

# テブナン定理を応用したMOSデジタル回路の論理シミュレーション方式

小島 智 山城 治 高橋 強 江口 一彦  
福田 秀樹 (株式会社日立製作所)

## 1. はじめに

論理LSIの大規模化、高機能化は目覚しく、ココソ、3年内には、1チップに搭載されるトランジスタ数は、100万個に達すると予想される。このような状況下で、低消費電力、チップサイズの面からMOS技術の使用が主流となり、双方向Transfer-gate、Wired-OR、Tri-state-gate等のMOSデバイスの特徴を生かした論理設計が一般的となっている。

現在、論理シミュレータは、大規模論理回路を検証する上で必要不可欠なツールとなっている。しかし従来のゲートレベル論理シミュレータは、ブール代数モデルを基礎としており、双方向性ゲート等のMOSデジタル回路特有の論理を直接シミュレーションするには無理があり、精度良く取り扱うには不十分である。このため現在までに、MOSデジタル回路を精度良く取り扱うためのシミュレーション方式が、数多く報告されている。(参考文献[1][2][3][4][5][6])

MOSSIM[4]を除いて、これらのシミュレーション方式の特徴は、論理値の強弱を表現するために、新たな論理値を追加している点にある。

MOSSEMは、MOSトランジスタをスイッチとしてモデル化し、回路を分割することにより、トランジスタレベルの論理シミュレーションを実現している。(しかしこれは、ユニットタイムの遅れを基礎としている。

LMOSII[5]は、実験式を用いて、MOSデジタル回路の伝播時間を精度良く計算するモデルを提唱している。またRSIM[6]は、MOSデジタル回路を線形回路にモデル化し、過渡状態における論理値とそこに到達する時間を精度良く求める計算法を提唱している。

本報告は、MOSデジタル回路の動作を精度良くシミュレーションする方式に関するものであり、テブナン定理を応用した論理シミュレーションのモデルとその計算法について述べる。これにより、双方向Transfer-gate、Wired-OR、Tri-state-gate等のMOSデジタル回路特有の論理を精度良くシミュレーションすることが可能となる。

## 2. MOSデジタル回路のモデル

一般にMOSデジタル回路は、MOSトランジスタ、ゲート(AND、OR、NAND、NOR)、Tri-stateゲート、フリップフロップ、ROM、RAM、PLA等の異種多様な素子により構成される回路である。

ここでは、MOSデジタル回路を精度良く取り扱うことを目的とした回路のモデルを提唱し、以下詳細に述べる。図2.1は、本モデルで対象とするMOSデジタル回路の例を示す。ここで、NAND、NOR等のゲートおよび図中子で示す何らかの論理機能を持つ論理素子は、信号の流れが入力から出力へと一方のみであり、出力側の負荷容量を充放電する能動素子である。また、MOSトランジ

スタと抵抗は、回路中の2つのノードを結び付ける双方向の受動素子であり、ノードは、対基板容量を集中して持つと捉える。表2.2に、MOSデジタル回路の構成要素とそのモデル化の考え方を整理して示す。MOSトランジスタは、スイッチ付の線形抵抗として表し、また論理素子は、内部抵抗を持つ電源と論理機能 $f$ を組み合わせたモデルとして表現する。ROM、RAM、PLA等の機能レベルに近い素子も論理素子として同様に扱う。こうすることで、レベルの異なる素子がすべて、線形抵抗、電源とスイッチにより線形回路として表現でき、素子相互間の動作を矛盾なく解析することが可能となる。

MOSトランジスタと論理素子の動作について、さらに詳細に述べる。図2.3に、MOSトランジスタの動作モデルを示す。MOSトランジスタは、ゲート電圧( $V_g$ )とそれ自身が持つスイッチングスレシヨールド( $V_{TH}$ )を比較して、論理的な「0」「1」を決定する。そして、この結果を入力として論理機能 $f$ を演算し、スイッチの状態(on/off)を決定するものとして取り扱う。図2.4に、論理素子(2入力ANDの例)の動作モデルを示す。論理素子は、入力側の電圧 $V_1$ および $V_2$ と素子自身が持つ $V_{TH}$ を比較して、論理的な「0」「1」を決定する。この結果を入力とし論理機能 $f$ を演算し、スイッチ $S$ の状態を決定する。その結果として出力端には、 $R_H$ または $R_L$ を内部抵抗として持つ電圧 $V_H$ または $V_L$ が印加され、出力側ノードの負荷容量を充電するための駆動力を発生するものとして取り扱う。

