^2$$

となり、この場合は明らかに定理の式は成立つ。 ■

4.2. 関数に対する面積時間積の下界

定理4.1は、 h を実現する回路 C_h が与えられたときに、その回路の面積時間積の下界を示している。従って、その回路の特徴を示すパラメータ w および e が式の右辺に現われている。ところが、関数 h に対する下界を示すには、 h を実現する任意の回路を評価しなければならない。換言すれば、定理4.1の右辺の式を w と e を変数とする関数とみたときの最小値を評価する必要がある。以下では、この関数に対する下界に関して考察する。

高さ H 、実効高度 e の二分木がもつ最大葉数 $L(H, e)$ については、 $H \geq e$ のとき、

$$L(H, e) = \sum_{i=0}^e \binom{H}{i}$$

が成立つ。この性質を用いて $e \cdot H(w, e)$ を評価する。

[補題4.4] 葉数 w の二分木の实効高度を e とする。
 $w \geq 4$ のとき、

$$e \cdot H(w, e) \geq \log^2 w / 2 \log \log w$$

が成立つ。

(証明) $w \geq 4$ であるから、 $e = 0$ とはなり得ない。従って、 $1 \leq e \leq \lfloor \log w \rfloor$ なる e について示せばよい。

$w \geq 4$ かつ $H(w, e) \geq e$ ならば、 $H(w, e) \geq 2$ となる。

一般に、二分木の高さを H とするとき、 $H \geq 2$, $H \geq e$ ならば、

$$\begin{aligned} L(H, e) &= \sum_{i=0}^e \binom{H}{i} \\ &\leq \sum_{i=0}^e H^i = (H^{e+1} - 1) / (H - 1) \\ &\leq (H / (H - 1)) \cdot H^e \\ &\leq 2 \cdot H^e \end{aligned}$$

が成立つ。従って、

$$w \leq L(\underline{H}(w, e), e) \quad (\leq 2 \cdot \underline{H}(w, e)^e)$$

となるためには、

$$\underline{H}(w, e) \geq (w/2)^{\frac{1}{e}} \quad (4.7)$$

でなければならぬ。

(1) $1 \leq e \leq \log w / 2 \log \log w$ の場合

$$(w/2)^{\frac{1}{e}} \geq \log^2 w / 2$$

となるので、 $e \geq 1$ と (4.7) より

$$e \cdot \underline{H}(w, e) \geq \log^2 w / 2$$

となる。 $\log \log w \geq 1$ であるから、

$$e \cdot \underline{H}(w, e) \geq \log^2 w / 2 \log \log w$$

が成立つ。

(2) $\log w / 2 \log \log w \leq e \leq \lceil \log w \rceil$ の場合
一般に,

$$\underline{H}(w, e) \geq \log w$$

であるので,

$$e \cdot \underline{H}(w, e) \geq \log^2 w / 2 \log \log w$$

が成立つ。 ■

定理 4.1 と補題 4.4 を用いると, h に対する面積時間積の 一つの下界が得られる。

[定理 4.2] h を $F(n, n, 1, 1)$ に属する関数とする。
 h に対して,

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

$$AT^\alpha = \Omega(n \log_a^\alpha n / \log_a \log_a n) \quad (\alpha \geq 2)$$

が成立つ。

(証明) $AT = \Omega(n)$ は明らかに成立つので, $\alpha \geq 2$ の場合を示す。 h を実現する回路が m 入力端子数 $\leq w$ とする。また,

$$R_\alpha = (AT^\alpha \log_a \log_a n) / (n \log_a^\alpha n)$$

とおく。

(1) $n^{\frac{1}{2}} < 4$ の場合は明らかに定理を満たす.

(2) $n^{\frac{1}{2}} \geq 4$ の場合, $n^{\frac{1}{2}} \geq 4 > w$, $n^{\frac{1}{2}} \geq w > 4$, $n \geq w \geq n^{\frac{1}{2}}$ に分けて証明する.

(2.1) $n^{\frac{1}{2}} \geq 4 > w$ の場合

$$A \geq w \geq 1, \\ T \geq \lceil n/w \rceil \geq n/4$$

となるので,

$$R_{\alpha} \geq \frac{n^{\alpha-1} \log_{\alpha} \log_{\alpha} n}{4^{\alpha} \log_{\alpha}^{\alpha} n} \\ \geq \frac{1}{4^{\alpha}} \left(\frac{n^{\frac{\alpha-1}{\alpha}}}{\log_{\alpha} n} \right)^{\alpha} \cdot \log_{\alpha} \log_{\alpha} n$$

が成立つ. $\delta > 0$ ならば, $n^{\delta} \geq n \log_{\alpha} n$ となる定数 η が存在する. $\delta = (\alpha-1)/\alpha$ とおけば, $\delta > 0$ 中え,

$$R_{\alpha} \geq \left(\frac{\eta}{4} \right)^{\alpha} \cdot \log_{\alpha} \log_{\alpha} n$$

となり, 定理が成立つ.

(2.2) $n^{\frac{1}{2}} \geq w > 4$ の場合, (2.1) と同様にして示される.

(2.3) $n \geq w \geq n^{\frac{1}{2}}$ の場合

定理 4.1, 補題 4.4 より,

$$AT^2 = \Omega(n \log_d^2 n / \log_d \log_d n)$$

が成立つ。また, $h \in F(n, n, 1, 1)$ 中え, 補題3.1より

$$T \geq \lceil \log_{d+1} n \rceil = \left\lceil \frac{\log_d n}{\log_d d+1} \right\rceil$$

である。従って, $d \geq 2$ ならば, ある定数 $c > 0$ に対して,

$$R_d = [(AT^2 \cdot \log_d \log_d n) / (n \cdot \log_d^2 n)] \cdot \left(\frac{T}{\log_d n} \right)^{d-2}$$

$$\geq c$$

が成立つので定理が成立つ。 ■

定理4.2は次のようにして, (n, N, m, M) -関数の場合に拡張される。

$f = (f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n))$ を $F(n, N, m, M)$ に属する関数とする。 f_i は N 個の冗長でない入力変数 x_{i1}, \dots, x_{iN} をもつ関数とし, x_j は M 個の関数 f_{j1}, \dots, f_{jm} に対して冗長でない入力変数とする。

f を実現する回路 C_f の回路グラフ $G(C_f)$ に対して, f_i に対応する出力節点を根とし, x_{i1}, \dots, x_{iN} に対応する入力節点を葉節点とする d 分木で高さ最小の $G(C_f)$ を張る木を $S[G(C_f)]_{f_i}$ とする。

また, $G(C_f)$ の有向辺をすべて逆向きにした有向グラフを $\tilde{G}(C_f)$ とする。 $\tilde{G}(C_f)$ に対して, x_j に対応する入力節点を根とし, f_{j1}, \dots, f_{jm} に対応する出力節点を葉節点とする d' 分木で高さ最小の $\tilde{G}(C_f)$ を張る木を $S[\tilde{G}(C_f)]_{x_j}$ とする。このように得られた $S[G(C_f)]_{f_i}$, $S[\tilde{G}(C_f)]_{x_j}$ に対

して、同様の議論をすることにより、次の定理を得る。

[定理 4.3] f を $F(n, N, m, M)$ に属する関数とする。
 f に対して、

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

$$AT^\alpha = \Omega(\max(n, m) \cdot \max(\log_d^\alpha N / \log_d \log_d N, \log_d^\alpha M / \log_d \log_d M)),$$

($\alpha \geq 2$) が成立つ。 ■

境界配置の仮定を要請しない場合は、定理 3.1 により
 f に対して、

$$AT^\alpha = \Omega(\max(n, m) [\max(\log_{d+1} N, \log_{d+1} M)]^{\alpha-1})$$

が成立つ。定理 4.3 の結果から、 $F(n, N, m, M)$ の関数の
クラスに対して、境界配置の仮定が面積時間複雑度に真
に影響をおよぼすことがわかる。

(注) Yao は、境界配置の仮定のもとで二分木を埋め込む
面積に関して、次の命題を証明なしで述べている。⁽³²⁾

[命題 4.3] 葉数 w 、高さ H の二分木を G とすると、

$$\text{area}(G) = \Omega(w \log w / \log(2H / \log w))$$

が成立つ。 ■

この命題が証明されると、この命題と上記の議論を用
いることにより、定理 4.3 の下界は改善される。

[予想] $f \in F(n, N, m, M)$ とする。このとき、

$$AT^\alpha = \Omega(\max(n, m) [\max(\log_d N, \log_d M)]^\alpha) \quad (\alpha \geq 2)$$

が成立つ。 ■

4.3. 関数に対する面積時間積の上界

ここでは、境界配置の仮定のもとで、いくつかの関数に対する面積時間積の上界を議論する。

3.2では、境界配置の仮定をおかない場合の回路構成を示した。表4.1には、論理積関数、比較、復号化関数に対して、面積時間積の下界および3.2で示した回路を境界配置を満たすように配置しなおした場合の面積時間積の上界を示す。表4.1からわかるように、単に3.2の回路を配置しなおすだけでは下界とかなりのギャップが残される。以下では、これらの関数に対して次のことを示す。

(1) n 入力論理積関数、2つの n ビット2進数の比較に対して、

$$AT = O(n),$$
$$AT^\alpha = O(n \log^\alpha n) \quad (\alpha \geq 2)$$

で実現できる回路が存在する。

(2) $(n, 2^n)$ -復号化関数に対して

$$AT^\alpha = O(2^n \cdot n^\alpha) \quad (\alpha \geq 0)$$

で実現できる回路が存在する。

すなわち、論理積関数、比較に対して、定理4.3の下界は、 AT に関しては最良であり、 AT^α ($\alpha \geq 2$)に関しては $\log \log n$ の範囲を除いて最良である。また、復号化関数に対しては、 AT^α ($\alpha \geq 2$) に関して $\log n$ の範囲を除

表 4.1 境界配置を満たす場合の関数の面積時間複雑度.

関数 (入力数/出力数)	下 界		上 界	
	AT	$AT^\alpha (\alpha \geq 2)$	AT	$AT^\alpha (\alpha \geq 2)$
論理積関数 ($n/1$)	$\Omega(n)$	$\Omega\left(\frac{n \log^\alpha n}{\log \log n}\right)$	$O(n \log n)$	$O(n \log^\alpha n)$
2進数の比較 ($2n/2$)	$\Omega(n)$	$\Omega\left(\frac{n \log^\alpha n}{\log \log n}\right)$	$O(n \log n)$	$O(n \log^\alpha n)$
復号化関数 ($n/2^n$)	$\Omega(2^n)$	$\Omega\left(\frac{2^n \cdot n^\alpha}{\log n}\right)$	$O(2^n \cdot n^2)$	$O(2^n n^{\alpha+1})$

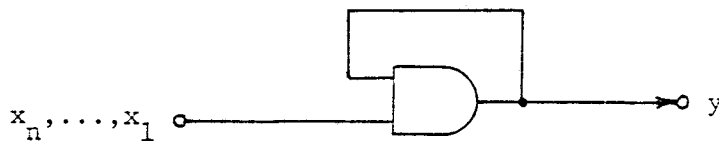


図 4.4 $AT=O(n)$ となる n ビット論理積関数を実現する回路.

Fig.4.4 Circuit to compute n -bit AND function with $AT=O(n)$.

いて最良である。

[定理 4.4] n 入力論理積 (論理和, 排他的論理和) 関数, 2^n の n ビット 2 進数の比較に対して,

$$AT = \Theta(n),$$

$$AT^\alpha = \Omega(n \log^\alpha n / \log \log n)$$

$$AT^\alpha = O(n \log^\alpha n) \quad (\alpha \geq 2)$$

が成立つ。

(証明) 定理 3.5 と同様にして n 入力論理積関数に対して示せばよい。 AT , AT^α ($\alpha \geq 2$) の下界は定理 4.3 より得られる。 AT^α ($\alpha \geq 2$) の上界は定理 3.5 の回路を境界配置の仮定を満たすように配置しなおすことにより得られる。 AT の上界は次のようにして示される。

図 4.4 の回路を構成し, n 個の入力を直列に順次論理積をとってゆく。このとき,

$$A = O(1),$$

$$T = O(n)$$

となり, $AT = O(n)$ が成立つ。 ■

[定理 4.5] $(n, 2^n)$ -復号化関数に対して,

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

$$AT^\alpha = \Omega(2^n \cdot n^\alpha / \log n) \quad (\alpha \geq 2)$$

$$AT^\alpha = O(2^n \cdot n^\alpha) \quad (\alpha \geq 0)$$

が成立つ。

(証明) 下界は定理 4.3 より得られる。上界は次のようにして示される。

まず, $(k, 2^k)$ -復号器 $D(k)$ を,

$$A(D(k)) = O(2^k \cdot k)$$

$$T(D(k)) = O(k)$$

で実現できることを示す. 図4.5(a)に $D(k)$ の構成を示す. $D(k)$ は分岐セル BC, 終端セル TC から構成される. BC, TC の状態遷移表をそれぞれ図4.5(b), (c)に示す.

各セルの初期状態を I とする. 各セルの計算時間を t とすると, 入力はシフトレジスタから t 単位時間ごと順次与えられる. $D(k)$ の動作例を図4.6に示す. 各出力は $2k$ t 時間後に得られるので

$$T(D(k)) = O(k)$$

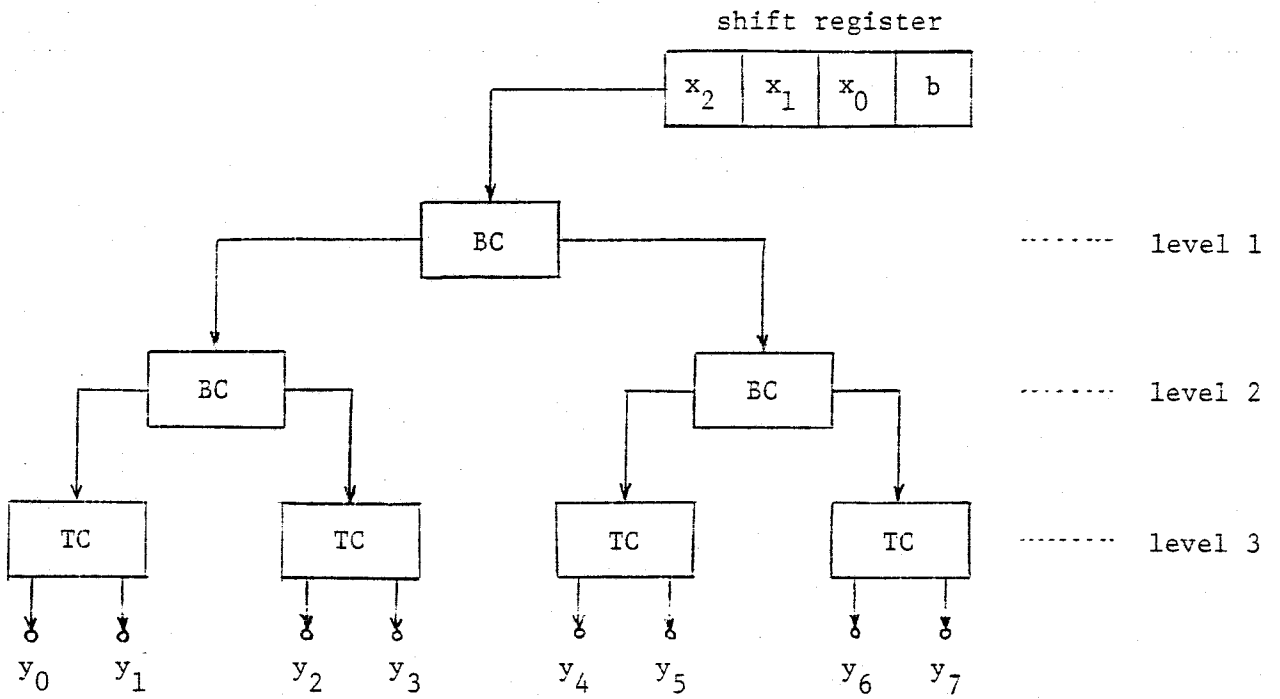
となる. また, BC, TC はいずれも定数面積で実現できるので,

$$A(D(k)) = O(2^k \cdot k)$$

を満たす.

