

2 段 バス ロジック に よ る 非 同 期 式 順 序 回 路 の 合 成

趙 慶 録 浅 田 邦 博

東京大学工学部

東京都文京区本郷7丁目3-1

あ ら ま し 本 稿 で は 内 部 状 態 割 当 に 線 形 コー ド を 用 い て 2 段 バス ロジック 構 造 に な る 新 し い 非 同 期 順 序 回 路 の 合 成 手 法 を 提 案 す る 。 状 態 割 当 は 遷 移 表 の 特 性 に 依 存 せ ず 一 義 的 に 行 わ れ , レー シング ・ フリー は 内 部 状 態 割 当 で は な く , 状 態 変 数 を 記 憶 す る 4 FET's セ ッ ト ・ リ セ ッ ト フ リ ッ プ ・ フ ロ ッ プ と そ れ を 駆 動 す る 回 路 で 実 現 す る 。 セ ッ ト と リ セ ッ ト の 駆 動 回 路 は 各 々 の バス 上 に 2 つ の ト ラ ン ジ ス タ (入 力 と 状 態 変 数 ) が 置 か れ る 2 段 バス ロジック 構 造 に な っ て い る 。 合 成 さ れ た 回 路 は 基 本 モー ド 動 作 で は ハ ザー ド フリー に な る コ ン パ ク ト な 回 路 で , ス タ ティ ッ ク な 電 力 消 費 が な い 。 ま た ハー ド ウェ ア の 大 き さ を 他 の 手 法 と 解 析 的 に 比 較 評 価 し た 。

Asynchronous sequential machine synthesis using Two-transistor pass logic

K. R. Cho and K. Asada

The faculty of Engineering, The university of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113

**Abstract** This paper presents a new design method for asynchronous sequential machines, which utilizes two-transistor pass logic circuits and linear code state assignment. Memory elements for internal state codes are set-reset flip-flops. Each set-reset control circuit for flip-flops is synthesized as a two-transistor pass logic circuit, which has only two input variables on each pass. Synthesized circuits are hazard and critical race free, and compact, compared with other methods. It is shown that the present method gives the smallest circuit in terms of numbers of transistor for large transition tables (more 40 cells).

## 2 段バスロジックによる非同期式順序回路の合成

Asynchronous sequential machine synthesis using Two-transistor pass logic

趙 慶 録 浅田邦博

K. R. Cho K. Asada

東京大学工学部

The faculty of Engineering, The university of tokyo

### I. はじめに

非同期回路の古典的な合成のアプローチとしてはクリティカル・レース・フリーになる最小な状態割当を見つけるのが主な研究として行われた<sup>[1]</sup>。よく知られている割当手法としてはLiu<sup>[2]</sup>, Tracey<sup>[3]</sup>, smith<sup>[4]</sup>らの single transition time(STT)割当て, Kuhl<sup>[5]</sup>の multicode transition time(MUSTT)割当手法などがある。しかしこれらのアルゴリズムは割当に時間がかかり, 大きい遷移表には適していない。かつ最小ビットの状態割当が必ず最小の回路を合成するとはいい難い<sup>[6]</sup>。回路合成手法としては最近になって, Makiら<sup>[7]</sup>はTracey<sup>[3]</sup>の割当手法を用いたパストランジスタネットワークとダブルインバータによる合成手法を提案した。しかしこの手法は回路が大きくなるとバスネットワークが長くなって, 回路の速度が落ちる。またWu<sup>[8]</sup>らはクロスカプルドインバータとツリーステートMOS回路で合成する手法を提案している。これらの手法はゲートレベルでは従来の回路より小さくなるが古典的な割当手法を用いている点で合成時間の問題がある。本稿では以下で取り上げる特徴の新しい非同期回路の合成手法を提案する。

- (1) 状態割当は線形コードを用いて回路の特徴に関係なく一義的に行う。
- (2) クリティカル・レース・フリーは状態割当からではなく回路で実現する。
- (3) 合成される回路は各バス上に2つのトランジスタが置かれる2段構造なので高速動作回路が作られる。