### 3. テブナン定理の応用による解析法

MOSデジタル回路を線形回路にモデル化し取り扱うことは既に述べたが、このモデル化により、MOSデジタル回路の動作解析にテブナン定理を応用した手法が適用できることを示し、その計算法を述べる。

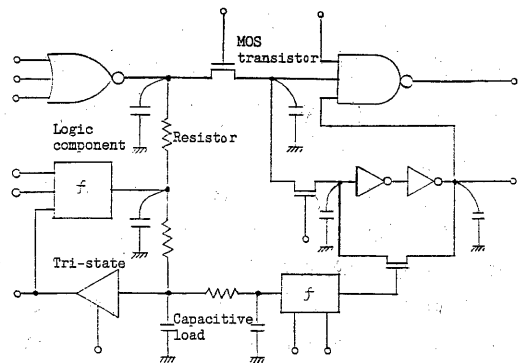


図 2.1 MOSデジタル回路の例

表 2.2 MOSデジタル回路の構成要素

	Component	Symbol	Model
Active component	Logic component		
	Resistor		the same as the left
Passive component	MOS transistor		
	Capacitive load		the same as the left

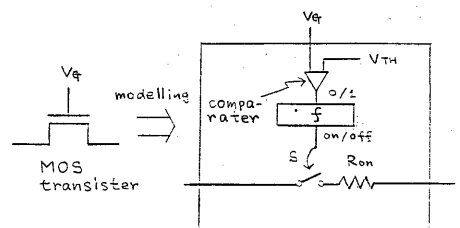


図 2.3 MOSトランジスタの動作モデル

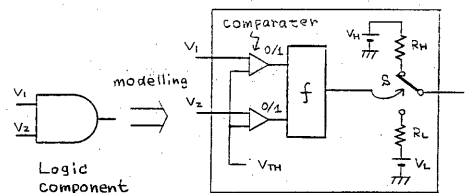


図 2.4 論理素子の動作モデル (2入力AND)

ここで、テブナン定理を以下に示す。(図3.1を参照のこと。)

[テブナン定理]

もし対象とする回路が線形回路であれば、グラウンドに対してあるノードから見た回路は、電源と内部インピーダンスの直列回路として表現できる。

以下図3.2に示すMOSデジタル回路のモデルを例に、テブナン定理を用いたMOSデジタル回路の動作解析法を述べる。図中で、ゲートG1の出力電圧は0V、トランジスタTR1は0Vの状態にあり、ノードN1とN2の電圧は、0Vと安定している。(ここで、容量の初期電荷は、容量と電源の直列で表現する。)このような定常状態にある回路で、ゲートG1の出力電圧が0Vから+5Vに変化したとする。この時ノードN2に注目するとノードN2の最終電圧とそこに到達する時間(以後ディレイ時間と呼ぶ。)は、図3.3に示すように、以下の手順で計算できる。

①テブナン定理を用い、ノードN2から見た等価回路を求める。ここで等価回路は、電源と抵抗の直列回路と容量の並列回路として計算する。

②計算したテブナン等価回路のパラメータより、ノードN2の最終電圧とディレイ時間は次のように求められる。

- ・最終電圧 = 電源電圧(+5V)
- ・ディレイ時間 = 比例定数 × RC 定数 ( $\alpha \cdot R \cdot C$ )

こうすることにより等価回路のパラメータは、回路中のすべてのパラメータより計算するため、過渡状態におけるノードの最終電圧とディレイ時間について精度良く計算することができる。

表3.4に、テブナン等価回路を示す。ここでノード等価回路とは、回路中のあるノードから見たテブナン等価回路をいい、また端子等価回路とは、ノード等価回路を計算する上で使用する便宜的なテブナン等価回路である。これは、ノードが同電位で互いに結合する端子の集合であるため、ノード等価回路が、そのノードを構成する端子のテブナン等価回路の並列として簡単に計算できるためである。表3.5には、テブナン等価回路のパラメータを導出するための変換式を示す。この変換式は、テブナン定理を基礎としているが、変換時に2次項以降を無視した1次近似式である。