次に, 定理3.3と同様にして, $D(n)$ を $D(w_1)$ と $D(w_2)$ から図4.7のように構成する. ここで $w_1 + w_2 = n$ である. $D(n)$ の各出力は, $D(w_1)$ の結果と $D(w_2)$ の結果のそれぞれ論理積をとることにより得られる. $D(w_1)$ の結果はシフトレジスタに貯えられ, $D(w_2)$ の結果はレジスタに貯えられる. また, シフトレジスタの先頭から木状

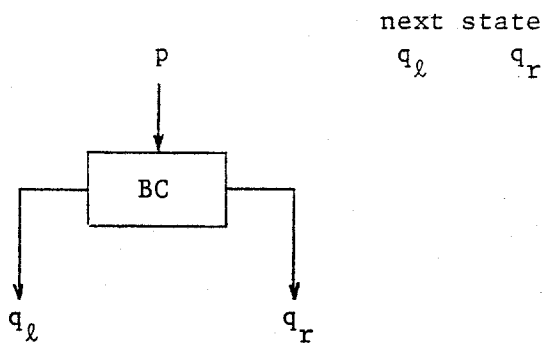
+ 簡単のため, BC, TC の計算時間は同じと仮定する.



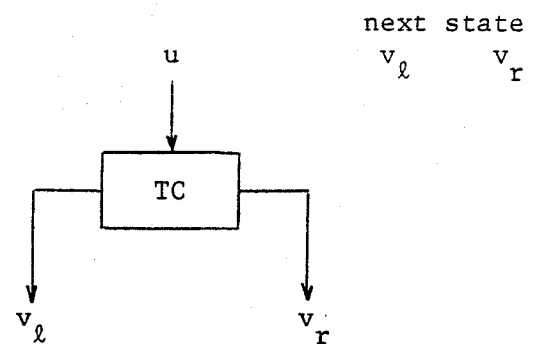
(a)

state \ P	b	0	1
I	I b b	L b b	R b b
R	I b b	L b 0	R b 1
L	I b b	L 0 b	L 1 b

state \ u	b	0	1
I	I 0 0	L 0 0	R 0 0
R	I 0 1	—	—
L	I 1 0	—	—



(b)



(c)

図4.5 $D(k)$ の構成.

Fig.4.5 Construction of $D(k)$.

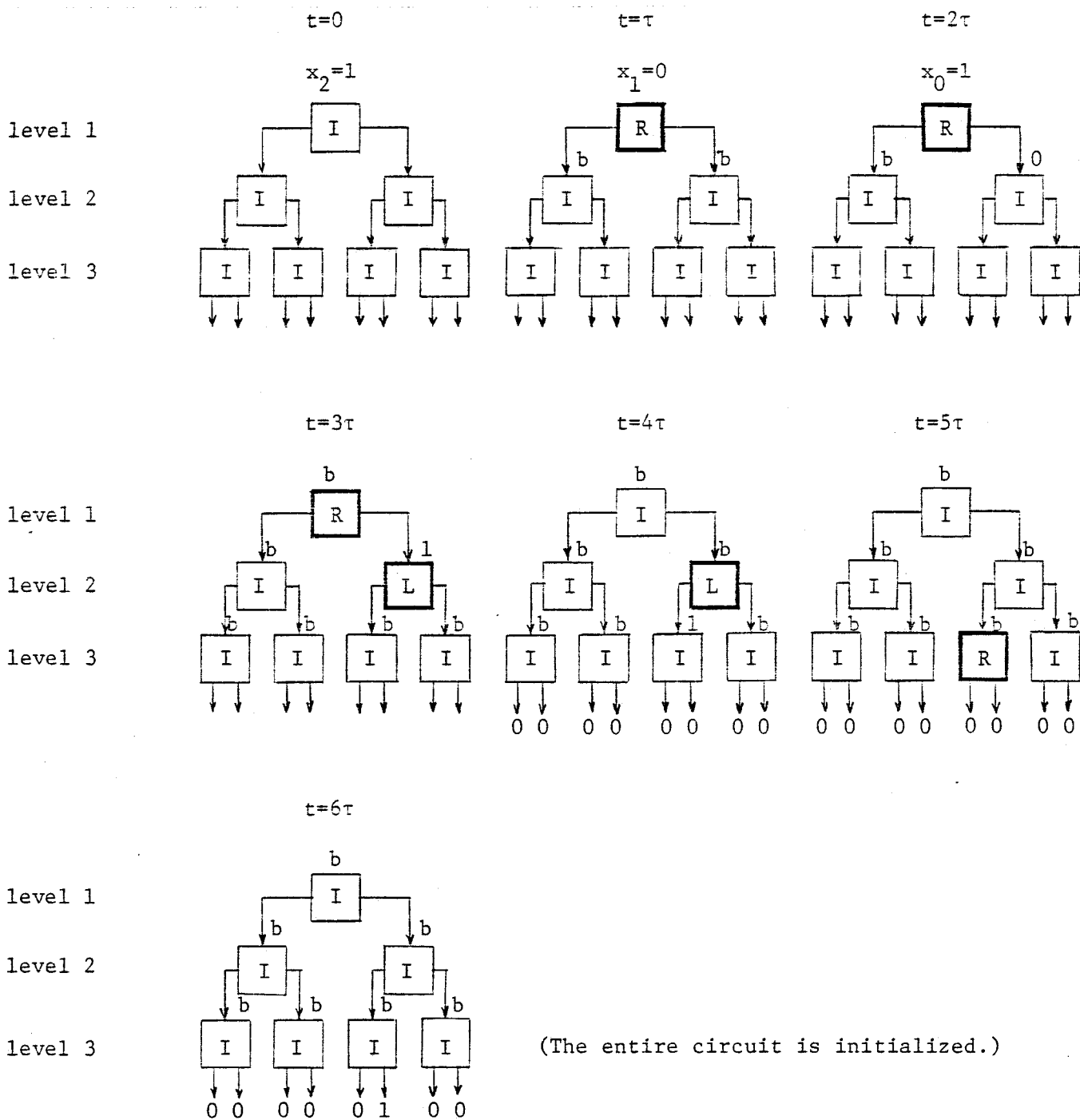


図4.6 $D(k)$ の動作例 ($x_0=1, x_1=0, x_2=1$ の場合).

Fig.4.6 Behaviour of $D(k)$ (the case that $x_0=1, x_1=0$ and $x_2=1$).

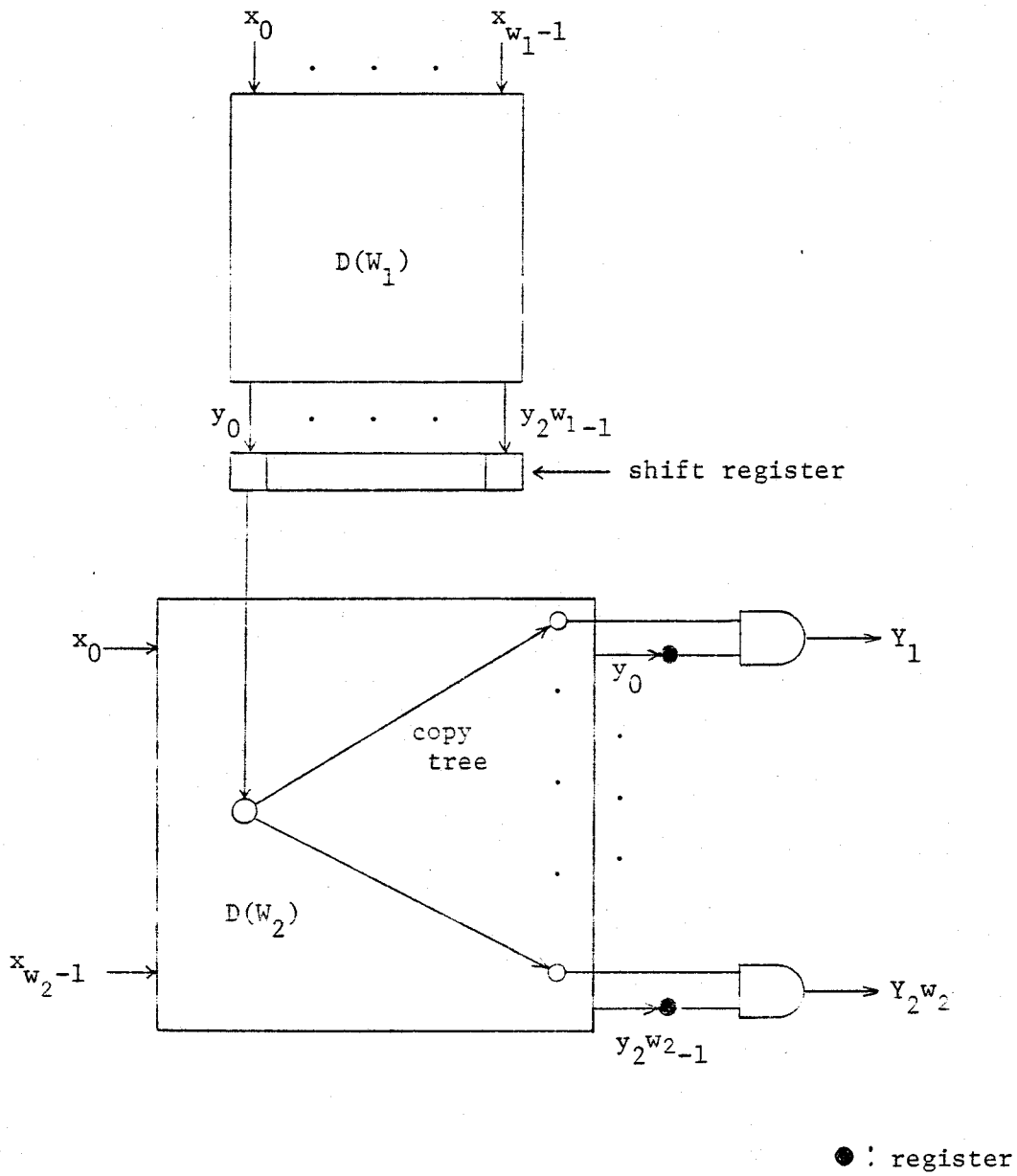


図4.7 $AT^\alpha = O(2^n \cdot n^\alpha)$ とする $D(n)$ の構成.

Fig.4.7 Construction of $D(n)$ with $AT^\alpha = O(2^n \cdot n^\alpha)$.

構造の回路により 2^{W_2} のコピーを作り (この木状回路をコピー木と呼ぶ), $D(W_2)$ の結果との論理積をとる.

$D(W_1)$ の結果はシフトレジスタに貯えられているので各結果を1単位時間遅れでコピー木に入力すると, $D(n)$ の計算時間は,

$$T(D(n)) = O(\max(t_1, t_2) + 2^{W_1} + W_2) \quad (4.8)$$

となる. ここで, t_1, t_2 はそれぞれ $D(W_1), D(W_2)$ の計算時間である. $D(W_1), D(W_2)$ とともに図4.5(a)の構成を用いると, (4.8) は

$$T(D(n)) = O(2^{W_1} + W_2) \quad (4.9)$$

となる. また, 面積については,

$$\begin{aligned} A(D(W_1)) &= O(2^{W_1} \cdot W_1), \\ A(D(W_2)) &= O(2^{W_2} \cdot W_2) \end{aligned}$$

であり, コピー木を $D(W_2)$ に重ねるように配置すると

$$A(D(n)) = O(2^{W_1} \cdot W_1 + 2^{W_2} \cdot W_2) \quad (4.10)$$

が成立つ. $W_1 = \lceil \log n \rceil, W_2 = n - \lceil \log n \rceil$ に設定すると (4.9), (4.10) より,

$$A(D(n)) \cdot T(D(n))^2 = O(2^n \cdot n^2) \quad (2.7.0)$$

が成立つ. ■

5. 面積複雑度 (1) —— 順序回路の場合 ——

本章では、 $F(n, m)$ に属する関数を実現する回路に対して、順序回路での実現を許したとき、必要となる回路面積を評価する手法を議論する。ここで得られる面積の下界は、計算時間を考慮しないときにその関数を実現するために必要となる最小の面積となる。すなわち、ここで示した面積より小さいオーダではその関数は実現できない。また、復号化関数、符号化関数などはそれらの面積下界を達成できる。

5.1. 回路に対する面積下界

$F(n, m)$ に属する関数を $(y_1, \dots, y_m) = f(x_1, \dots, x_n)$ とする。

[定義 5.1] $X = (x_1, \dots, x_n)$ とする。 X の部分系列を $X_1 = (x_{i_1}, \dots, x_{i_\ell})$ とし、 $X - X_1 = (x_{h_1}, \dots, x_{h_{n-\ell}})$ とする。 $Q_1 \in \{0, 1\}^\ell$ 、 $Q_2 \in \{0, 1\}^{n-\ell}$ としたとき、 f に対して、 $f(Q_1 | X_1, Q_2 | X - X_1)$ は、

$x_{ij} \in S(X_1)$ ならば、 f の x_{ij} に Q_1 の第 j 番目の要素を、
 $x_{hj} \in S(X - X_1)$ ならば、 f の x_{hj} に Q_2 の第 j 番目の要素をそれぞれ代入したときの関数値を表す。

$Y = (y_1, \dots, y_m)$ とする。 Y の部分系列を、 $Y_1 = (y_{j_1}, \dots, y_{j_k})$ とする。 f に対して、 $f_{Y_1}(x_1, \dots, x_n)$ は、 $(f_{j_1}(x_1, \dots, x_n), \dots, f_{j_k}(x_1, \dots, x_n))$ を表すとする。 ■

[定義 5.2] 入力変数系列 X の部分系列 X_1 に対して、 $|S(X_1)| = \ell$ とする。このとき、 X_1 によって値の定まる

出力変数の集合を次のように定義する。

$$y_i \in Y(X_1) \iff \forall Q_1 \in \{0,1\}^l \forall Q_2, Q_2' \in \{0,1\}^{n-l} \\ [f_i(Q_1 | X_1, Q_2 | X - X_1) = f_i(Q_1 | X_1, Q_2' | X - X_1)].$$

すなわち、 $Y(X_1)$ の出力変数の値は、 X_1 の入力変数の値が決まれば、 $X - X_1$ の入力変数の値に依存せずに常に確定する。一方、 $S(Y) - Y(X_1)$ ($\overline{Y(X_1)}$ と表す)の出力変数の値は、 X_1 の入力変数の値だけで確定しなり場合があり $X - X_1$ の入力変数の値に依存する。 ■

(n, m) -関数 f を実現する回路の計算過程において、 n 個の入力変数のうち、ある時刻 t までに回路にすでに読み込まれた入力変数の集合を $R(t)$ 、まだ値が読み込まれていない入力変数の集合を $NR(t)$ とする。仮定 (A-6) により、任意の時刻 t に対して、

$$R(t) \cap NR(t) = \emptyset, \\ R(t) \cup NR(t) = S(X)$$

である。従って、 $R(t) = S(X_1)$ とすると、 $NR(t) = S(X - X_1)$ となる。このとき、 $Y(X_1)$ の出力変数は、 $X - X_1$ の値にかかわらず、時刻 t において出力できる。ところが、 $\overline{Y(X_1)}$ の出力変数の値は、 $X - X_1$ の値に依存するので、異なる X_1 の値 Q_1, Q_1' に対して、

$$f_{\overline{Y(X_1)}}(Q_1 | X_1, Q_2 | X - X_1) \neq f_{\overline{Y(X_1)}}(Q_1' | X_1, Q_2 | X - X_1)$$

となる $X - X_1$ の値 Q_2 が存在する。従って、時刻 t において、 $\overline{Y(X_1)}$ の出力変数は出力できない。また、 Q_2 によっ

て出力変数の値が異なるので、回路は七の時点で Q_1, Q_1' の値を区別しなければならない。次に示す関係は、 $\overline{Y(X_1)}$ の値を決定するために区別しなければならない X_1 の値を指定するものである。

[定義 5.3] X_1 を X の部分系列とし、 $|S(X_1)| = l$ とする。 X_1 に対して $\{0, 1\}^l$ の上の関係 $\varepsilon(X_1)$ を次のように定義する。 $Q_1, Q_2 \in \{0, 1\}^l$ であるとき、

$$(Q_1, Q_2) \in \varepsilon(X_1) \iff \forall Q \in \{0, 1\}^{m-l} \\ [f_{\overline{Y(X_1)}}(Q_1 | X_1, Q | X - X_1) \\ = f_{\overline{Y(X_2)}}(Q_2 | X_1, Q | X - X_2)].$$