また合成された回路は従来の結果と解析的及びシミュレーションによる比較検討をした。

### II. 線形コード割当のクリティカル・レースフリー回路概念

本稿での回路合成は遅延素子を用いずに合成する。回路は次の状態で必ず安定するし, 内部状態が安定するまでは入力に変化しないnormal flow tableの基本モードで動作することを前提とする。回路は記号  $M(I, S, Z, \delta, \omega)$  で表現する。ここでI, SとZはそれぞれ入力, 内部状態, 出力を表す。また $\delta$ は状態遷移関数,  $\omega$ は出力関数で $\delta: S \times I \rightarrow S$ ,  $\omega: S \rightarrow Z$  (ムーア型順序マシン) である。状態割当は線形コード(one-hot code)を用いる。それは重み1のベクトル(各状態割当が違う変数のところに1ビットのみ1で残りは0ベクトル)で状態遷移表の特性には関係なく各状態にそれぞれ1つのベクトルが割り当てられる。従って, 状態割当の長さは状態の数と一致する。表1は順序回路マシン-Aとその線形コードが割当られた例である。

	I1	I2	I3	out	assignment
a	(a)	f	e	0	100000
b	a	(b)	d	1	010000
c	a	(c)	d	1	001000
d	(d)	b	(d)	0	000100
e	d	c	(c)	0	000010
f	d	(f)	c	0	000001

表1 順序マシン-Aと線形コード割当

線形コードを用いたクリティカル・レース・フリー回路は状態{y}を記憶する4FET'sフリッ

プーフロップとそのフリップフロップを駆動するセットーリセット回路で合成される。その回路モデルが図1である。

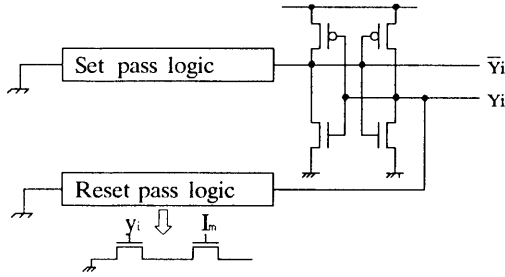


図1 2段パスロジックの回路モデル

本研究では回路の入力を  $m$  とし、 $I = \{I_1, I_2, \dots, I_m\}$  で表わす。このとき入力はすでにデコードされたものであり、入力ベクトルの中でどれか1つが"high"の状態になる。状態の集合は、 $S = \{S_1, S_2, \dots, S_n\}$  と置く。各状態が線形コードで割り当てられるので状態  $S_j$  を着目すると、 $y_1, y_2, \dots, y_j, \dots, y_n$  ( $y_j=1, y_i=0, i=0, \dots, n, i \neq j$ ) になる。さらに入力  $I_k$  で安定状態  $S_i$  から  $S_j$  への遷移を考えると内部状態は  $\{000, \dots, 1^i, \dots, 0\} \rightarrow \{000, \dots, 1^j, \dots, 0\}$  に変化する。この遷移で発生する中間状態としては  $\{000, \dots, 0\}$  と  $\{000, \dots, 1^i, 1^j, \dots, 0\}$  が存在する。ここで、 $\{000, \dots, 0\}$  状態は全ての遷移に共有されるが、後者は各遷移に唯一、存在する。本研究では後者を中間状態として通るように回路を合成する。そして内部状態遷移  $S_i \rightarrow S_j$  は2段階遷移になる。第1段階は  $S_i \rightarrow S_i \parallel S_j$ 、第2段階は  $S_i \parallel S_j \rightarrow S_j$  である。 $S_i \parallel S_j$  は2つの状態  $S_i, S_j$  のbit-or演算を意味する。図1のモデルで  $S_i \rightarrow S_j$  のセットーリセット条件は以下の式(1)のようになる。

$$\begin{aligned} \text{Set}(y_i) &= I_k y_i \\ \text{Reset}(y_i) &= I_k y_j \end{aligned} \quad (1)$$

全ての遷移を考慮した状態変数  $y_i$  に対するセット条件の一般式は

$$\begin{aligned} \text{Set}(y_i) &= I_1 \delta_{1,i} + I_2 \delta_{2,i} + \dots + I_m \delta_{m,i} \\ &= \sum_{\alpha=1}^m I_\alpha \cdot \delta_{\alpha,i} \end{aligned} \quad (2)$$