またテブナン等価回路は、ForcingとFloatingの2つのタイプがあり、前者は、ノードまたは端子が能動素子に直接あるいは双方向素子を介し

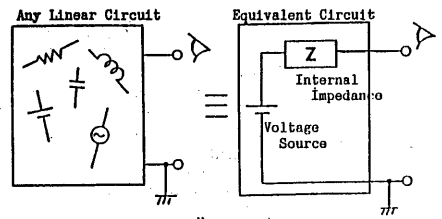


図3.1 テブナン定理

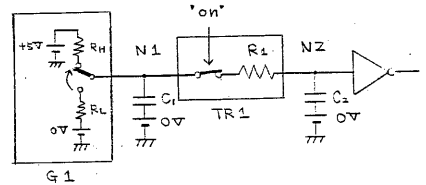
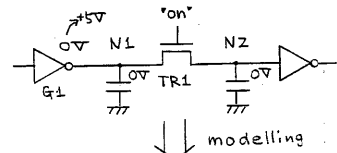


図3.2 MOSデジタル回路のモデル

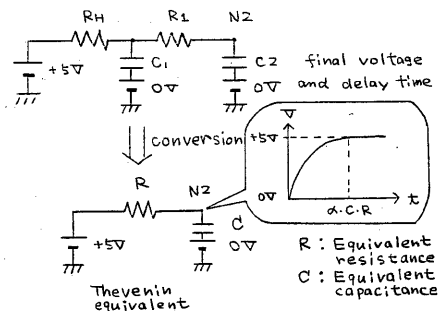


図3.3 ノードN2から見たテブナン等価回路

表 3.4 テブナン等価回路

Case		Node	Terminal
"Forcing"	Modified Thevenin equivalent		
	Parameters	Vn; Open-circuit voltage at node Gn; Internal conductance Cn; Collective capacitance at node Vcn; Voltage for initial charge	Vt; Open-circuit voltage at terminal Gt; Internal conductance Ct; Collective capacitance at terminal Vct; Voltage for initial charge
"Floating"	Modified Thevenin equivalent		
	Parameters	Vn; the same as above Gn; the same as above Cn; Collective capacitance at terminals connected to the node Cno; Capacitance inherent in node Vcn/Vcno; Voltage for initial charge	Vct; the same as above Gt; the same as above Ct; the same as above

表 3.5 等価回路パラメータの変換式

		Equivalent circuit conversion	Parameter conversion
Terminal equivalent	"Forcing"		$E = E_1$ $R = R_1 + R_2$ $C = \left(\frac{R_1}{R_1 + R_2}\right)^2 \cdot C_1$ $e = e_1$
	"Floating"		$R = R_2 + \left(\frac{C_1}{C_1 + C_2}\right)^2 \cdot R_1$ $C = C_1 + C_2$ $e = \frac{C_1 e_1 + C_2 e_2}{C_1 + C_2}$
Node equivalent	"Forcing"		$E = \frac{\sum \frac{E_i}{R_i}}{\sum \frac{1}{R_i}}$ $\frac{1}{R} = \sum \frac{1}{R_i}$ $C = C_{no} + \sum C_i$ $e = \frac{\sum C_i e_i}{\sum C_i}$
	"Floating"		$R = \frac{\sum C_i^2 R_i}{(\sum C_i)^2}$ $C = \sum C_i$ $e = \frac{\sum C_i e_i}{\sum C_i}$
	"Forcing" + "Floating"		$E = E_1$ $R = R_1$ $C = C_{no} + C_1 + C_2$ $e = \frac{C_{no} e_{no} + C_1 e_1 + C_2 e_2}{C_{no} + C_1 + C_2}$

つながる場合である。また後者は、ノードまたは端子が能動素子と切り離される場合である。図中でFlippingのノード等価回路の場合に、容量 $C_m$ と $C_{m0}$ を分けて持つのは、チャージシェア動作時の遅延時間を求めるためである。

図3.6に、テブナン等価回路を基礎にした回路のモデルを示す。ここで図中のボックスは、端子等価回路を表わす。いま図中のノードAに注目すると、ノードAのテブナン等価回路は、ノードAを構成する端子のテブナン等価回路の並列として計算できることがわかる。またノードAを構成する端子T1Aのテブナン等価回路は、ノードBのテブナン等価回路からトランジスタTR1の左側からの影響を除いたテブナン等価回路とトランジスタのオン抵抗の直列として計算できる。このようにノード等価回路は、そのノードを構成する端子のテブナン等価回路から簡単に計算することができ、逆に端子等価回路は、双方向素子を介して隣接するノードのテブナン等価回路から簡単に計算することができる。