この関係 $\varepsilon(X_1)$ は同値関係であることが容易に確かめられるので、それによって決まる同値類の個数を $|\varepsilon(X_1)|$ と表し、

$$I_f(X_1) = \lceil \log |\varepsilon(X_1)| \rceil$$

と定義する。 ■

関数 f を回路で実現したとき、同値関係 $\varepsilon(X_1)$ が成立たない 2 つの X_1 の値 $Q_1, Q_2 \in \{0, 1\}^l$ に対して、回路はこの 2 つの入力値を区別しなければならない。従って、 $|\varepsilon(X_1)|$ は X_1 の値が回路に与えられた時点で、 $\overline{Y(X_1)}$ の値を定めるために区別する必要がある状態数に、 $I_f(X_1)$ はその状態数を識別するために必要なビット数に相当する。

[例 5.1] 次に示す関数について、 $|\varepsilon(X_1)|$ および $I_f(X_1)$ を求める。ここで考える関数は n 入力 x_1, \dots, x_n のうち 1 となる変数の個数がちょうど $\lceil n/2 \rceil$ に等しいとき 1 を

出力し、そうでないとき 0 を出力する $(n, 1)$ -関数 $y = f(x_1, \dots, x_n)$ である。 f は対称関数ゆえ、入力変数の部分系列 X_1 をとったとき、 $S(X_1)$ に含まれる入力変数の添字には依存せず、入力変数の個数 $|S(X_1)|$ により、 $|E(X_1)|$, $I_f(X_1)$ が一意に定まる。

このとき、 $|S(X_1)| = l$ ($0 \leq l \leq n-1$) となる X_1 に対しては、

$$Y(X_1) = \emptyset, \quad \overline{Y(X_1)} = \{y\}$$

であり、 $|S(X_1)| = n$ のとき、

$$Y(X_1) = \{y\}, \quad \overline{Y(X_1)} = \emptyset$$

である。

図 5.1 に、 $n = 8$ の場合について、 $|E(X_1)|$ および $I_f(X_1)$ の変化を入力による状態遷移によって表したものを示す。 図 5.1 において、各状態は入力に含まれる 1 の個数を表し、矢印につけられたラベルは入力値を示す。ここで、 $|S(X_1)| = i$ における縦一列に含まれる状態数が $|E(X_1)|$ に相当する。

一般の n の場合は次のようになる。

$$|S(X_1)| = 0 \text{ のとき, } |E(X_1)| = 0, \quad I_f(X_1) = -\infty.$$

$$|S(X_1)| = l \quad (1 \leq l \leq \lceil n/2 \rceil) \text{ のとき,}$$

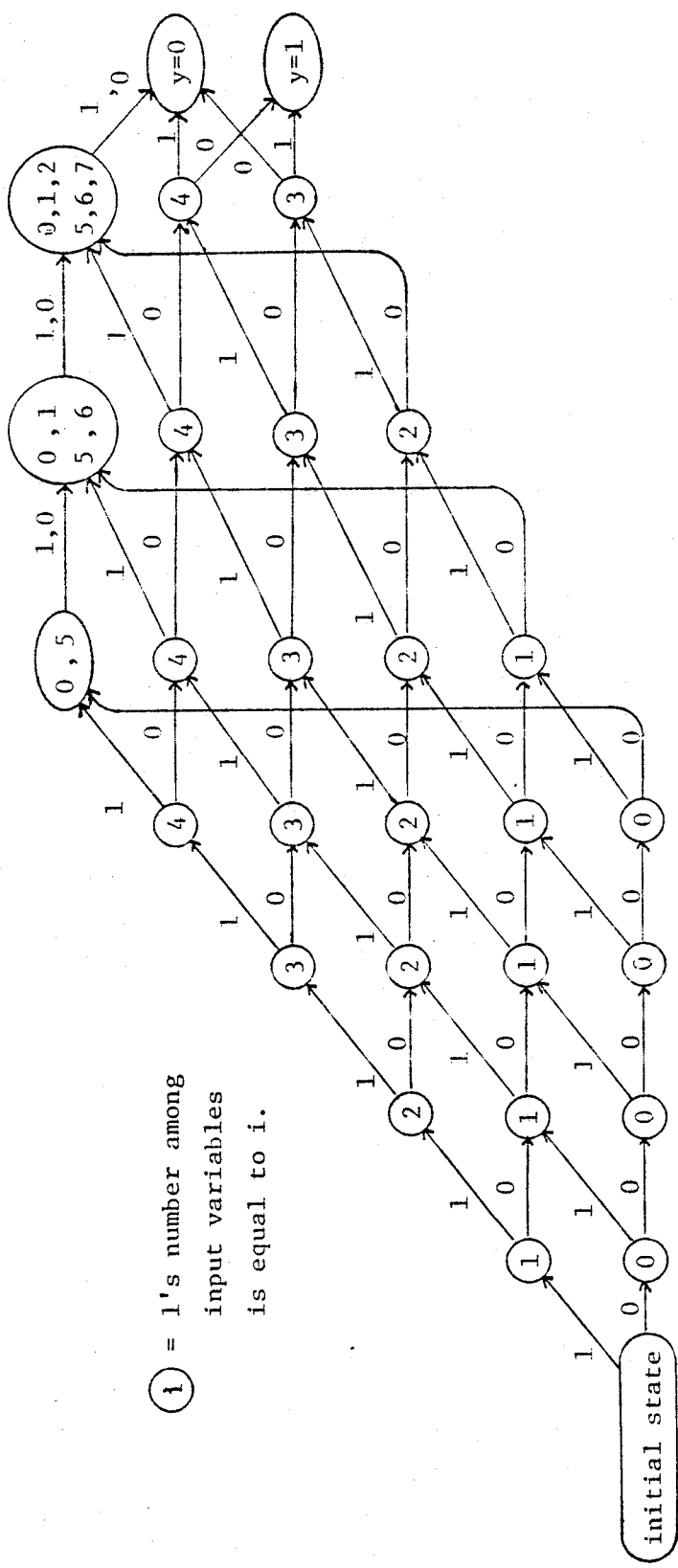
$$|E(X_1)| = l+1, \quad I_f(X_1) = \lceil \log(l+1) \rceil.$$

$$|S(X_1)| = \lceil n/2 \rceil + l \quad (1 \leq l \leq n - \lceil n/2 \rceil - 1) \text{ のとき,}$$

$$|E(X_1)| = \lceil n/2 \rceil - l + 2,$$

$$I_f(X_1) = \lceil \log(\lceil n/2 \rceil - l + 2) \rceil.$$

$$|S(X_1)| = n \text{ のとき, } |E(X_1)| = 0, \quad I_f(X_1) = -\infty. \quad \blacksquare$$



$ S(X_1) $	0	1	2	3	4	5	6	7	8
$ \epsilon(X_1) $	0	2	3	4	5	5	4	3	0
$I_f(X_1)$	$-\infty$	1	2	2	3	3	2	2	$-\infty$

図5.1 例5.1の関数の $|\epsilon(X_1)|$, $I_f(X_1)$ の変化 ($n=8$).

Fig.5.1 (X_1) and $I_f(X_1)$ for the function in Example 5.1 ($n=8$).

VLSIモデルの仮定(A-7)より、実現される回路は、入力を与える順序と時刻、および入出力端子の割当ては固定されており、ある時刻を基準として、入力を与えて出力が得られる時刻も、入力値に依らず一定である。従って、 f を実現する回路において、ある時刻までに X_1 の値が読み込まれたとき($R(t) = S(X_1)$)、 X_1 の入力値だけでは値の確定しな $\overline{Y(X)}$ のどの出力変数も出力できない。また、各入力データは1回だけ読み込まれる(仮定(A-6))ので、その時点で回路は少なくとも $I_f(X_1)$ ビットを記憶しなければならぬ。このことから、 f を実現する回路の面積について次の定理が成立つ。

[定理5.1] $(y_1, \dots, y_m) = f(x_1, \dots, x_n)$ を実現する回路 C について、 C に一度に入力される入力変数の個数の最大値を p とする。この回路において、入力が与えられる時刻を t_1, \dots, t_g とし、 $t_j (1 \leq j \leq g)$ までに入力された変数の系列を X_j 、すなわち、 $R(t_j) = S(X_j)$ とすると、

$$A \geq \max(p, I_f(X_1), \dots, I_f(X_g))$$

が成立つ。

(証明) C に一度に入力される入力変数の個数の最大値が p ゆえ、 C は少なくとも p 個の入力端子をもつ。仮定(A-3)より、

$$A \geq p \tag{5.1}$$

が成立つ。

ある $R(t_j)$ に対して $A < I_f(X_j)$ と仮定する。ここで、 $|S(X_j)| = l_j$ とする。この時点で回路が記憶できる状態数

は高々 2^A であり, $|E(X_j)|$ より少なくなる. 従って, 同値関係 $E(X_j)$ において, 異なる同値類に属する X_j の値 $Q_1, Q_2 \in \{0, 1\}^k$ を C は同じ状態で記憶している. すなわち, C は次の条件を満たす異なる X_j の値 Q_1, Q_2 を同じ状態で記憶しなければならない.

$$\exists Q \in \{0, 1\}^{n-k_j} [f_{\overline{Y}(X_j)}(Q_1 | X_j, Q | X - X_j) \neq f_{\overline{Y}(X_j)}(Q_2 | X_j, Q | X - X_j)].$$

ゆえに, C は異なる出力をしなければならない X_j の値を区別できないことになり, f を正しく実現できない. 従って, すべての j ($1 \leq j \leq g$) に対して

$$A \geq I_f(X_j) \quad (5.2)$$

でなければならない. (5.1), (5.2) より次式が成立つ.

$$A \geq \max(p, I_f(X_1), \dots, I_f(X_g)).$$

関数 f を実現する回路の面積の下界を議論するためには, f を実現する任意の回路を対象にしなければならない. 一方, 定理 5.1 は f を実現するある回路について, その回路面積 A と, 入力端子数 p および $I_f(X_j)$ の関係を与えている. そこで, 面積の下界は f を実現する任意の回路を, その回路のもつ入力端子の個数 p によって分類して議論すればよい. 例えば, 関数 f に対する面積下界 $A = \Omega(g_f)$ を示す場合, $p \geq g_f$ となる回路は明らかにこの面積下界を満たす. また, $g_f > p$ となる場合, 一般には $n \geq g_f$ となるので, 入力をいくつか分割して入力する回路になる. このとき, $R(t) = S(X_t)$ ($S(X_t) \in S(X)$)

となる時刻 t が存在し、定理 5.1 から $I_f(X_1)$ が回路の面積を支配するので、 $I_f(X_1) \approx g_f$ となることを示せばよい。ただし、一般の場合は、 X_1 の選び方によって $I_f(X_1)$ は異なるので、 $|S(X_2)| = l$ となる X_1 についての最小値を用いる。

5.2. 関数に対する面積下界

定理5.1を用いて、重み関数、復号化関数、符号化関数、 n ビット2進教長値の加算などに対する面積下界を与える。まず、各関数の $I_f(X_1)$ を求める。

[補題5.1] X_1 を入力変数系列 X の部分系列とし、 $|S(X_1)| = l$ とする。 $(n, \lceil \log(n+1) \rceil)$ -重み関数、 $(n, 2^n)$ -復号化関数、 $(n, \lceil \log n \rceil)$ -符号化関数、 n ビット2進教長値の加算をそれぞれ、 f_w, f_d, f_e, f_a とする。このとき

$$(a) \min_{|S(X_1)|=l} I_{f_w}(X_1) = \lceil \log(l+1) \rceil \quad (1 \leq l \leq n-1),$$

$$(b) \min_{|S(X_1)|=l} I_{f_d}(X_1) = l \quad (1 \leq l \leq n-1),$$

$$(c) \min_{|S(X_1)|=l} I_{f_e}(X_1) = \max(\lceil \log l \rceil, 1) \quad (1 \leq l \leq \lceil n/2 \rceil - 1),$$

$$(d) \min_{|S(X_1)|=l} I_{f_a}(X_1) = \lceil \log(l+1) \rceil \quad (1 \leq l \leq l-1)$$

がそれぞれ成立つ。

(証明) (a) $(n, \lceil \log(n+1) \rceil)$ -重み関数

$|S(X_1)| = l$ ($1 \leq l \leq n-1$) なる入力変数系列 X_1 が与えられたとき、 $Y(X_1) = \phi$ となり、どの出力変数の値も $X-X_1$ の値に依存する。また、各出力変数は入力中に現れる1の個数にのみ依存し、入力変数の添字には依存しない。すなわち、対称関数である。従って、入力変数のどの組合せについてもその個数で議論すればよく、以下の議論は、 $S(X_1) = l$ となる任意の X_1 について成立つ。

$S(X_1) = l$ ($1 \leq l \leq n-1$) のとき、 $Q_1, Q_2 \in \{0, 1\}^l$

なる入力に対して、 Q_1, Q_2 に現れる1の個数が異なれば
 任意の追加入力 $Q \in \{0, 1\}^{n-l}$ について、出力値が異なる
 ので、

$$(Q_1, Q_2) \in \varepsilon(X_1)$$

である。従って、1の個数のとり得る値の数が同値類の
 個数になる。すなわち、

$$|\varepsilon(X_1)| = l + 1$$

である。よって、

$$\min_{|S(X_1)|=l} I_{f_w}(X_1) = \lceil \log(l+1) \rceil \quad (1 \leq l \leq n-1)$$

が成立す。

(b) $(n, 2^n)$ -復号化関数

$1 \leq l \leq n-1$ のとき、 $|S(X_1)| = l$ となるどのような
 X_1 についても、入力の組の可能な組合せは 2^l である。
 このとき、どの出力変数の値も $X - X_1$ の値に依存する
 ($Y(X_1) = \emptyset$ である)。また、異なる $Q_1, Q_2 \in \{0, 1\}^l$
 については、任意の追加入力 $Q \in \{0, 1\}^{n-l}$ に対して、
 出力値が異なるので、

$$(Q_1, Q_2) \in \varepsilon(X_1)$$

である。従って、 $|S(X_1)| = l$ となる任意の X_1 について、

$$|\varepsilon(X_1)| = 2^l$$

となり

$$\min_{|S(X_1)|=l} I_{fd}(X_2) = l \quad (1 \leq l \leq n-1)$$

が成立つ。

(c) $(n, \lceil \log n \rceil)$ -符号化関数

各出力変数が依存する入力変数の個数は、高々 $\lceil n/2 \rceil$ ゆえ、 $1 \leq l \leq \lceil n/2 \rceil - 1$ ならば、 $|S(X_1)| = l$ なる任意の X_1 に対して、どの出力変数の値も (少なくとも $\lceil n/2 \rceil + 1$ ある) $X - X_1$ の値に依存する。

また、任意の X_2 について、高々 1 つの入力変数が 1 になるので、 X_1 の入力の可能な組合せは $l+1$ とおりである。ここで、符号化関数の最下位桁入力 x_0 に依存する出力変数は存在しないので、 X_1 に x_0 が含まれるか否かにより場合を分けて考える。

X_2 に x_0 が含まれない場合は、 $l+1$ とおりの場合について、任意の追加入力 $Q \in \{0, 1\}^{n-l}$ に対して出力値が異なるので、

$$|\varepsilon(X_1)| = l + 1$$

となる。

X_1 に x_0 が含まれる場合は、 l とおりの場合について、任意の追加入力 $Q \in \{0, 1\}^{n-l}$ に対して出力値が異なるので、

$$|\varepsilon(X_1)| = l$$

となる。従って、

$$\min_{|S(X_1)|=l} I_{f_e}(X_1) = \min(\lceil \log l \rceil, 1)$$

が成立つ。

(d) 長個の n ビット 2 進数の加算

長 n 個の入力変数を n ビット 2 進数として $x_{(n-1)i} \cdots x_{0i}$ ($1 \leq i \leq k$) とする。 $|S(X_1)| = l$ ($1 \leq l \leq k-1$) となる任意の X_1 について、どの出力変数の値も $X-X_2$ の値に依存する。

今、 $X_1 = (x_{i_1 j_1}, \dots, x_{i_l j_l})$ とする。この l 個の変数で作られる 2 進数を $\text{bin}(X_1)$ で表すと、

$$\text{bin}(X_1) = \sum_{h=1}^l x_{i_h j_h} 2^{i_h}$$

となる。 $Q_1, Q_2 \in \{0, 1\}^l$ のとき、 $\text{bin}(Q_2 | X_1)$; $(\text{bin}(Q_2 | X_1))$ と $x_{i_h j_h}$ に $Q_1(Q_2)$ の第 h 番目の値を代入したものとす。このとき、 $\text{bin}(Q_1 | X_1)$, $\text{bin}(Q_2 | X_2)$ の値が異なれば、出力が異なるような $Q \in \{0, 1\}^{kn-l}$ が存在する。すなわち、

$$\text{bin}(Q_1 | X_1) \neq \text{bin}(Q_2 | X_1) \iff (Q_1, Q_2) \in \mathcal{E}(X_1)$$