但し、 $\delta_{\alpha,i} = \sum_{j \in A_{\alpha,i}} y_j$ 、 $A_{\alpha,i} = \{j \mid \delta(I_\alpha, S_j) = S_i\}$

ここで、 $A_{\alpha,i}$  は状態遷移  $S_j \rightarrow S_i$  になる指数 "j" の

集合である。上記と同じようにリセット条件を一般化すると以下の式(3)になる。

$$\begin{aligned} \text{Reset}(y_i) &= I_1 \gamma_{1,i} + I_2 \gamma_{2,i} + \dots + I_m \gamma_{m,i} \\ &= \sum_{\alpha=1}^m I_\alpha \cdot \gamma_{\alpha,i} \end{aligned} \quad (3)$$

但し、 $\gamma_{\alpha,i} = \sum_{j \in B_{\alpha,i}} y_j$ 、 $B_{\alpha,i} = \{j \mid \delta(I_\alpha, S_i) = S_j\}$

ここで、 $B_{\alpha,i}$  は状態遷移  $S_i \rightarrow S_j$  になる指数 "j" の集合である。

例として、(2)(3)式をもとに表1の状態 "a" のセット条件とリセット条件を求める。状態 "a" は入力 I1 の下で安定状態であるし、"b", "c" I2 で安定である。現在の入力が I2 で I2 → I1 の変化が起きた場合、状態は "b", "c" から "a" に遷移する。従って、 $(I_1 \cdot y_b)$  或は  $(I_1 \cdot y_c)$  は状態 "a" のセット条件になる。今度は状態 "a" のリセット条件を考える。状態 "a" から起こり得る遷移は "f" か "e" で、すなわち  $\delta(I_2, a) = f$  で  $\delta(I_3, a) = e$  である。状態 "a" は "f" か "e" に遷移した後は必ずリセットされる。従って、 $(I_2, f)$  と  $(I_3, e)$  は "a" のリセット条件に含まれる。状態 "a" のセットーリセット条件を表で表したのが表2である。

set_map				reset_map			
	I1	I2	I3		I1	I2	I3
a	0	0	0	a	0	0	0
b	1	0	0	b	0	0	0
c	1	0	0	c	0	0	0
d	0	0	0	d	0	0	0
e	0	0	0	e	0	0	1
f	0	0	0	f	0	1	0

表2 表1の状態 "a" のセットーリセットマップ

また、表1のマシナー A の全ての状態 (a, b, ..., f) に対してセットとリセット条件を求めたのが表3である。

set_map				reset_map			
	I1	I2	I3		I1	I2	I3
a	b,c			a	f	c	
b		d		b	a		d
c		e		c	a		d
d	c,f		b,c	d		b	
e			a,f	e	d	c	
f		a		f	d		c

表3 マシナー A のセットーリセットマップ

### III. 2段パスロジックの回路合成

### III. 1 状態記憶回路と駆動回路

各状態のセットリセット条件を式(2)、(3)を用いて遷移表から求む。表2から状態"a"のセットリセットは式(4)になる。

$$\text{set}(a) = 11y_b + 11y_c$$

$$\text{reset}(a) = 12y_f + 13y_e \quad (4)$$

式(4)を図1の回路モデルで合成したのが図2の回路である。

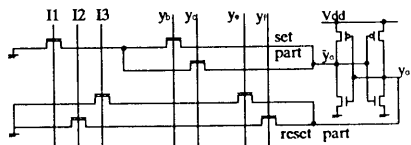


図2 マシン-Aの状態"a"の2段バスロジック

ここで、 $y_a \sim y_f$ は状態変数で、入力(11, 12, 13)は前段でデコードされたものでどれか1つの入力しか"high"にならない。回路は全てのセットとリセットのパスが1つの入力と状態変数で合成される2段バスロジックになっている。表3からマシン-A全体の回路を合成したのが図3である。