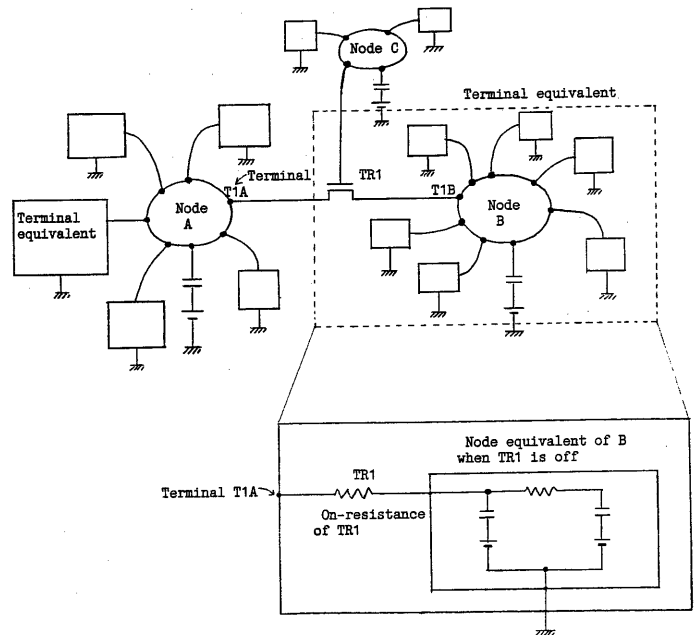


図3.6 テブナン等価回路を基礎にした回路のモデル

#### 4. 擬似イベントを用いた論理シミュレーション方式

ここで述べる論理シミュレーション方式は、MOSデジタル回路中の各ノードのテブナン等価回路を計算することを目的としており、擬似イベントという概念を導入し、Selective-trace法がこの計算に適用できることを示す。

擬似イベントは、テブナン等価回路パラメータの変化をいり、論理素子の出力変化またはMOSトランジスタのon/off(いわゆるイベント)により発生する。擬似イベントの対象となるのは、図4.1に示すようにMOSトランジスタと抵抗により構成される部分回路(以後トランジスタ回路と呼ぶ。)のノードと端子であり、それ以外の論理素子により構成される回路は、ゲートレベル論理シミュレータで用いられるイベント・ドリブ法により実現する。

図4.2に、イベントと擬似イベントによるシミュレーション制御のメカニズムを示す。イベント

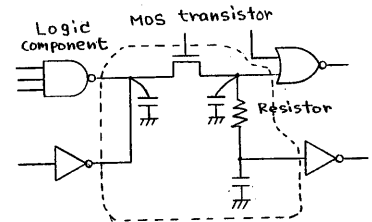


図4.1 トランジスタ回路

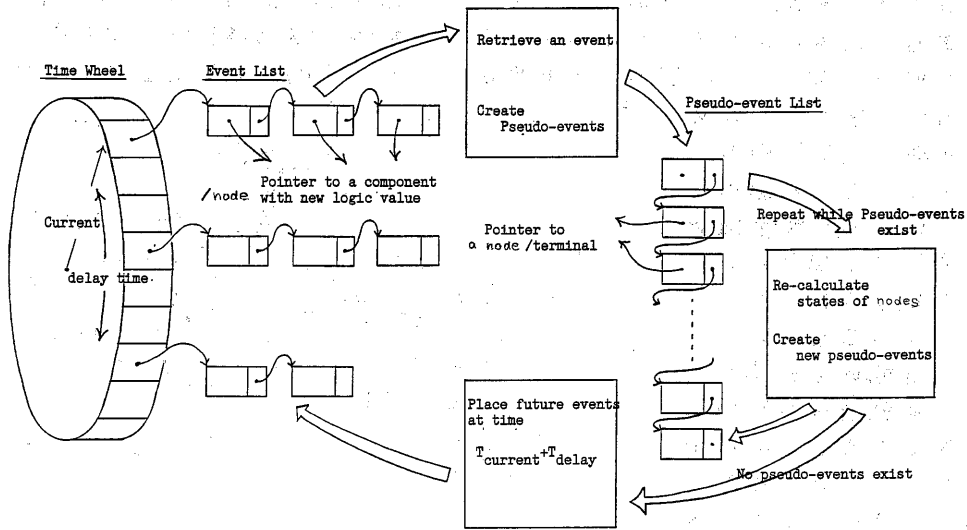


図 4.2 イベントと擬似イベントによるシミュレーション制御のメカニズム

は、将来変化する素子またはノードの論理値を持ち、タイムホイールにキューとして登録する。また擬似イベントは、次にテブナン等価回路パラメータを計算すべきトランジスタ回路中のノードまたは端子をポイントするキューである。シミュレーションは、イベント処理と擬似イベント処理という二つの処理を行うことにより実現する。以下に、1シミュレーション単位時間におけるシミュレーション制御の流れを示す。