である。従って、 $\text{bin}(Q_1 | X_1)$ のとり得る個数が $|\mathcal{E}(X_1)|$ となる。

さて、 X_1 の l 個の入力変数のうち、2 進数の i ビット目 ($1 \leq i \leq n$) に対応する変数の個数を p_i とすると

$$\sum_{i=1}^n p_i = l \quad (0 \leq p_i \leq k) \quad (5.3)$$

である。1 ビット目の p_1 個で作られる値は 0 から p_1 で、

その個数は p_1+1 とおりである。1ビット目で作られる値の最大値を g_1 とする。2ビット目で作られる値のうち0を除く p_2 とおり ($2, 4, \dots, 2p_2$) を g_1 に加えて得られる値はいずれも異なる。従って、

$$\begin{aligned} & \text{(1ビット目と2ビット目で作られる値の個数)} \\ & \geq 1 + p_1 + p_2 \end{aligned}$$

が成立つ。以下、同様にすると、1ビット目から n ビット目で作られる値の個数 ($|E(X_1)|$) について、

$$|E(X_1)| \geq 1 + \sum_{i=1}^n p_i$$

が成立つ。(5.3) と $\min_{|S(X_1)|=l} I_f(X_1)$ の定義により、

$$\min_{|S(X_1)|=l} I_{f_a}(X_1) = \lceil \log(l+1) \rceil$$

である。 ■

ここに示した関数は、どの出力変数も出力できないような入力変数の個数 l については、いずれも $I_f(X_1)$ は単調増加になっているが、一般の場合は必ずしもこのようにはならない。例5.1の関数はその例となっている。この補題と定理5.1を用いると、各関数を実現する回路の面積下界が次の定理によって示される。

[定理5.2] $(n, \lceil \log(n+1) \rceil)$ -重み関数, $(n, 2^n)$ -復号化関数, $(n, \lceil \log n \rceil)$ -符号化関数, 各個 n ビット2進数の加算をそれぞれ f_w, f_d, f_e, f_a とする。このとき、

$$\begin{aligned}
 A(C_{fw}) &= \Omega(\log n), \\
 A(C_{fd}) &= \Omega(n), \\
 A(C_{fe}) &= \Omega(\log n), \\
 A(C_{fa}) &= \Omega(\log k)
 \end{aligned}$$

がそれぞれ成立つ。

(証明) いずれの場合も、関数を実現する回路をその回路がもつ入力端子の個数 p により分けて証明する。ここでは、復号化関数の場合(復号器)について示すが、他の関数も全く同様に証明される。

復号器 C_{fd} は

$$\begin{aligned}
 (1) \quad p &\geq n-1 \\
 (2) \quad p &\leq n-2
 \end{aligned}$$

の場合に分かれる。

(1) $p \geq n-1$ の場合、仮定(A-3)より

$$A(C_{fd}) \geq n-1$$

となるので、 $A(C_{fd}) = \Omega(n)$ が成立つ。

(2) $p \leq n-2$ の場合、仮定(A-3)より

$$A(C_{fd}) \geq p \tag{5.4}$$

が成立つ。また、 $p \geq 1$ ゆえ、

$$n-1 \geq n-p \geq 2$$

となるので, C_{fd} に対する $R(t)$ の個数が条件

$$n-1 \geq l' \geq n-p-1 \quad (\geq 1) \quad (5.5)$$

を満たす時刻 t が存在する. 補題 5.1 より, $|S(x_i)| = l'$
($1 \leq l' \leq n-1$) なる x_i に対して,

$$\min_{|S(x_i)|=l'} I_{fd}(x_i) = l'$$

が成立つ. 定理 5.1 と (5.5) より

$$A(C_{fd}) \geq n-p-1 \quad (5.6)$$

となる. (5.5), (5.6) より

$$A(C_{fd}) \geq \frac{1}{2}(n-1)$$

となるので, $A(C_{fd}) = \Omega(n)$ となる.

従って, $(n, 2^n)$ -復号化関数に対して

$$A(C_{fd}) = \Omega(n)$$

が成立つ. ■

5.3. 関数に対する面積上界

ここでは、前節で示した関数のいずれもがその下界のオーダーで実現できることを示す。

[定理 5.3] $(n, \lceil \log(n+1) \rceil)$ -重み関数, $(n, 2^n)$ -復号化関数, $(n, \lceil \log n \rceil)$ -符号化関数, 長個の n ビット 2 進数の加算をそれぞれ f_w, f_d, f_e, f_a とすると, それぞれ

$$A(C_{f_w}) = O(\log n),$$

$$A(C_{f_d}) = O(n),$$

$$A(C_{f_e}) = O(\log n)$$

$$A(C_{f_a}) = O(\log k)$$

で実現できる。

(証明) 図 5.2 にそれぞれの関数を実現する回路を示す。ここで計数回路は入力 1 のとき 1 増し, 0 のとき内容が変化しないとする。一致回路は入力が一致するとき 1, そうでないとき 0 を出力するものとする。また, k ビット計数回路, k ビット一致回路, k ビットシフトレジスタはいずれも面積 $O(k)$ で実現できる。

(a) $(n, \lceil \log(n+1) \rceil)$ -重み関数

$\lceil \log(n+1) \rceil$ ビット計数回路を 0 にセットし, x_i ($0 \leq i \leq n-1$) を順次入力してゆき, 計数回路の結果を出力する。このとき, $A(C_{f_w}) = O(\log n)$ である。

(b) $(n, 2^n)$ -復号化関数

ここで構成する復号器は, 2^n 個の出力を添字の小さい

ものから順に出力するものである。 n ビット計数回路を 0 にセットし、0 から始めて 1 ずつ増しながら、 $2^n - 1$ までのついて入力との一致をとりその結果を順次出力する。ここで、計数回路の内容が i ($0 \leq i \leq 2^n - 1$) のときの一致回路の出力が z_i である。このとき、 $A(C_{fd}) = O(n)$ である。

(c) $(n, \lceil \log n \rceil)$ -符号化関数

$\lceil \log n \rceil$ ビット計数回路を 0 にセットし、 x_i ($0 \leq i \leq n-1$) を順次入力する。図 5.2 (c) のように制御素子を用いて回路を構成すれば、入力 x_i が 0 のあいだは計数回路に 1 が入力され、 x_i が 1 になれば、それ以降は 0 が入力される (S の動作を図 5.2 (c) に示す)。従って、計数回路の結果が符号器の出力である。このとき、S は定数面積で実現できるので、 $A(C_{fe}) = O(\log n)$ である。

(d) 長個の n ビット 2 進数の加算

長個の 2 進数を $x_{m-1i} \dots x_{0i}$ ($1 \leq i \leq k$) とする。 $(\lceil \log(k+1) \rceil + 1)$ ビット計数回路を 0 にセットする。1 ビット目長個 x_{0i} ($1 \leq i \leq k$) を順次計数回路に入力し、計数回路の結果をシフトレジスタへ移す。シフトレジスタの最下位ビットを y_0 として出力する。次にシフトレジスタを右へ 1 ビットシフトした結果を計数回路にセットし、2 ビット目長個 x_{1i} ($1 \leq i \leq k$) を計数回路に順次入力する。

これらの操作を $n-1$ ビット目まで繰り返すと、 y_0, \dots, y_{n-2} ままで得られる。最後に n ビット目長個 x_{n-1i} ($1 \leq i \leq k$) を計数回路へ順次入力し、シフトレジスタへ移し、その下位ビットから順に $y_{n-1}, \dots, y_{n+\lceil \log k \rceil - 1}$ とする。

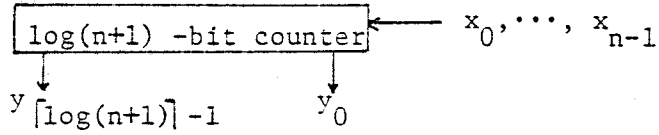
各ビットの和をとる操作において、 $(\lceil \log(k+1) \rceil + 1)$ ビットの計数回路で十分であることは次のようにして示される。1ビット目の和については明らかに十分である。 j ビット目 ($2 \leq j \leq n$) を入力するとき、計数回路の内容は $\sum_{i=1}^{j-1} k/2^i$ をこえない。また、 j ビット目の和は k をこえないので、 j ビット目を入力したとき、値の最大値を v_j とすると、

$$v_j \leq k + \sum_{i=1}^{j-1} k/2^i < 2k$$

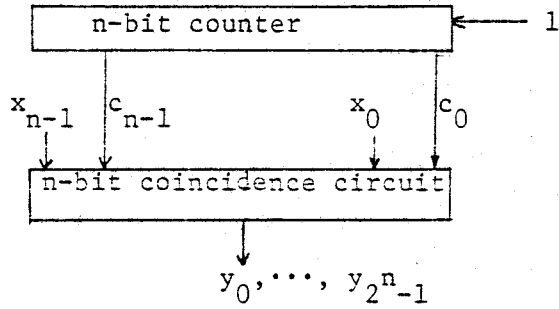
が成立つ。一方、 $(\lceil \log(k+1) \rceil + 1)$ ビットでは $2k+1$ までの2進数を表せる。よって、 $(\lceil \log(k+1) \rceil + 1)$ ビットの計数回路で十分である。

この回路について、各出力は (A-7) の仮定から固定されているのが n には依存しない。ところが、 k 個の入力を k 個ずつ区切って入力し、 k 個入力されるごとに計数回路の内容をシフトレジスタに移さなければならぬ。従って、入力を k 個ずつ入力するという制御が必要となる。これは $\lceil \log k \rceil$ ビットの計数回路で制御できるので、この制御を加えた全体の面積は $A(C_f) = O(\log k)$ である。 ■

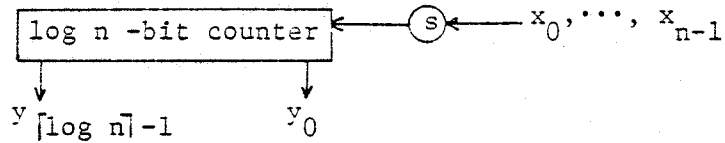
本章の面積下界の導出では、境界配置の仮定 (A-9) はおいていない。ところが、定理 5.3 の回路構成は、いづれも (A-9) を満足しているので、この仮定をおいても定理 5.2 の面積下界は改善できない。



(a) $(n, \lceil \log(n+1) \rceil)$ -weight function



(b) $(n, 2^n)$ -decoder



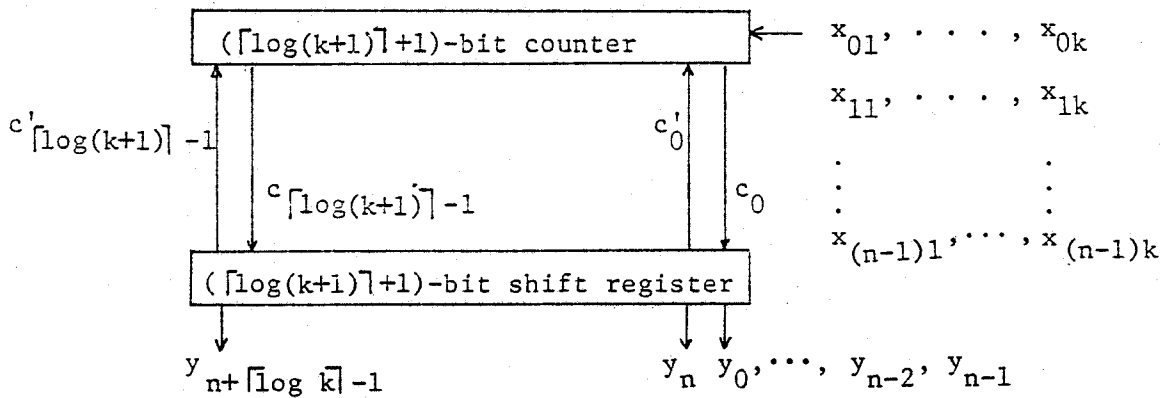
(c) $(n, \lceil \log n \rceil)$ -encoder

state	x	next state	y
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



initial state = 0

(c') State transition table of s



(d) Addition of k n -bit binary number

図5.2 各関数の回路構成.

Fig.5.2 Construction for each function.

6. 面積複雑度 (2) —— 組合せ回路の場合 ——

第3章～第5章では、順序回路による実現を許したとき、 $F(m,m)$ に属する関数を実現する回路の面積時間複雑度および面積複雑度を議論した。本章では実現を組合せ回路に限定したとき、必要となる回路面積を評価する手法について考察する。また、VLSIモデルにおける境界配置の仮定、凸平面領域の仮定の関係について議論する。

組合せ回路 C は記憶素子とループを含まない回路で、 $G(C)$ は非サイクル有向グラフになる。 C 中のPE（ポート、入出力端子）には次に定義する論理関数が結びつけられていると考える。

各入力端子 u_i には、入力変数 x_i と恒等論理関数 $f_{u_i}(x_i) = x_i$ が結びつけられる。入力線数 d で入力 u_1, \dots, u_d をもつゲート w は $f_w(u_1, \dots, u_d)$ が結びつけられる。 C は各出力端子に結びつけられた関数の集合を計算する。

また、以下では、 d 入力 u_1, \dots, u_d d' 出力 $v_1, \dots, v_{d'}$ をもつゲート w が $f_w(u_1, \dots, u_d)$ を実現しているとき $v_i = f_w(u_1, \dots, u_d)$ ($1 \leq i \leq d'$)と仮定する。この仮定のもとでは、仮定(A-8)の d 、 d' は定数でなくともよい。

6.1. 入出力位置の指定と回路面積の関係

ここで考える組合せ回路は(A-1), (A-3), (A-4), (A-6), (A-7), (A-9)を満たすものとする。すなわち本節では回路が埋め込まれる領域は凸領域でなくともよく、一般の閉領域でよい。

V を節点集合、 E を有向辺集合としてもつ有向グラフ

を $G = (V, E)$ とする. また, 有向辺は節点の順序対 $\langle u, v \rangle$ ($u, v \in V$) で表し, 有向道

$$\langle v_1, v_2 \rangle \langle v_2, v_3 \rangle \dots \langle v_{k-1}, v_k \rangle \quad (\langle v_i, v_{i+1} \rangle \in E, \quad 1 \leq i \leq k-1)$$

を $\langle v_1, \dots, v_k \rangle$ と表す.

G の節点で, 入次数が 0 の節点を始点, 出次数が 0 の節点を終点という. また, 始点または終点を端点という.

[定義 6.1] $G = (V, E)$ を有向グラフとする. G の 2 つの有向道 $p = \langle v_1, \dots, v_k \rangle$, $q = \langle u_1, \dots, u_\ell \rangle$ に対して,

$$\{v_1, \dots, v_k\} \cap \{u_1, \dots, u_\ell\} = \emptyset,$$

すなわち, 共通の節点を持たないとき, p と q は節点独立 (node-disjoint) であるという. G の有向道の集合 P に対して, P のどの 2 つの有向道も節点独立ならば, P は節点独立であるという.

G の節点集合 V の互いに素な部分集合を V_1, V_2 とする. G の有向道 $p = \langle v_1, \dots, v_k \rangle$ が次の条件を満たすとき, p は V_1 と V_2 を結ぶ道であるという.

(1) ($v_1 \in V_1$ か) $v_k \in V_2$) または ($v_1 \in V_2$ か) $v_k \in V_1$).

(2) 各 i ($2 \leq i \leq k-1$) に対して, $v_i \in V - V_1 - V_2$. ■

次の 2 つの補題は, 回路の境界上におかれる入出力端子の位置の指定とその回路の面積の関係を与える.