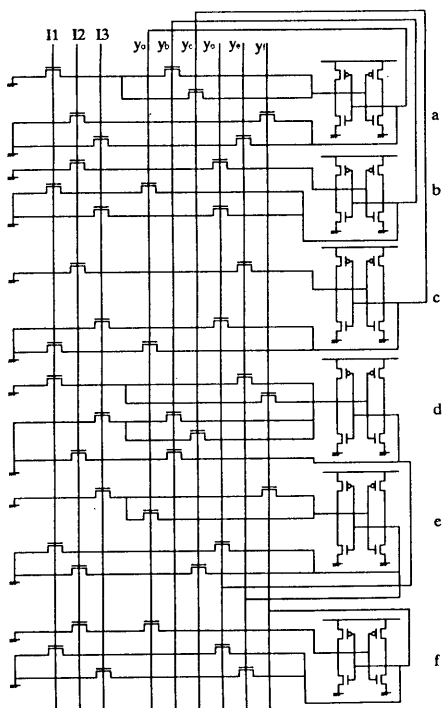


図3 本手法で合成したマシン-Aの回路

回路シミュレーションで動作の正しさを検証した結果が図4である。

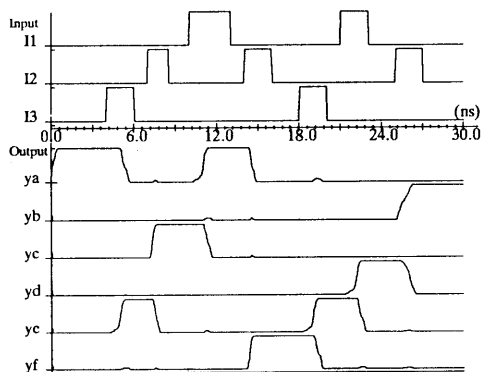


図4 マシン-Aの動作シミュレーション結果

### III. 2 出力回路合成

本稿での順序回路モデルはムーア型を前提しているので出力は状態のみに依存する。その式は次の(5)で表せる。

$$Z = g(y_1, y_2, \dots, y_n) \quad (5)$$

表1の回路の出力は状態"b"と"d"の時に"high"になる。 $Z = y_b + y_d$ である。この出力回路をプリアップとプルダウントランジスタで合成したのが図5の回路である。この回路をCMOSのように動作をさせるため出力を"high"にするpMOSの入力は状態"b", "d"変数の反転出力を用い、nMOSの入力は他の状態変数の正の出力を用いる。その結果、静的な電力消費がないCMOS回路の動作をする。

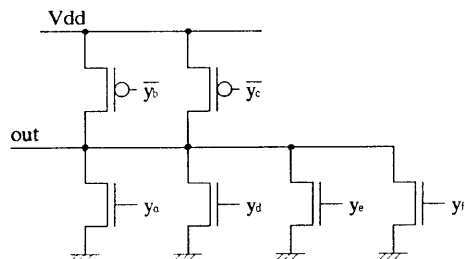


図5 表1 マシン-Aの出力回路

### IV. ハードウェアの大きさの解析的な検討

本章ではMaki<sup>[7]</sup>らの手法, Wu<sup>[9]</sup>らの手法

と本手法のハードウェアの限界をトランジスタ数の側面で見解的な比較を行う。

【用語の定義】

- n : 状態変数の数
- k : 状態の数
- k s : 安定状態の数
- m : 入力の数
- C : 遷移表のセルの数 (m × k)
- T p : 状態変数のセット・リセットバス上のトランジスタ数
- T m : 状態変数記憶回路のトランジスタ数
- T t : 各状態変数のトランジスタの数 (T p + T m)
- T total : 回路全体のトランジスタの数
- t p<sub>j</sub> : j 番目の入力でTwo-set分割対の数
- T t<sub>0</sub> : 遷移表のTwo-set分割対の数
- τ<sub>j</sub><sup>0</sup> : 状態変数 y<sub>j</sub> の "0" ブロック
- τ<sub>j</sub><sup>1</sup> : 状態変数 y<sub>j</sub> の "1" ブロック