(Step 1) イベントを取り出し、論理素子またはノードの論理値を変更する。

(Step 2) 論理素子に対して、もしファンアウトの素子が、MOSトランジスタまたはトランジスタ回路中のノードであれば、これを擬似イベントとして登録する。これ以外であれば、その素子の出力値を計算し、変化すればこれをイベントとしてタイムホイールに登録する。

ノードに対して、もしこのノードを入力とする論理素子が存在すれば、その素子の出力値を計算し、変化すればこれをイベントとしてタイムホイールに登録する。

(Step 3) もしイベントが存在すれば、Step 1へ飛ぶ。これ以外であれば、Step 4へ。

(Step 4) 擬似イベントを取り出し、ノードまたは端子のテブナン等価回路のパラメータを再計算する。

(Step 5) もし前回のパラメータと再計算したパラメータが不一致であれば隣接するノードまたは双方向素子の反対側端子を擬似イベントとして登録し、Step 6へ飛ぶ。

一致すれば、Step 6へ。

(Step 6) もし擬似イベントが存在すれば、Step 4へ飛ぶ。これ以外であれば、テブナン等価回路を再計算したノードについてのみ、論理値と遅延時間を計算し、これをイベントとしてタイムホイール

に登録する。

このように擬似イベントは、トランジスタ回路外部からの影響により変化するトランジスタ回路内の各ノードに対するテブナン等価回路を求めるために使用する。そしてこれは、タイムホイールの時間を一時的に停止し、トランジスタ回路内のノードに対するテブナン等価回路が安定するまで繰り返し処理する。トランジスタ回路が定常状態に達したら、テブナン等価回路を再計算したノードに対してのみ、論理値とディレイ時間を計算し、これをイベントとしてタイムホイールに登録する。

擬似イベントを用いた論理シミュレーション方式の特徴は、Selective-trace法を基礎にしている点にある。すなわち、MOSトランジスタ回路内の変化するノードのみを計算し、変化しないノードについては計算する必要がない点にある。

#### 5. まとめ

MOSデジタル回路の動作を精度良くシミュレーションする方式について述べた。本方式は、MOSデジタル回路を線形回路にモデル化し、回路中のノードの論理値とディレイ時間をテブナン定理を応用した等価回路のパラメータから計算する。そしてこれは、Selective-trace法を基礎にした擬似イベントによるシミュレーション方式により実現できる。

本シミュレーション方式は、論理/故障シミュレータ(HASL-GT)にインプリメントされている。論理シミュレーションの性能は、12Kイベント/秒(HITAC-M200H)を得ている。

#### [参考文献]

- (1) J. P. Hayes: "A Logic Design Theory for VLSI", Proc. of 2nd Caltech Conf. on VLSI, Jan. 1981
- (2) LOGIS user's Manual, ISD Inc. Santa Clara, CA
- (3) H. N. Nham, et al: "A Multiple Delay Simulator for MOS LSI Circuit", Proc. of 17th DA Conf., June 1980
- (4) R. E. Bryant: "MOSSIM: A Switched-level Simulator for MOS LSI", Proc. of 18th DA Conf., June 1981
- (5) M. E. Bush, B. D. Prazic: "LMOSII - A Logic Simulation for MOS Integrated Circuits", Proc. of '83 CICC
- (6) C. J. Terman: "RSIM - A Logic-level Timing Simulator", Proc. of '83 ICCD

- (7) T. Takahashi, S. Kojima, O. Yamashiro,  
K. Eguchi, H. Fukuda: "An MOS Digital  
Model Based on A Modified Thevenin  
Equivalent for Logic Simulation",  
Proc. of 21st DA Conf., June 1984
- (8) K. Eguchi, O. Yamashiro, S. Kojima,  
T. Takahashi, H. Fukuda: "A Logic Sim-  
ulation Technique for Gate/Transi-  
stor level circuits with Precise  
delay estimation", Proc. of ICCAD '84  
Nov. 1984