回路が埋め込まれる閉領域を R とする. R の境界 B は

閉曲線になるが、 B の連続した部分閉曲線を B の部分境界という。

[補題 6.1] 回路 C の回路グラフを $G(C)$ とし、 $G(C)$ の入出力節点の集合を IO とする。次の (1) ~ (5) の条件を満たす IO の部分集合 V_1, V_2, V_3 が存在するならば、次式が成立つ。

$$A = \Omega(|V_3|^2).$$

(1) $V_i \cap V_j = \emptyset \quad (1 \leq i < j \leq 3).$

(2) $G(C)$ は V_1 と V_3 を結ぶ節点独立な道の集合 P_1 を含む。

(3) $G(C)$ は V_2 と V_3 を結ぶ節点独立な道の集合 P_2 を含む。

(4) $|P_1| = |P_2| = |V_3|.$

(5) C が埋め込まれる閉領域 R の境界上に、互いに素な部分境界 B_1, B_3 が存在し、 $IO_i \ (i=1, 3)$ を B_i 上におかれる入出力節点の集合とすると、

(i) $V_1 \subseteq IO_1$ かつ $(V_2 \cup V_3) \cap IO_1 = \emptyset,$

(ii) $V_3 \subseteq IO_3$ かつ $(V_1 \cup V_2) \cap IO_3 = \emptyset$

を満たす。

(証明) $l = |V_3|$ とし、部分境界 B_3 上におかれる V_3 の入出力節点を、 B_3 におかれた順に v_1, \dots, v_l とする。各 $v_i \ (1 \leq i \leq l)$ に対して、 v_i を端点とする P_1 および P_2 の有向道をそれぞれ p_i, q_i とする。 q_i の v_i でない端点は B_1 上にも B_3 上にもおかれるないので、 q_i は R 上で少なくとも 1 回各 $p_j \ (j < i)$ と交差するが、または少なくとも 1 回

各 $p_h (h > i)$ と交差する。ここで、有向道 p と q が交差するというのは、有向道のある 2 つの有向辺が交差する場合と、有向辺がある節点に合流分岐する場合の 2 つを意味するものとする (図 6.1)。また、 p と q が交差する場所を交差点という。従って、各 p_i は R 上で P_i の有向道と少なくとも $\min(i-1, k-i)$ 回交差する。

仮定 (A-4) より単位面積は高々 ν 本の導線が交差できるので、単位面積は高々 $\binom{\nu}{2}$ 個の交差点をもつ。ゆえに

$$\begin{aligned} A &\geq \frac{1}{\binom{\nu}{2}} \sum_{i=1}^k \min(i-1, k-i) \\ &= \begin{cases} \frac{1}{2\nu(\nu-1)} k(k-2) & (\text{ } k \text{ が偶数の場合}) \\ \frac{1}{2\nu(\nu-1)} (k-1)^2 & (\text{ } k \text{ が奇数の場合}) \end{cases} \\ &= \Omega(|V_3|^2) \end{aligned}$$

となり、補題が成立つ。■

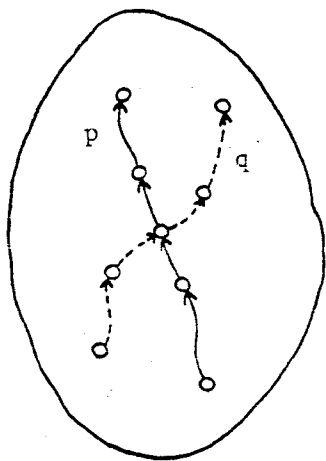
次の補題は、補題 6.1 と同様にして示される。

[補題 6.2] 回路 C の回路グラフを $G(C)$ とし、 IO を $G(C)$ の入出力節点の集合とする。次の (1) ~ (4) の条件を満たす IO の部分集合 $V_1 \sim V_4$ が存在するとする。

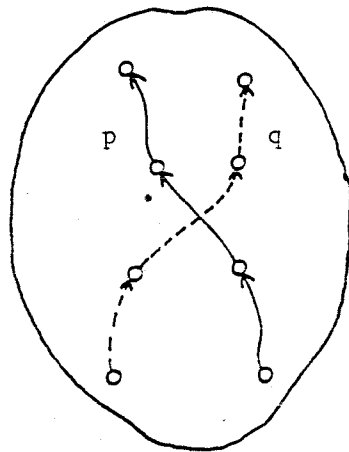
(1) $V_i \cap V_j = \emptyset (1 \leq i < j \leq 4)$.

(2) $G(C)$ は V_1 と V_2 を結ぶ節点独立な道の集合 P_{12} を含む。

(3) $G(C)$ は V_2 と V_4 を結ぶ節点独立な道の集合 P_{24} を含む。



(a) crossing with a common vertex



(b) multi-level crossing

図6.1 有向道 p と q の交差.

Fig.6.1 Two kind of crossing.

(4) C の境界上で、互いに素な部分境界 $B_1 \sim B_4$ が時計回りの順に存在し、 IO_i ($i=1 \sim 4$) を B_i 上におかれる入出力節点の集合とすると、

$$V_i \subseteq IO_i \quad (i=1 \sim 4)$$

を満たす。

このとき、次式が成立つ。

$$A = \Omega(|P_\beta| \cdot |P_\alpha|). \quad \blacksquare$$

補題 6.1, 6.2 は、回路グラフがある性質を満足したとき、回路面積がどのようなようになるかについて述べている。これらの補題を用いて、関数を実現する組合せ回路の面積下界を導出するために、関数を実現する回路に対する回路グラフの性質について考える。

[定義 6.2] $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_m)$ とし、 (n, m) -関数を $Y = f(X)$ とする。 $X_1 = (x_{i_1}, \dots, x_{i_h})$ を X の部分系列とし、 $X - X_1 = (x_{j_1}, \dots, x_{j_{n-h}})$ とする。また、 Y_1 を Y の部分系列とし、 $Q = (q_1, \dots, q_{n-h}) \in \{0, 1\}^{n-h}$ とする。

f の入力部分系列 $X - X_1$ の各入力 x_{j_r} ($1 \leq r \leq n-h$) に q_r を代入し、出力を Y_1 に制限して得られる関数 $Y_1 = f'(X_1)$ を f の部分関数といい、 $Y_1 \uparrow f(X, Q) \uparrow X_1$ と表す。

$(1, \dots, k)$ のある置換を $(p(1), \dots, p(k))$ とする。 (k, k) -関数 $(y_1, \dots, y_k) = g(x_1, \dots, x_k)$ に対して、

$$y_i = x_{p(i)} \quad (1 \leq i \leq k)$$

が成立つとき、 g を k -恒等関数という。 \blacksquare

[命題 6.1] (Menger の定理)⁽¹⁴⁾

$G = (V, E)$ を有向グラフとし, $a (\in V)$ を G の始点,
 $b (\in V)$ を G の終点とする.

U を次の条件を満たす $V - \{a, b\}$ の部分集合とする.

(1) a から b への任意の有向道は, U の節点を通る.

(2) U は極小集合である. すなわち, $U' \subsetneq U$ なる集合 U' は (1) を満たさない.

このとき, $\{a\}$ と $\{b\}$ を結ぶ節点独立な道[†] の最大個数は, U の節点数に等しい. ■

この命題を用いると, 閾値を実現する組合せ回路に対する回路グラフの性質が次の補題によって示される.

閾値 $\langle y_1, \dots, y_m \rangle = f(x_1, \dots, x_n)$ に対して, f の入力値の個数 $z(f)$, f の出力値の個数 $\alpha(f)$ を以下のように定義する.

$$z(f) = \{ (a_1, \dots, a_n) \mid (a_1, \dots, a_n) \in \{0, 1\}^n, \\ f(a_1, \dots, a_n) \text{ が定義されている} \},$$

$$\alpha(f) = \{ f(a_1, \dots, a_n) \mid (a_1, \dots, a_n) \in \{0, 1\}^n, \\ f(a_1, \dots, a_n) \text{ が定義されている} \}.$$

また, f の入力系列 (x_1, \dots, x_n) の部分系列に対する入力値の個数, f の出力系列 $\langle y_1, \dots, y_m \rangle$ の部分系列に対する出力値の個数も同様に定義する.

[†] この場合, a, b 以外に互いに共通節点をもたない有向道を意味する.

関数 f を組合せ回路で実現するとき、 C_f の入出力端子と f の入出力変数は 1対1 に対応する。従って、 IO を回路グラフ $G(C_f)$ の入出力節点の集合とするとき、全単射写像

$$\pi_{f, G(C_f)} : \{x_1, \dots, x_n\} \cup \{y_1, \dots, y_m\} \rightarrow IO$$

によって、各入出力変数がどの入出力節点に対応するかを表すものとする。 $W \subseteq \{x_1, \dots, x_n\} \cup \{y_1, \dots, y_m\}$ のとき、

$$\pi_{f, G(C_f)}(W) = \{ \pi_{f, G(C_f)}(w) \mid w \in W \}$$

とする。以下では、 f と $G(C_f)$ が明らかの場合、 $\pi_{f, G(C_f)}$ を π と略記する。

[補題 6.3] $(y_1, \dots, y_m) = f(x_1, \dots, x_n) \in F(n, m)$ に属する関数とし、 $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_m)$ とする。このとき、 $G(C_f)$ は $\pi(S(X))$ と $\pi(S(Y))$ を結ぶ節点独立な道を少なくとも $\lceil \log_2 o(f) \rceil$ 本含む。

(証明) 回路グラフ $G(C_f) = (V, E)$ から、次のような有向グラフ $\hat{G}(C_f)$ を構成する。

$$\hat{G}(C_f) = (V \cup \{a, b\}, E \cup \{ \langle a, p \rangle \mid p \in \pi(S(X)) \} \cup \{ \langle b, q \rangle \mid q \in \pi(S(Y)) \})$$

ここで、 $a, b \notin V$ である。

$\hat{G}(C_f)$ において、 $U \subseteq V$ の部分集合で、 a から b への任意の有向道は U の節点を通るとする。

今、

$$|U| < \lceil \log \Theta(f) \rceil$$

とすると, $U = \{u_1, \dots, u_j\}$ ($j \leq \lceil \log \Theta(f) \rceil - 1$) とおける. C_f は組合せ回路であるから, 入力に対して u_1, \dots, u_j に対応する C_f の PE の出力値は一意に定まり, f の出力はそれらの値によって一意に定まる. 各 PE の出力値は同じであるから, u_1, \dots, u_j に対応する PE の出力値のとり得る個数は高々 2^j であり

$$2^j \leq 2^{\lceil \log \Theta(f) \rceil - 1} < \Theta(f)$$

となる. 従って, C_f は正しく f を実現できないので

$$|U| \geq \lceil \log \Theta(f) \rceil$$

が成立つ. この結果と命題 6.1 および $\hat{G}(C_f)$ の作り方から, $\hat{G}(C_f)$ は $\pi(S(X))$ と $\pi(S(Y))$ を結ぶ節点独立な道を少なくとも $\lceil \log \Theta(f) \rceil$ 本含む. ■

補題 6.1, 6.3 より, 次の定理を得る.

[定理 6.1] $Y = f(X)$ を関数とし, 次の条件 (1) ~ (3) を満たす X の部分系列 X_1, X_2 , および Y の部分系列 Y_1 が存在するとする.

(1) $S(X_1) \cap S(X_2) = \emptyset$.

(2) $Y_1 \upharpoonright f(X, Q_1) \upharpoonright X_1, Y_1 \upharpoonright f(X, Q_2) \upharpoonright X_2$ が $|S(Y_1)|$ -恒等関数となるような $X - X_1, X - X_2$ への代入 Q_1, Q_2 がそれぞれ存在する.

(3) C_f の境界の互いに素な部分境界 B_1, B_2 が存在し、
 $B_i (i=1, 2)$ 上におかれた入出力節点の集合を IO_i
 とすると、(i), (ii) を満たす。

$$(i) \pi(S(X_1)) \subseteq IO_1 \text{ かつ } (\pi(S(X_2)) \cup \pi(S(Y_1))) \cap IO_1 = \emptyset.$$

$$(ii) \pi(S(Y_1)) \subseteq IO_2 \text{ かつ } (\pi(S(X_1)) \cup \pi(S(X_2))) \cap IO_2 = \emptyset.$$

このとき、

$$A = \Omega(|S(Y_1)|^2)$$

が成立つ。

(証明) k -恒等関数 g は $F(k, k)$ に属する関数であり、
 $\alpha(g) = 2^k$ である。従って、条件(2)と補題6.3より、 $G(C_f)$
 は、 $\pi(S(X_1))$ と $\pi(S(Y_1))$ を結ぶ節点独立な道を $|S(Y_1)|$ 本、
 および、 $\pi(S(X_2))$ と $\pi(S(Y_1))$ を結ぶ節点独立な道を $|S(Y_1)|$
 本含む。また、条件(1)より

$$\pi(S(X_1)) \cap \pi(S(X_2)) = \emptyset$$

が成立つので、補題6.1の条件(1)~(4)は満たされる。補
 題6.1の条件(5)は、条件(3)により満たされるので、

$$A = \Omega(|S(Y_1)|^2)$$

が成立つ。 ■

定理6.1において、入出力変数の関係を入れかえると
 次の定理が得られる。

[定理6.2] $Y = f(X)$ を関数とし、次の条件(1)~(3)
 を満たす X の部分系列 X_1 、および Y の部分系列 Y_1, Y_2 が

$$(1) S(Y_1) \cap S(Y_2) = \emptyset.$$

(2) $Y_1 f(x, Q_1) \uparrow X_1$, $Y_2 f(x, Q_2) \uparrow X_1$ が $|S(X_1)|$ -恒等関数となるような, $X \rightarrow X_1$ の代入 Q_1, Q_2 がそれぞれ存在する.

(3) C_f の境界の互いに素な部分境界 B_1, B_2 が存在し, $B_i (i=1, 2)$ 上におかれた入出力節点の集合を IO_i とすると, (i), (ii) を満たす.

$$(i) \pi(S(Y_1)) \subseteq IO_1 \text{ かつ } (\pi(S(X_1)) \cup \pi(S(Y_2))) \cap IO_1 = \emptyset.$$

$$(ii) \pi(S(Y_2)) \subseteq IO_2 \text{ かつ } (\pi(S(X_1)) \cup \pi(S(Y_1))) \cap IO_2 = \emptyset.$$

このとき,

$$A = \Omega(|S(X_1)|^2)$$

が成立つ. ■

定理 6.1 により, 加算, 乗算, 最大値関数, 最小値関数などを實現する組合せ回路は, 各オペラントに対応する入力端子, および演算結果に対応する出力端子をそれぞれ回路の境界上に分離しておくならば, 入力のビット数に比例する回路面積を必要とする (系 6.1).

加算などの 2 項演算 β は,

$$(y_1, \dots, y_m) = \beta(x_1, \dots, x_n, w_1, \dots, w_k)$$

として表す. ここで, $x_n \dots x_1, w_k \dots w_1$ はそれぞれのオペラントの 2 進数表示, $y_m \dots y_1$ は演算結果の 2 進数表示とする.

2 項演算 β に対して, $n = k$, $m \geq n$ とする. n ビット 2 進数 $a_n \dots a_1$ が, 任意の n ビット 2 進数

$a_n \cdots a_1$ に対して,

$$(a_1, \dots, a_n, 0, \dots, 0) = \beta(\lambda_1, \dots, \lambda_n, a_1, \dots, a_n),$$

$$(a_1, \dots, a_n, 0, \dots, 0) = \beta(a_1, \dots, a_n, \lambda_1, \dots, \lambda_n)$$

が成立つとき, $\lambda_n \cdots \lambda_1$ は β の単位元という.

[系 6.1] $(y_1, \dots, y_m) = \beta(x_1, \dots, x_n, w_1, \dots, w_n)$ を単位元をもつ 2 項演算とする. C_β の境界 E に, 互いに素な部分境界 B_x, B_w, B_y が存在して, $\pi(\{x_1, \dots, x_n\})$ は B_x 上に, $\pi(\{w_1, \dots, w_n\})$ は B_w 上に, $\pi(\{y_1, \dots, y_m\})$ は B_y 上にそれぞれおかれるとする. このとき,

$$A = \Omega(n^2)$$

が成立つ. ■

6.2. 乗算, 除算, リーティングの面積下界

本節では, 前節の結果を用いて, 乗算, 除算, リーティングを組合せ回路で実現するとき, それらの回路の面積下界を議論する.