IV. 1 Maki<sup>[7]</sup>の回路のトランジスタの数

この手法は第1段階で古典的なTracey<sup>[3]</sup>及びLiu<sup>[2]</sup>の状態割当手法を用いてクリティカル・レース・フリー割当を行う。第2段階では各入力での遷移対を作る。最後に各遷移対を区別する回路をバーストランジスタネットワークで合成する。回路の大きさ (T t) の式 (6) のようにバーストランジスタの数 (T p) と記憶回路 (T m) の和になる。

$$T t = T p + T m \quad (6)$$

入力 i j で遷移バス t p<sub>j</sub> が存在する場合はそれを区別する変数の数は log<sub>2</sub>(t p<sub>j</sub>) 必要であるし、トランジスタの数は t p<sub>j</sub> ([log<sub>2</sub>(t p<sub>j</sub>)] + 1) になる<sup>[3]</sup>。また1つの状態変数を合成するためのバーストランジスタの数は式 (7) になる。

$$T p \leq \sum_{j=1}^m t p_j ([\log_2 t p_j] + 1) \quad (7)$$

全回路のトランジスタの数は状態変数が n なので式 (8) になる。

$$T t_{total} \leq n (T p + T m) \quad (8)$$

トランジスタの数の近似式は式 (9) になる。

$$T t_{total} \propto n \times (4 + m \times \log_2 k \times k / 2) \quad (9)$$

式 (9) でわかるようにこの手法は状態変数の

数、入力と遷移分割対に比例する。

IV. 2 Wu<sup>[10]</sup>の手法のトランジスタの数

この手法は第1段階は遷移表から次状態変数表を作る。第2段階で各状態変数回路をCMOSで合成するため次状態表からpMOSとnMOSブロックの代数式を求め。最後にshort-cut手法を用いてCMOS回路を合成する。状態変数ごとの回路は次の式 (10) になる。

$$T t = T p + T m \quad (10)$$

T p を pMOS (T p<sub>p</sub>) と nMOS (T p<sub>n</sub>) にわけて求めると式 (11) (12) になる。

$$T p^p \leq |\tau_j^0 \cap \phi_j^1| \times (n - 2 + m) \quad (11)$$

$$T p^n \leq |\tau_j^1 \cap \phi_j^0| \times (n - 2 + m) \quad (12)$$

ここで、 $|\tau_j^1 \cap \phi_j^0|$  は  $\tau_j^1$  と  $\phi_j^0$  集合の norm である。また  $\phi_j^0 = \bigcup_{\alpha=1}^m \tau_{\alpha j}^0$  で、 $\tau_{\alpha j}^0$  は入力 i α で状態変数 y<sub>j</sub> = 1 になる状態の集合である。(n - 2 + m) は pMOS と nMOS バスロジックの入力の変数の数である。この Wu<sup>[8]</sup>の手法では回路 (T p) が各状態変数ごとに違うので、y<sub>j</sub> 変数の場合 T p<sup>j</sup> で表わす。回路のトランジスタの数は式 (13) になる。

$$T t_{total} \leq \sum_{j=1}^n T p^j + T m \times n \quad (13)$$

従って、この手法でのトランジスタの数は近似で式 (14) になって回路の大きさは状態変数の長さで入力の数に比例する。

$$T t_{total} \propto n \times (n - 1 + m) \times n \quad (14)$$

IV. 3 本研究の手法で合成

本研究の回路は4 FET's の状態記憶回路と駆動回路バスに2つのトランジスタが置かれるバスロジックである。各状態変数の回路は次の式で表わす。

$$T t = T p + T m \quad (15)$$

ここでバーストランジスタをセット回路 (T p<sup>s</sup>) とリセット回路 (T p<sup>r</sup>) に分ける。

$$T p = T p^s + T p^r \quad (16)$$

更に、式 (16) の各々の項を求めると式 (17) (18) になる。

$$T p^a = \sum_{\alpha=1}^m 2 \cdot |A_{\alpha i}| \quad (17)$$

$$T p^r = \sum_{j=1}^m 2 \cdot |B_{\alpha j}| \quad (18)$$

常数は各バスに2つのトランジスタが置かれることを意味する。トータルトランジスタの数は

$$T_{total} = \sum_{i=1}^k T p^i + k \times T_m \quad (19)$$

従って、本手法でのトランジスタ数の近似式は式(20)になる。

$$T_{total} \propto 4(C - ks) + 4k \quad (20)$$

#### IV. 4 3つの手法のトランジスタ評価

本節では近似式を用いてMaki<sup>[7]</sup>, Wu<sup>[8]</sup>本手法のトランジスタの数を検討した。近似式の中で用いられるパラメータはヒューリスティックな方法で代入した。Ttp = (m × k) / 2,

n = log<sub>2</sub>(Ttp), C = m × kである。結果としては入力小さいときはWuの手法が最もよくて入力が増加するごとに本手法の回路合成がコンパクトになる。その結果が図6(a), (b), (c)である。回路の大きさをパラメータに取った場合は40セル以下ではWuの手法がそれ以上大きい回路では本手法が適している。その評価を図7(a), (b), (c)に示している。

#### IV. 5 実際の回路合成での評価

本節では3つの手法を用いて例題の回路を合成した結果を評価する。表1 マシン-AはMaki<sup>[7]</sup>の例題から取った回路で、表4(a)のように4ビットで割り当てている。この場合の回路の大きさはMaki Ttotal=152, Wu Ttotal=92, 本手法 Ttotal=68で、本手法で合成する回路が最もトランジスタが少ない結果を示している。

	I1	I2	I3	Linear	4bit	3bit
a	(a)	f	e	100000	0000	000
b	a	(b)	d	010000	0110	100
c	a	(c)	d	001000	0101	101
d	(d)	b	(d)	000100	1111	110
e	d	c	(c)	000010	1000	011
f	d	(f)	e	000001	1010	010

表4 マシン-AにS T T割当を行った表

しかし、マシン-Aは表4(b)のように3ビットでもクリティカル・レース・フリー割当ができるので、この場合はMaki Ttotal=99, Wu Ttotal=42, 本手法 Ttotal=68 で Wu<sup>[10]</sup>の手法がよい結果を示している。表5はその他の例題の遷移表の特性を表したもので、図8は回路のトランジスタの数を表した図である。

#### V. まとめ

本研究では線形コード状態割当と2段バスロジックを用いる新しい非同期順序回路の合成手法を提案した。クリティカル・レース・フリーは状態割当ではなく回路合成によって実現した。合成される回路に対し、Maki<sup>[7]</sup>, Wu<sup>[8]</sup>らの手法と解析的な手法と例題でハードウェアの大きさの比較を行った。結果は状態割当が4ビット以下(約40セル)ではWu<sup>[8]</sup>の手法がそれ以上では本研究の手法が最も回路が小さくなる結果を得た。更に、合成された回路はクリティカル・レース・フリーであるし、基本モード動作遷移表ではハザド・フリーになる。出力回路はCMOS的な働きの回路合成ができた。今後は回路の動作速度の限界について検討を行う予定である。

#### 【参考文献】

- [1] S. H. Unger : Asynchronous Sequential Switching Circuit, John Wiley & Sons, 1969.
- [2] C. N. Liu, "A state variable assignment method for asynchronous sequential switching circuit," J. Ass. Comput. Mach., vol. 10, pp. 209-21, Apr. 1963.
- [3] J. H. Tracey, "Internal State Assignment for Asynchronous Sequential Machine," IEEE Trans. EC, vol. EC-15, no. 4, pp. 551-560, Aug 1966.
- [4] R. J. Smith, II, "Generation of internal state assignments for large asynchronous sequential machines," IEEE Trans. Comput., vol. C-23, pp. 924-932, Sept. 1974.
- [5] J. G. KUHL and S. M. REDDY, "A Multicode Single Transistion-Time State Assign-

ment for Asynchronous Sequential Machines," IEEE Trans. Comput., vol. C-18, pp. 541-547, Jun. 1969

- [6] C. J. Tan, "State assignments for asynchronous sequential machines," IEEE Trans. Comput., Vol. C-20, pp. 382-391, Apr. 1971.
- [7] S. Whitaker and G. K. Maki, "Pass-Transistor Asynchronous Sequential Circuits," IEEE J. SSC., vol. 24 No. 1, pp. 641-644, Feb 1989.

," IEEE J. SSC., vol. 24 No. 1, pp. 641-644, Feb 1989.

- [8] M. Y. Wu and I. N. Hajj, "Switching Network Logic Approach to Sequential MOS Circuit Design," IEEE Trans. CAD., vol. 8, NO. 7, pp. 782-794, Jul 1989.