系 6.1 によれば, 2 つの n ビット 2 進数の加算を実現する組合せ回路は, 加数と被加数に対応する入力端子, および和に対応する出力端子を回路の境界 E に分離しておけば, $\Omega(n^2)$ の面積を必要とする. しかしながら, 加算の場合, 全加算器を用いた通常の構成で, 加数と被加数に対応する入力端子を交互におくことにより, $O(n)$ の面積ができる. 以下では, 乗算, 除算, リーティングに対しては, 入出力端子を回路の境界 E にどのようにおい

とも、 $O(n^2)$ より小さい面積では実現できなりのことを示す。すなわち、これらの関数を実現する組合せ回路は、境界配置の仮定を満たす限り、 $\Omega(n^2)$ の面積を必要とする。

6.2.1. 乗算, 除算

乗算, 除算を実現する組合せ回路の面積下界を求めるために、次のようなシフト関数を考える。

$$(y_1, \dots, y_N) = f_n(x_1, \dots, x_N, \alpha_0, \dots, \alpha_{N-1})$$

(1) $\alpha_0, \dots, \alpha_{N-1}$ のうちちょうど $1 > \alpha_i \geq 1$ になる。

(2) $\alpha_i = 1$ のときかつそのときに限り

$$y_j = \begin{cases} x_{j-i} & (i \leq j \leq N) \\ \text{定義せず} & (j < i). \end{cases}$$

乗算, 除算はシフト関数 f_n を部分関数として含むので、乗算, 除算を実現する組合せ回路の1つの面積下界は、 f_n を実現する組合せ回路の面積下界を求めることにより得られる。まず、 f_n の性質について考える。

[定義 6.3] \mathbb{Z} を整数の集合とし、

$$[k, k'] = \{i \mid i \in \mathbb{Z}, k \leq i \leq k'\}$$

とする。整数 r と非負整数 a, b ($a \geq b$) に対して,

$$L_r(a, b) = \{ J \mid 2^{[r+1, r+a]} \text{ か } |J| = b \}$$

と定義する。ここで, $2^{[r+1, r+a]}$ は $[r+1, r+a]$ の部分集合全体の集合を表す。

\mathbb{Z} の部分集合 $K = \{ l_1, \dots, l_b \}$ に対して, K の $i \in \mathbb{Z}$ に対して $sh(K, i)$ を次のように定義する。

$$sh(K, i) = \{ l_1 + i, \dots, l_b + i \}.$$

\mathbb{Z} の2つの部分集合 K_1, K_2 に対して, $|K_1 \cap K_2|$ を $m(K_1, K_2)$ で表し, K_1 と K_2 の一致数という。 ■

[補題 6.4] $K_1, K_2 \in L_r(a, b)$ とするとき,
 $m(K_1, sh(K_2, i)) \geq \lfloor b^2/2a \rfloor$
となる i ($-a \leq i \leq a$) が存在する。

(証明) $L_r(a, b)$ の定義より, K_1 の要素 p , K_2 の要素 q に対して,

$$r+1 \leq p, \quad q \leq r+a$$

が成立つ。従って,

$$-(a-1) \leq p - q \leq a-1$$

となるので, K_1 と K_2 の要素の各対 (p, q) に対して,

$p = q + i$ となる整数 i ($-(a-1) \leq i \leq a-1$) がち
ように一つ存在する。このことから,

$$\begin{aligned}
\sum_{i=-(a-1)}^{a-1} m(k_1, sh(k_2, i)) &= \sum_{i=-(a-1)}^{a-1} |k_1 \cap sh(k_2, i)| \\
&= \sum_{i=-(a-1)}^{a-1} \sum_{p \in k_1} \sum_{q \in k_2} |\{p\} \cap \{q+i\}| \\
&= \sum_{p \in k_1} \sum_{q \in k_2} \sum_{i=-(a-1)}^{a-1} |\{p\} \cap \{q+i\}| \\
&= \sum_{p \in k_1} \sum_{q \in k_2} 1 \\
&= k^2
\end{aligned}$$

が成立つ。従って、 k_1 と $sh(k_2, i)$ の一致数が $\lfloor k^2/2a \rfloor$ 以上となる i ($-a \leq i \leq a$) が少なくとも 1 つ存在する。 ■

[補題 6.5] $(y_1, \dots, y_{2N}) = f_a(x_1, \dots, x_{2N}, a_0, \dots, a_{2N-1})$ を $2N$ ビットシフト関数とする。 X, Y をそれぞれ、 $(x_1, \dots, x_N), (y_{N+1}, \dots, y_{2N})$ の任意の部分系列とし、 $|S(X)| = |S(Y)| = k$ とする。このとき、 f_a は次の条件を満たす k -恒等関数を部分関数として含む。

$$(1) \quad S(X_1) \subseteq S(X) \quad \text{かつ} \quad S(Y_1) \subseteq S(Y).$$

$$(2) \quad k \geq \lfloor k^2/2N \rfloor.$$

(証明) $K_1 = \{h+N \mid x_h \in S(X)\}$, $K_2 = \{j \mid y_j \in S(Y)\}$ とおくと、

$$K_1, K_2 \in L_N(N, k)$$

である。補題 6.4 により、

$$m(k_1, sh(k_2, i)) \geq \lfloor k^2/2N \rfloor$$

となる i ($-N \leq i \leq N$) が存在する。

$$S(X_1) = \{ x_{h-N} \mid h \in k_1 \cap sh(k_2, i) \},$$

$$S(Y_1) = \{ y_j \mid j \in k_1 \cap sh(k_2, i) \}$$

とおけば、 $m(k_1, sh(k_2, i))$ -恒等関数を得られる。 ■

各個の入出力節点の集合を IO とし、回路の境界（または部分境界） B 上に IO の節点がおかれているとする。 B を l 個の互いに素な部分境界 B_1, \dots, B_l に分割するとき、いずれの部分境界上にも IO の節点が少なくとも $\lfloor k/l \rfloor$ 個おかれているならば、この分割を IO に対する B の l 分割という。

補題 6.2, 6.5 より、シフト関数 f_0 を実現する組合せ回路の面積下界が次の定理により示される。

[定理 6.3] $(y_1, \dots, y_{3N}) = f_0(x_1, \dots, x_{3N}, a_0, \dots, a_{3N-1})$ を $3N$ ビットシフト関数とする。 C を f_0 を部分関数として含む関数を実現する組合せ回路とするとき

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

が成立つ。

(証明) $G(C)$ において、 $I = \pi(\{x_1, \dots, x_N\})$, $O = \pi(\{y_{N+1}, \dots, y_{2N}\})$ とする。 B を C の境界とし、 I に対して B を 2 分割したときの部分境界を B_1, B_2 とする。 $N = 4t + \delta$ ($0 \leq \delta \leq 3$) とおくと、 O の少なくとも

$2t$ は B_1 , または B_2 上におかれている. 一般性を失うことなく, B_2 上には 0 の出力節点が $2t$ 以上おかれていると仮定してよい.

B_1 上におかれた 1 の入力節点に対して B_1 を 2 分割したときの部分境界を D_1, D_2 とし, B_2 上におかれた 0 の出力節点に対して B_2 を 2 分割したときの部分境界を E_1, E_2 とする. このとき, D_1, D_2, E_1, E_2 が境界 B 上で時計回りの順であるとして一般性を失わない.

D_1, D_2 上におかれる 1 の入力節点の集合をそれぞれ I_1, I_2 とする. 同様に, E_1, E_2 上におかれる 0 の出力節点の集合をそれぞれ O_1, O_2 とする. ここで,

$$|I_1|, |I_2|, |O_1|, |O_2| \geq t$$

である. 補題 6.3, 6.5 によって, $G(C)$ は I_1 と O_1 を結ぶ節点独立な道を $\ell_1 (\geq Lt^2/2N)$ 本以上, および I_2 と O_2 を結ぶ節点独立な道を $\ell_2 (\geq Lt^2/2N)$ 本以上含む.

$t = \lfloor N/4 \rfloor$ であるから, 補題 6.2 より

$$\begin{aligned} A &= \Omega(\ell_1 \cdot \ell_2) \\ &= \Omega\left(L \frac{1}{2N} \lfloor N/4 \rfloor^2\right)^2 \\ &= \Omega(N^2) \end{aligned}$$

が成立つ. ■

2^n の m ビット 2 進数の乗算, 2^n ビット 2 進数を m ビット 2 進数で割る乗算は n ビットシフト関数を部分関数として含むので, 次の 2 つの系が成立つ.

[系 6.2] 2つの n ビット 2 進数乗算を実現する組合せ回路に対して,

$$A = \Omega(n^2)$$

が成立つ。 ■

[系 6.3] $2n$ ビット 2 進数を n ビット 2 進数で割る除算を実現する組合せ回路に対して,

$$A = \Omega(n^2)$$

が成立つ。 ■

乗算, 除算はともに組合せ回路によって, $O(n^2)$ の面積で実現できる⁽⁵⁾。従って, ここで示した下界は定数の範囲を除いて最良のものである。

Lipton, Tarjan は, 平面グラフの分割定理 (Planar Separator Theorem) を用いて, n ビット乗算などを実現する組合せ回路の面積下界が $\Omega(n^2)$ となることを示している⁽¹¹⁾。系 6.2, 6.3 によって, 本論文の手法を用いても同様の面積下界が導出できることがわかる。

6.2.2. リーティング

[定義 6.4]⁽²²⁾ 関数 $(y_1, \dots, y_N) = f(x_1, \dots, x_N, a_1, \dots, a_e)$ が置換群 G を計算するというのは, G の各要素 g に対して, a_1, \dots, a_e の値を定めることにより

$$y_i = x_{g(i)} \quad (1 \leq i \leq N)$$

となることである。ここで, $(g(1), \dots, g(N))$ は g によって定まる $(1, \dots, N)$ の置換を表す。 ■

n 個の長ビット2進数 ($n \geq \lceil \log n \rceil$) をソートする関数は $\lfloor n/2 \rfloor$ 次の対称群 $S_{\lfloor n/2 \rfloor}$ を計算する関数を部分関数として含んでいる⁽⁴⁾. ここでは, S_N を計算する関数を実現する組合せ回路の面積下界を導出する. リーティングに対する面積下界はその結果から得られる.

[定理 6.4] $(y_1, \dots, y_N) = f(x_1, \dots, x_N, a_1, \dots, a_k)$ を N 次の対称群 S_N を計算する関数とする. このとき,

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

が成立つ.

(証明) $G(C_f)$ において, $I = \pi(\{x_1, \dots, x_N\})$, $O = \pi(\{y_1, \dots, y_N\})$ とする. B を C_f の境界とする. O に対して B を3分割したときの部分境界を B_1, B_2, B_3 とする. このとき, B_1 上に少なくとも $\lfloor N/3 \rfloor$ の I の入力節点がおかれているとして一般性を失わぬ.

$n = \lfloor N/3 \rfloor$ とする. B_1 上におかれている n 個の I の入力節点の集合を I_1 とし, それらに対応する入力変数の系列を

$$X_1 = (x_{i_1}, \dots, x_{i_n})$$

とする. また, B_1 および B_2 上におかれる n 個の O の出力節点の集合をそれぞれ O_1, O_2 とし, それらに対応する出力変数の系列をそれぞれ,

$$Y_1 = (y_{j_1}, \dots, y_{j_n}),$$

$$Y_2 = (y_{h_1}, \dots, y_{h_n})$$

とする. ここで, $I_1 \subseteq I$, $O_1, O_2 \subseteq O$, $O_1 \cap O_2 = \emptyset$ である.

関数 f は対称群 S_N を計算するので、

$$y_{jp} = X_{g_1(i_p)} \quad (1 \leq p \leq k),$$
$$y_{hg} = X_{g_2(i_g)} \quad (1 \leq g \leq k)$$

となる置換 $g_1, g_2 (\in S_N)$ が存在する。このとき、定理 6.2 の条件はすべて満たされ、 $|S(x_i)| = \lfloor N/3 \rfloor$ であるので、

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

が成立つ。 ■

[系 6.4] n 個の k ビット 2 進数 ($k \geq \lceil \log_2 n \rceil$) をソートする関数に対して、

$$A = \Omega(n^2)$$

が成立つ。 ■

n 個の $\lceil \log_2 n \rceil$ ビット 2 進数のソートは、組合せ回路によって、面積 $O(n^2 \log n)$ で実現できる⁹⁾。系 6.4 で得られた面積下界は $\Omega(n^2)$ であるので、 $\log n$ のギャップが残される。しかしながら、次節で議論するように、回路を凸領域に埋め込むと仮定すると、面積下界が $\Omega(n^2 \log n)$ (定理 6.6) になることが示される。従って、 $O(n^2 \log n)$ の実現が定数の範囲を除いて改善の余地がないことになる。

6.3. 回路を凸領域に埋め込む場合の面積下界

本節では、回路が埋め込まれる領域を凸領域と仮定する (A-2) とし、シフト関数、リーディング、復号化関数、符号化関数などを実現する組合せ回路の面積下界について考察する。

前節で得られた面積下界は、実際に回路中の PE と導線の占める部分を評価している。従って、回路が埋め込まれる領域は凸領域でなくても、境界配置の仮定を之を満たしていれば成立つ。ここでは、まずシフト関数 (乗算、除算)、リーディングを実現する組合せ回路を凸領域に埋め込んだ場合、境界配置の仮定をとり除いても、定理 6.3, 6.4 と同じ面積下界が成立つことを示す。また、リーディングに対しては、凸領域と境界配置の仮定を共に要請したとき、系 6.4 で示した面積下界が真に上がることを示す。

次に、境界配置の仮定のもとで、復号化関数、符号化関数を実現する組合せ回路を凸領域に埋め込んだときの面積下界を示す。また、ここで得られた下界は最良のものであることを示す。

以下では、ことわりがない限り、回路は凸領域に埋め込まれるとする。

6.3.1. シフト関数、リーディング

[定理 6.5] $(y_1, \dots, y_{3N}) = f_0(x_1, \dots, x_{3N}, a_0, \dots, a_{3N-1})$ を $3N$ ビットシフト関数とする。これを f_0 を部分関数として含む関数を実現する組合せ回路とすると、境界配置の仮定なしに、次式が成立つ。

$$A = \Omega(n^2).$$

(証明) C が埋め込まれる凸領域を R とする. $G(C)$ において, $I = \pi(\{x_1, \dots, x_N\})$, $O = \pi(\{y_{N+1}, \dots, y_{2N}\})$ とする. R の直径に垂直な弦によって, R は2つの領域 R_1, R_2 に分けられる. このとき, R_1 におかれる I の入力節点数を i 個とすると, $N-i$ 個の I の入力節点が R_2 におかれる. R の直径に垂直な弦を直径に沿って動かすことにより, R_1, R_2 ともに少なくとも $\lfloor N/2 \rfloor$ の I の入力節点を含むようにできる. そのときの弦の長さを L とする.

O の出力節点の少なくとも $\lfloor N/2 \rfloor$ は R_1 または R_2 におかれるので, R_2 に少なくとも $\lfloor N/2 \rfloor$ の O の出力節点がおかれていますと仮定してよい. R_1 におかれたちょうど $\lfloor N/2 \rfloor$ の I の入力節点の集合を I_1 とし, R_2 におかれたちょうど $\lfloor N/2 \rfloor$ の O の出力節点の集合を O_2 とする. 補題 6.3, 6.5 より, $G(C)$ は I_1 と O_2 を結ぶ節点独立な道を少なくとも $\lfloor \lfloor N/2 \rfloor^2 / 2N \rfloor$ 本含む. 従って, L の弦を $\lfloor \lfloor N/2 \rfloor^2 / 2N \rfloor$ 本以上の節点独立な道が横切る. このとき, L の弦上には, $\lfloor \lfloor N/2 \rfloor^2 / 2N \rfloor$ 個以上の点がおかれるので,

$$L \geq \frac{1}{\lfloor \frac{N}{2} \rfloor} \lfloor \frac{\lfloor N/2 \rfloor^2}{2N} \rfloor$$

が成立つ. 命題 4.1 (4.2) より,

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

が成立つ. ■

[定理 6.6] n 個の長さビット 2 進数 (長さ $\geq \lfloor \log n \rfloor$) のリーディングに対して, 境界配置の仮定なしに

$$A = \Omega(n^2)$$

が成立つ. さらに, 境界配置の仮定をおくと,

$A = \Omega(n^2 \log n)$
が成立つ。

(証明) リーディングを実現する回路が埋め込まれる凸領域を R とする。境界配置の仮定にかかわらず、定理 6.5 の証明と同様に R を分割することにより、 R の直径に垂直な弦の長さを L とすると、

$$L = \Omega(n)$$

が成立つ。従って、境界配置の仮定をおかない場合は、命題 4.1 (4.2) により、

$$A = \Omega(n^2)$$

が成立つ。

境界配置の仮定をおいた場合、 R の境界 Γ に $2nk$ ($\geq 2n \lceil \log n \rceil$) の入出力節点がおかれるので、 R の周長を P とすると、

$$P = \Omega(n \log n)$$

となる。従って、命題 4.1 (4.1) より

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

が成立つ。 ■

6.3.2. 復号化関数, 符号化関数

ここでは, 前節の結果の応用として, 復号化関数, 符号化関数を実現する組合せ回路の面積下界を示す.

$(n, 2^n)$ -復号化関数の定義より, $x_i = x_i^1$, $\bar{x}_i = x_i^0$ と表すと, 各出力 y_j は次のように表される.

$$y_j = x_{n-1}^{a_{n-1}(j)} \cdots x_0^{a_0(j)} \quad (0 \leq j \leq 2^n - 1)$$

ここで, $a_{n-1}(j) \cdots a_0(j)$ は j の 2 進数表現である.

$X = (x_0, \dots, x_{n-1})$ とする. X の部分系列を $X_1 = (x_{i_1}, \dots, x_{i_k})$ とし, $X - X_1 = (x_{h_1}, \dots, x_{h_{n-k}})$ とする.

X_1 と $n-k$ 組 $(a_{h_1}, \dots, a_{h_{n-k}})$ ($\in \{0, 1\}^{n-k}$) に対して, 次のような出力変数の集合を定義する.

$$\gamma(a_{h_1}, \dots, a_{h_{n-k}})$$

$$= \{ y_j = x_{n-1}^{a_{n-1}(j)} \cdots x_0^{a_0(j)} \mid a_{h_1}(j) = a_{h_1}, \dots, a_{h_{n-k}}(j) = a_{h_{n-k}} \}$$

すなわち, $\gamma(a_{h_1}, \dots, a_{h_{n-k}})$ は, y_j の右辺に現れる (X_1 に属さない) すべての $x_{h_p}^{a_{h_p}(j)}$ の $a_{h_p}(j)$ が a_{h_p} に等しいような出力変数の集合である. 各 y_j はちょうど 1 つの $\gamma(a_{h_1}, \dots, a_{h_{n-k}})$ に属するので $\gamma(a_{h_1}, \dots, a_{h_{n-k}})$ は 2^{n-k} 種類あり, 各 $\gamma(a_{h_1}, \dots, a_{h_{n-k}})$ の要素数は 2^k である. この性質より次の補題が成立つ.

[補題 6.6] $X = (x_0, \dots, x_{n-1})$, $Y = (y_0, \dots, y_{2^n-1})$ とし, $(n, 2^n)$ -復号化関数を $Y = f_d(X)$ とする. X の任意の部分系列を X_1 とし, $|S(X_1)| = k$ とする. また, Y の任

意の部分系列を Y_1 とし, $|S(Y_1)| = l$ とする. このとき, 出力値の個数が少なくとも $l/2^{n-k}$ となるような部分関数 $Y_1 f_d(X, Q) \upharpoonright X_1$ が存在する.

(証明) $X - X_1 = (x_{h_1}, \dots, x_{h_{m-k}})$ とする. $\gamma(a_{h_1}, \dots, a_{h_{m-k}})$ は 2^{m-k} 種類あり, $|S(Y_1)| = l$ なので少なくとも一つの $\gamma(a_{h_1}, \dots, a_{h_{m-k}})$ は Y_1 の出力変数を $l/2^{n-k}$ 以上含む. その集合を $\gamma(a_{h_1}^*, \dots, a_{h_{m-k}}^*)$ とする. $X - X_1$ への代入として $Q^* = (a_{h_1}^*, \dots, a_{h_{m-k}}^*)$ とすると, 部分関数 $Y_1 f_d(X, Q^*) \upharpoonright X_1$ は $\gamma(a_{h_1}^*, \dots, a_{h_{m-k}}^*)$ に属する出力変数の出力値の個数は, 少なくとも $l/2^{n-k}$ である. 従って $Y_1 f_d(X, Q^*) \upharpoonright X_1$ の出力値の個数は少なくとも $l/2^{n-k}$ である. ■

この補題と, 6.1 の結果より次の定理が成立つ.

[定理 6.7] $(n, 2^m)$ -復号化関数を $(y_0, \dots, y_{2^m-1}) = f_d(x_0, \dots, x_{m-1})$ とする. このとき, 境界配置の仮定を満たすならば,

$$A = \Omega(2^m \cdot n)$$

が成立つ.

(証明) R を C_{f_d} が埋め込まれる凸領域とする. R の直径 D によって, R の境界は2つの部分境界に分けられるが, 少なくとも一方の部分境界上には $G(C_{f_d})$ の $\lfloor L/2 \rfloor$ 以上の入力節点がおかれている. その部分境界を B_1 とする. B_1 におかれた $\lfloor L/2 \rfloor$ 個の入力節点に着目する. この入力節点に対して B_1 を2分割したときの部分境界を E_1, E_2 とする. また, E_1 と E_2 の接する点を通り, D に垂直な弦の長さを L とする.

$G(C_{f_d})$ の 2^m 個の出力節点の少なくとも 2^{m-1} 個は L の弦によって分けられる R の 2 つの部分境界のうち的一方の上におかれる。今、 E_2 を含む部分境界上に 2^{m-1} 以上の出力節点がおかれていると仮定してよい。この部分境界を F_2 とする。

E_1 上におかれる入力節点に対応する入力変数系列を X_1 とし、 F_2 上におかれる出力節点に対応する出力変数系列を Y_1 とする。このとき、

$$\begin{aligned} |S(X_1)| &= \lfloor L^m/4 \rfloor, \\ |S(Y_1)| &\geq 2^{m-1} \end{aligned}$$

である。補題 6.6 より、出力値の個数が $2^{\lfloor L^m/4 \rfloor - 1}$ 以上もつ部分関数 $Y_1 f_d(X, Q) | X_1$ が存在する。従って、補題 6.3 より、 $G(C_{f_d})$ は $\pi(S(X_1))$ と $\pi(S(Y_1))$ を結ぶ節点独立な道を少なくとも $\lfloor L^m/4 \rfloor - 1$ 本含む。従って、

$$L \geq \frac{1}{\binom{m}{2}} (\lfloor L^m/4 \rfloor - 1)$$

が成立つ。また、 R の周長を P とすると、

$$P \geq 2^m + m$$

となる。命題 4.2 (4.1) より

$$A = \Omega(2^m \cdot n)$$

が成立つ。 ■

$(n, \lceil \log m \rceil)$ -符号化関数についても同様にして示すこ

とができる。簡単のため $n = 2^{2k}$ と仮定する。 $(2^{2k}, 2k)$ -符号化関数 $(y_0, \dots, y_{2k-1}) = f_e(x_0, \dots, x_{2^{2k}-1})$ の入力変数を添字によってそれぞれ 2^k 個の変数を含む 2^k 個の集合

$$J_i = \{x_i, \dots, x_{i+2^k-1}\} \quad (i=0, 2^k, \dots, 2^{2k}(2^k-1))$$

に分割する。符号化関数の定義より、ちょうど1個の入力変数が1になるので、各 J_i の入力値の個数は 2^k とおりある。このとき、出力変数の下位 k ビット (y_0, \dots, y_{k-1}) の出力値はすべて異なり 2^k とおりある。このことから次の補題が成立つ。

[補題6.7] $X = (x_0, \dots, x_{2^{2k}-1})$, $Y = (y_0, \dots, y_{2k-1})$ とし、 $(2^{2k}, 2k)$ -符号化関数を $Y = f_e(X)$ とする。 X の任意の部分系列を X_1 とし、 $|S(X_1)| = p$ とする。また、出力変数の下位 k ビット系列 (y_0, \dots, y_{k-1}) の任意の部分系列を Y_1 とし、 $|S(Y_1)| = q$ とする。このとき、出力値の個数が $p/2^{k-q}$ 以上もつような部分関数 $Y_1 | f_e(X, Q) | X_1$ が存在する。

(証明) $|S(X_1)| = p$ であり、集合 J_i は 2^k とおりあるので、 X_1 の要素を $p/2^k$ 以上含むような J_i が少なくとも1つ存在する。その集合を J_i^* とする。 J_i^* の $p/2^k$ の入力変数に着目する。このとき、この入力変数に対する入力値の個数は $p/2^k$ あり、これらの入力値に対して (y_0, \dots, y_{k-1}) の出力値はすべて異なる。 $(y_0, \dots, y_{k-1}) - Y_1$ の出力値の個数は高々 2^{k-q} なので、 Y_1 の出力値の個数は $o(Y_1)$ とすると、

$$o(Y_1) \cdot 2^{k-q} \geq p/2^k$$

なければならぬ). 従って,

$$o(Y_1) \geq p/2^{2k-\delta}$$

が成立つ. ■

補題 6.7 において,

$$p = 2^{2k}/4 (= 2^{2k-2}),$$

$$\delta = \lfloor k/2 \rfloor$$

とおけば, 任意の X_1, Y_1 に対する部分関数 $Y_1 f_e(X, Q)$ (X_1 の出力値の個数は少なくとも $2^{\lfloor k/2 \rfloor - 2}$ となる). 従って, 補題 6.3 を用いると, $(n, \lceil \log n \rceil)$ -符号化関数を実現する組合せ回路 C_{f_e} の回路グラフ $G(C_{f_e})$ は, 任意の X_1 ($|S(X_1)| = \lfloor n/4 \rfloor$), Y_1 ($|S(Y_1)| = \lfloor \log n / 4 \rfloor$) に対して, $\pi(S(X_1))$ と $\pi(S(Y_1))$ を結ぶ節点独立な道を $\frac{1}{4} \log n - 2$ 本以上含む. 定理 6.7 の証明と同様の境界の分割をすることにより, 次の定理が成立つ.

[定理 6.8] $(n, \lceil \log n \rceil)$ -符号化関数を実現する組合せ回路に対して, 境界配置の仮定を満たすならば,

$$A = \Omega(n \log n)$$

が成立つ. ■

$(n, 2^n)$ -復号器, $(n, \lceil \log n \rceil)$ -符号器は, 図 6.2, 図 6.3 の回路構成により, 境界配置の仮定のもとで, それぞれ $O(2^n \cdot n)$, $O(n \log n)$ の面積で実現できる. 従って, 定理 6.7, 6.8 で得られた面積下界はいずれも定数の範囲を除いて最良のものである.

定理 6.7, 6.8 で得られた面積下界はいずれも計算時間には無関係に成立つ. もし $(n, 2^n)$ -復号器 ($(n, \lceil \log n \rceil)$ -符号器) を計算時間 $O(n)$ ($O(\log n)$) での実現を考えるならば, この回路グラフは葉数 2^n (n) の完全二分木を含むので, 境界配置の仮定のもとでは $A = \Omega(2^n \cdot n)$ ($A = \Omega(n \log n)$) となる. ここでの結果は, 計算時間を考慮しなくとも $A = \Omega(2^n \cdot n)$ ($A = O(n \log n)$) となることを主張している. なお, $(n, 2^n)$ -復号器, $(n, \log n)$ -符号器をそれぞれ $O(n)$, $O(\log n)$ の計算時間で実現するときは, 面積がそれぞれ $O(2^n \cdot n^2)$, $O(n \log^2 n)$ となる実現しか知られていない.

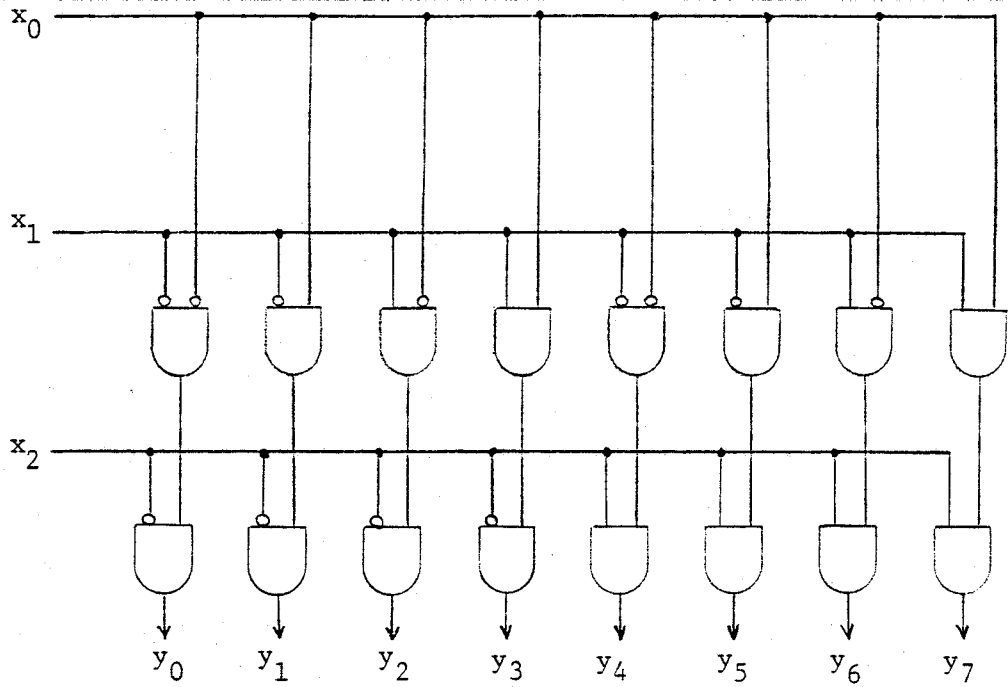


図6.2 $A = O(2^n \cdot n)$ の $(n, 2^n)$ -復号器 ($n=3$).

Fig.6.2 $(n, 2^n)$ -decoder with $A=O(2^n \cdot n)$ ($n=3$).

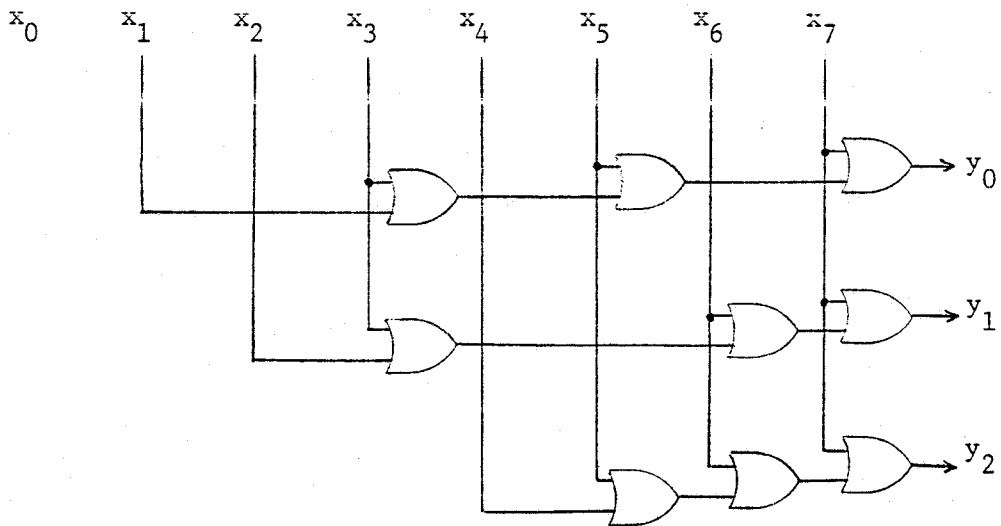


図6.3 $A = O(n \log n)$ の $(n, \lceil \log n \rceil)$ -符号器 ($n=8$).

Fig.6.3 $(n, \lceil \log n \rceil)$ -encoder with $A=O(n \cdot \log n)$ ($n=8$).