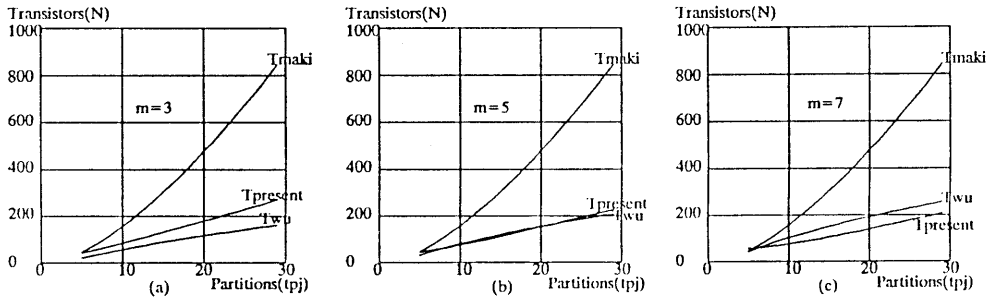


図6 遷移分割対をパラメータとしたトランジスタの数の評価  
(a)入力 = 3 (b)入力 = 5 (c)入力 = 7

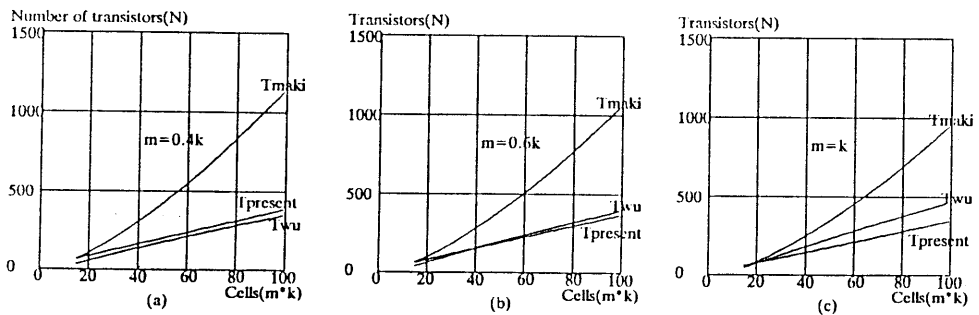


図7 回路の大きさセルをパラメータとしたトランジスタの数の評価  
(a)入力 = 0.4k (b)入力 = 0.6k (c)入力 = k

test circuit	input (m)	state (k)	cell (m×k)	unstable cells	assign bits	Maki[7] tr(N)	Wu[8] tr(N)	present tr(N)
sat-16	2	8	16	8	3	87	30	64
sat-21	3	7	21	9	3	123	36	64
sat-24	4	6	24	12	3	123	52	72
sat-32	4	8	32	15	4	356	88	92
sat-50	5	10	50	30	5	695	195	152
sat-63	7	9	63	34	6	988	278	172

表 5 例題の回路の状態遷移表の特性

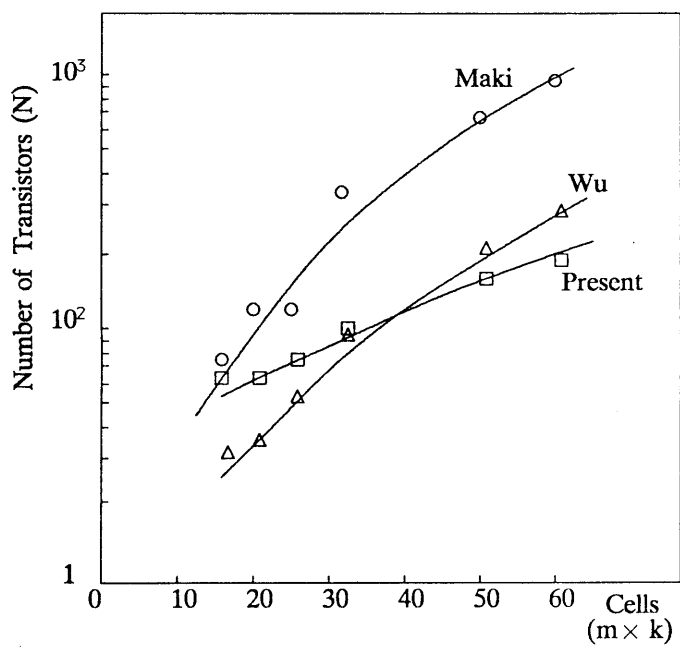


図 8 3つ (Maki[7], Wu[8], 本研究の手法) の手法を用いて表 5 の例題を合成した回路のトランジスタの数