7. 結 論

本論文では、VLSI モデルのもとで、VLSI の面積時間複雑度および面積複雑度に関する基本的な問題の考察を行った。

まず、一般の自明でない論理関数のクラスに対して、VLSI モデルに境界配置の仮定を要請しない場合と、する場合について、その関数を実現する回路の面積時間積 AT^{α} ($\alpha > 1$) の下界を導出した。また、順序回路による構成を許した場合、組合せ回路に限定した場合について、面積の下界を導出する手法を示した。

一般に、複雑さの理論においては、下界を求める一般的な方法は存在しないので、できるだけ広いクラスに対する下界や広い範囲に適用できる手法が望ましい。ここで導出した下界は一般の論理関数のクラスに対するものであり、ここで面積の下界の導出法は広い範囲の回路に適用できるものである。さらに、そのような広い範囲に対する下界でありながら自明なものではなく、復号化関数、符号化関数、対称関数族、乗算、リーディングなど基本的に重要な関数に対して、ここで得られた下界のオーダーで実現できる。

本論文の結果より、回路の面積は実現法（組合せ回路か順序回路か）、境界配置の仮定、計算時間のいずれにも大きく依存することがわかる。例えば、 $(n, 2^n)$ -復号化関数を実現する回路（ $(n, 2^n)$ -復号器）の面積は表7.1 のような関係になる。

今後に残された課題としては、まず面積時間積 AT^{α} で回路を評価する場合、現実面ではどのような重み α を用いればよいかという問題をあげることができる。さらに、より現実面に見合った仮定を定式化し、VLSI モデルに

付加した場合の、面積や計算時間の評価に関する研究がある。また、VLSIの評価に関する研究はまだ始まったばかりであり、どのような評価項目を用いるか、その項目を評価できる理論的なモデルに関する研究も重要な主題である。

表7.1 復号器の面積

実現	境界配置を仮定するか	計算時間を $O(n)$ で実現するか	面積	参照
順序	する	しない	$\theta(n)$	定理5.2 定理5.3
	しない	する	$\theta(2^n/n)$	定理3.3
回路	する	する	$O(2^n), \Omega(2^n/\log n)$	定理4.5
組合せ	しない	する	$\theta(2^n)$	補題3.2
	する	しない	$\theta(2^n \cdot n)$	定理6.7
回路	する	する	$O(2^n n^2), \Omega(2^n n)$	補題3.2 定理6.7

謝 辞

本研究の全過程を通じて、直接理解ある御指導を賜わり、つねに励ましていただいた都倉信樹教授に心から深謝いたします。

大学院前期および後期課程において御教示、御指導いただいた情報工学科田中幸吉教授、嵩忠雄教授、藤澤俊男教授、高島堅助教授、図書館情報大学木澤誠教授、大阪大学産業科学研究所豊田順一教授に心から感謝いたします。

大学院を通じて御指導いただいた田村進一助教授、谷口健一助教授、宮原秀夫助教授に心から感謝いたします。

本研究を通じて有益な御討論、御指導いただいた荒木俊郎講師に心から感謝いたします。

本研究を通じて有益な御助言、御指導いただいた萩原兼一博士に心から感謝いたします。

在学中、御討論いただいた都倉研究室の諸氏に心から感謝いたします。

文 献

- (1) R.P.Brent and H.T.Kung : "The Area-Time Complexity of Binary Multiplication," Technical Report CMU-CS-79-136, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa. (July 1979).
- (2) R.P.Brent and H.T.Kung : "The Chip Complexity of Binary Arithmetic," Proceedings 12th Annual ACM Symposium on Theory of Computing, ACM, pp.190-200 (April 1980).
- (3) R.P.Brent and H.T.Kung : "On the Area of Binary Tree Layouts," Information Processing Letters, Vol.11, No.1, pp.46-48 (August 1980).
- (4) B.Chazelle and L.Monier : "A Model of Computation for VLSI with Related Complexity Results," Technical Report CMU-CS-81-107, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa. (February 1981).
- (5) I.Deegan : "Concise Cellular Array for Multiplication and Division," Electronics Letters, Vol.7, No.23 (November 1971).
- (6) 藤澤, 嵩 : 電子通信用数学 II 離散構造論, 日科技社, 東京 (1977).
- (7) K.Hagihara, K.Wada and N.Tokura : "Effects of Practical Assumption Area Complexity in VLSI Computation," to appear in Lecture Notes in Computer Science, Springer-Verlag.

- (8) R.B.Johnson Jr. : "The Complexity of a VLSI Adder,"
Information Processing Letters, Vol.11, No.2, pp.92-93
(October 1980).
- (9) D.E.Knuth : The Art of Computer Programming Vol. 3:
Sorting and Searching, Addison-Wesley, Reading,
Massachusetts (1973).
- (10) H.T.Kung : "Let's Design Algorithms for VLSI Systems,"
Technical Report CMU-79-151, Department of Computer
Science, Carnegie-Mellon University Pittsburgh, Pa. (January
1979).
- (11) R.J.Lipton and R.E.Tarjan : "Applications of a Planar
Separator Theorem," SIAM Journal on Computing, Vol.9, No.3,
pp.615-627 (August 1980).
- (12) R.J.Lipton and R.Sedgewick : "Lower bounds in VLSI,"
Proceedings 13th Annual ACM Symposium on Theory of
Computing, ACM, pp.300-307 (May 1981).
- (13) C.A.Mead and L.A.Conway : Introduction to VLSI Systems,
Addison-Wesley, Menlo Park(1980).
- (14) K.Menger : "Zur Allgemeinen Kurventheorie," Fundamental
Mathematics, Vol.10, pp.96-115 (1927).
- (15) D.E.Muller and F.P.Preparata : "Bounds to Complexities of
Network for Sorting and for Switching," JACM, Vol.22, No.2,
pp.195-201 (April 1975).
- (16) F.P.Preparata and J.E.Vuillemin : "Area-Time Optimal VLSI
Networks Based on the Cube-Connected-Cycles," INRIA Rapports
de Recherche No.13 (March 1980).
- (17) J.E.Savage : The Complexity of Computing,
Wiley-Interscience, New York (1976).

- (18) J.E.Savage : "Area-Time Tradeoffs for Matrix Multiplication and Related Problems in VLSI Models," JCSS, Vol.22, No.2 pp. 230-242 (April 1981).
- (19) J.E.Savage : "Planar Circuit Complexity and the Performance of VLSI Algorithms," INRIA Rapports de Recherche, No.77 (April 1981).
- (20) C.D.Thompson : "A Complexity Theory for VLSI," Technical Report CMU-80-140, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa. (August 1980).
- (21) S.H.Unger : "Tree Realizations of Iterative Circuits," IEEE Transactions on Computers, Vol.C-26, No.4, pp.365-383 (April 1977).
- (22) J.Vuillemin : "A Combinational Limit to the Computing Power of V.L.S.I. Circuits," IEEE 21st Annual Symposium on Foundation of Computer Science, pp.294-300 (October 1980).
- (23) 和田, 萩原, 都倉 : "VLSI回路の面積時間複雑度," 信学技報, EC81-2 (1981-04).
- (24) 和田, 萩原, 都倉 : "VLSI回路における面積時間のトレードオフについて," 信学技報, EC81-16 (1981-06).
- (25) 和田, 萩原, 都倉 : " n 変数論理関数の面積時間複雑度," 信学論(D), Vol.J64-D, No.8, pp.676-681 (昭56-08).
- (26) K.Wada, K.Hagihara and N.Tokura : "Area-Time Bounds for VLSI Circuits," Proceedings 19th Annual Allerton Conference on Communication, Control, and Computing, p.234 (September-October 1981).

- (27) 和田, 萩原, 都倉: "面積時間積を用いた論理関数の複雑さについて," 信学技報, AL81-77(1981-11).
- (28) 和田, 萩原, 都倉: "境界配置条件を課したVLSIモデルにおける面積複雑度について," 信学技報, AL81-96(1982-01).
- (29) 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. J65-D, No. 4, pp. 478-485 (昭57-04).
- (30) K.Wada, K.Hagihara and N.Tokura: "Area and Time Complexities of VLSI Computations," Proceedings the 7th IBM Symposium on Mathematical Foundations of Computer Science, Mathematical Theory of Computations, IBM Japan, pp. 1-77 (June 1982).
- (31) 和田, 萩原, 都倉: "VLSIモデルにおける組合せ論理回路の面積下界について," 信学技報, AL82-30, (1982-09).
- (32) A.C.Yao: "The Entropic Limitations of VLSI Computations," Proceedings 13th Annual ACM Symposium on Theory of Computing, ACM, pp.308-311 (May 1981).
- (33) 安浦, 矢島: "論理関数を実現するのに必要な段数について," 信学論(D), Vol. J62-D, No. 9, pp.561-568 (昭54-09).
- (34) 安浦, 矢島: "論理回路のVLSI上での面積について," 信学論(D), Vol. J65-D, No. 8, pp. 1080-1087 (昭57-08).

付 録

(A-1) 図3.4の $D(k, d)$ の面積, 計算時間はそれぞれ,
 $A = O(2^k)$, $T = O(k)$ である.

(証明) $D(k, d)$ の面積, 計算時間をそれぞれ $A(k)$,
 $T(k)$ とする. 図3.4の構成より,

$$A(k) \leq 2A(k-1) + c_1 k^2 \quad (1)$$

$$T(k) \leq T(k-1) + c_2 \quad (2)$$

が成立つ. ここで c_1, c_2 は定数である. $A(1), T(1)$ は定数になるので, (1), (2) を解けば,

$$A(k) = O(2^k),$$

$$T(k) = O(k)$$

が成立つ. ■

(A-2) 図3.4の $D(k, d)$ を境界配置の仮定を満たすように配置すると, $A = O(2^k \cdot k^2)$ である.

(証明) 図A.1に $D(k, d)$ の配置を示す. この $D(k, d)$ の面積, 縦の長さ, 横の長さをそれぞれ $A(k), W(k), L(k)$ とする. 図A.1の構成より,

$$L(k) \leq 2L(k-1)$$

$$W(k) \leq W(k-1) + ck$$

が成立つ. ここで c は定数である. これらを解けば,

$$L(k) = O(2^k),$$

$$W(k) = O(k^2)$$

となる。従って、

$$A(k) = L(k) \cdot W(k) = O(2^k k^2)$$

が成立つ。 ■

(A-3) 図3.8の $E(2^k, e)$ の面積、計算時間はそれぞれ、 $A = O(2^k)$ 、 $T = O(k)$ である。

(証明) $D(2^k, e)$ の面積、計算時間をそれぞれ $A(k)$ 、 $T(k)$ とすると、図3.8の構成より、(A-1)の(1)、(2)が成立つ。従って、(A-1)と同様にして、

$$A(k) = O(2^k),$$

$$T(k) = O(k)$$

が成立つ。 ■

(A-4) 図3.23の $(n, \lceil \log(n+1) \rceil)$ -重み関数を実現する回路の面積に対して、 $A = O(n)$ である。

(証明) $n = 2^p$ とおく。図3.23の $W(2^p)$ の面積を $A(p)$ とする。図3.23の構成より

$$A(p) \leq 2A(p-1) + c p^2 \quad (3)$$

が成立つ。ここで、 c は定数である。また、 $p=1$ の場合

は1個の全加算器で実現できるので $A(1)$ は定数である。
従って、(3)より、

$$A(p) = O(2^p)$$

が成立つ。 ■

(A-5) 図3.25(a)の l 入力を k 個コピーする回路の面積は $O(k \cdot l^2)$ である。

(証明) $k=2^p$ とする。入力数 l , コピー数 2^p のときの面積 $A(p, l)$ とする。 $p=0$ のとき、図A.2(a)より、

$$A(0, l) \leq c_1 l^2 \quad (4)$$

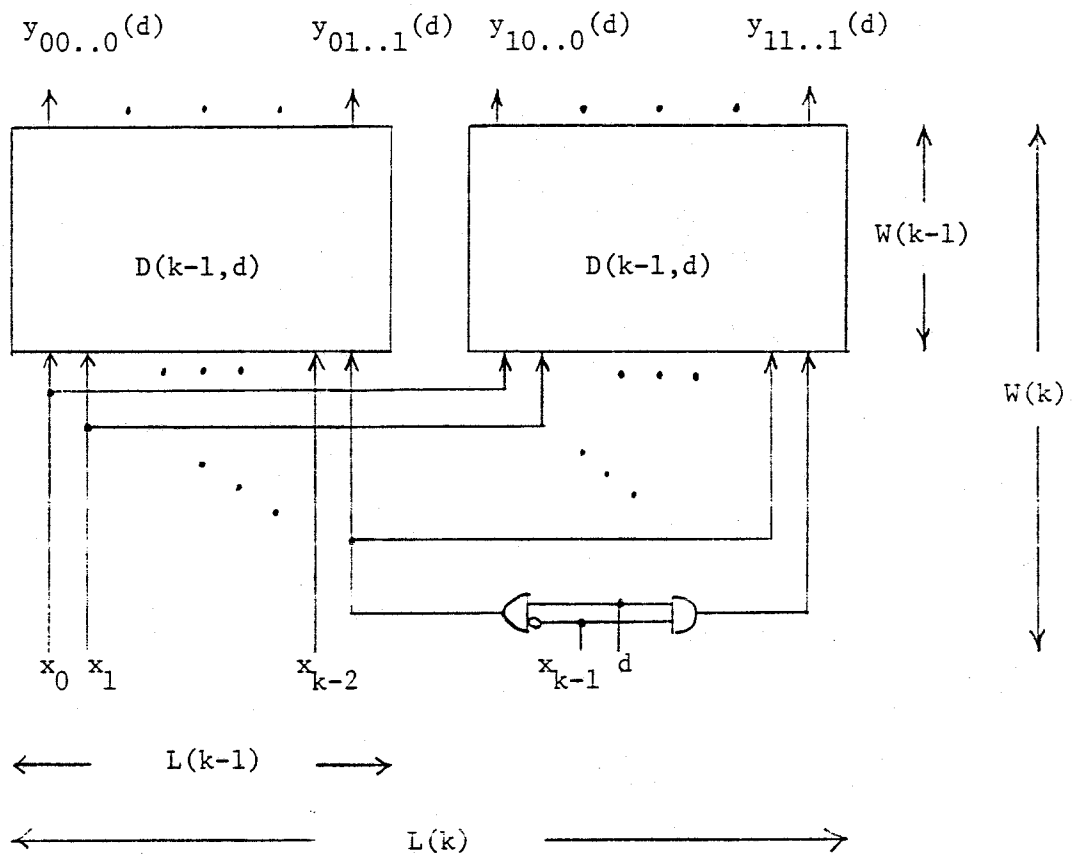
が成立つ。ここで c_1 は定数である。また、一般の場合は図A.2(b)より、 c_2 を定数とすると、

$$A(p, l) \leq 2A(p-1, l) + c_2 l^2 \quad (5)$$

が成立つ。(4), (5)より、

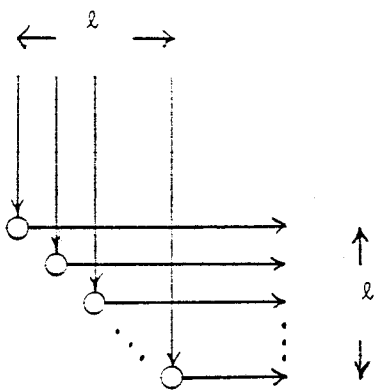
$$A(p, l) = O(2^p l^2)$$

である。 ■

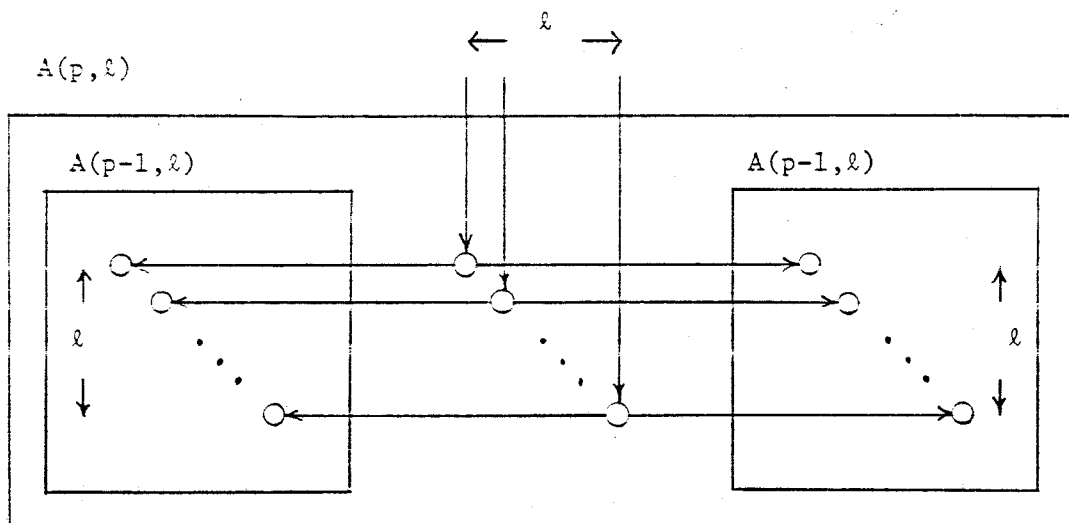


図A.1 境界配置の仮定を満たす $D(k, d)$ の配置.

Fig.A.1 Layout of $D(k, d)$ with the boundary layout assumption.



(a) $p = 0$



(b) general case

図A.2 図3.25(a)の回路の面積

Fig.A.2 Area of the circuit in Fig.3.25(a).