

## 確率的符号化時間記号シミュレーションによる タイミングエラー確率の解析

出口 豊<sup>†</sup> 石浦 菜岐佐<sup>††</sup> 矢島 脩 三<sup>†</sup>

非同期回路に対するタイミング検証において、遅延の不確実性が原因で生じるハザードなどによるタイミングエラーのおこる確率を求めることは、タイミングエラーの有無を判定することと同様に重要な問題である。本論文では、遅延の大きさが不確定で、その最小値と最大値が与えられているような論理素子によって構成された非同期回路において、ハザードなどによるタイミングエラーがおこる確率を正確に求める手法として、確率的符号化時間記号シミュレーションという新たな手法を提案する。本手法は、以前に提案したタイミング検証の一手法であり、記号シミュレーションによって正確な結果を得る「符号化時間記号シミュレーション」を基にしている。本方法では、信号線の依存関係を論理関数の形で保持することにより、従来回路中の再収れんによって不正確にしか求められなかったタイミングエラー確率を正確に計算することができる。

### Analysis of Timing Error Probability Using Probabilistic Coded Time-Symbolic Simulation

YUTAKA DEGUCHI,<sup>†</sup> NAGISA ISHIURA<sup>††</sup> and SHUZO YAJIMA<sup>†</sup>

On the verification concerned with timing, the changes of delay values in circuits have to be treated because the behavior of circuits depend on subtle timing relations. It is an important research theme to develop effective methods of timing verification that detects the existence of timing errors by incorporating the delay uncertainty. On the other hand, analysis of timing error probabilities in circuits caused by the delay uncertainty is also important. In this paper, we propose a new technique of calculating timing error probabilities named probabilistic CTSS. We are concerned with timing verification of logic circuits consisting of gates whose delay values are uncertain but specified by their minimum and maximum values. This technique is based on coded time-symbolic simulation (CTSS), which we proposed in our previous work. Probabilistic CTSS calculates timing error probabilities closer to reality than a conventional method which fails to do the task because of reconvergences in the circuits.

#### 1. はじめに

近年の集積回路技術の進歩に伴い、より大規模で高機能な論理システムが実現されるようになってきたが、その設計をいかに検証するかということが重要な課題となっている。特にタイミングに関する検証は複雑で困難な場合が多い。大規模な論理回路は同期式順序回路として設計されることが多く、これらに対するタイミング検証は比較的容易である。しかし、通信制御などの周辺論理のように非同期回路として設計される部分も存在している。非同期回路の設計法として

は、自己同期回路やスピードインディペンデント回路<sup>12)</sup>のように、遅延に依存しない回路を設計するものも存在するが、まだ十分に確立されているとはいえず、必ずしもすべての設計に対応できるわけではない。従って、回路の振るまいが微妙なタイミング関係に依存する場合には、製造条件や使用環境の違いに起因する素子遅延のばらつきを考慮する必要がある。論理シミュレーションでは、このような素子遅延の不確実性を最大/最小遅延モデル<sup>1)</sup>により解析する。このモデルによるシミュレーションは、高速ではあるが、回路中の再収れんが原因でシミュレーション結果が悲観的になることが知られており、回路中の設計誤りの有無を正確に判定することは難しい<sup>1)</sup>。そのため、遅延の不確実性を考慮したタイミング検証に関する研究が数多くなされている。これまでに提案された代表的な検証手法としては、時間記号シミュレーション<sup>3)</sup>、素子

<sup>†</sup> 京都大学工学部情報工学科

Department of Information Science, Faculty of Engineering, Kyoto University

<sup>††</sup> 大阪大学工学部情報システム工学科

Department of Information Systems Engineering, Faculty of Engineering, Osaka University

間の独立性を利用したタイミング検証法<sup>2)</sup>, 非決定性オートマトンの比較による検証方法<sup>4)</sup>, 符号化時間記号シミュレーション<sup>7)</sup>などが挙げられる。

これらの手法は, エラーの有無の判定だけを問題にする検証には有効であるが, 実際にはタイミングエラーは歩留りと深い関連があるので, エラーの存在のみならずエラーの起こる確率を考慮することも重要な問題であると考えられる。遅延のばらつきによるエラー確率の増大は歩留りの低下となって現れる。一方, 遅延のばらつきによるエラーの可能性があっても, その確率が十分小さく歩留りに与える影響が小さければ, これを無視することができる。本論文では, 論理回路におけるタイミングエラー確率の解析について考察する。タイミングエラーの確率を解析する方法としては, 信号値確率 (信号値が1となる確率) を各時刻, 各信号線について近似的に計算する方法が考えられる。しかし, この近似計算手法はすべての信号線の信号値確率を独立なものと仮定して計算を行うため, 再収れんが存在する回路では不正確な結果を出力する。最大/最小遅延モデルによるシミュレーションの結果が悲観的であるのに対し, この近似計算では楽観的な (信号値確率を過小評価する) 結果を出力することもありうる。

本論文ではエラー確率を正確に求める手法として, 確率的符号化時間記号シミュレーション (probabilistic CTSS) を提案する。この手法では, 回路中に再収れんが存在しても正確な結果を得ることができる。本手法は, タイミング検証の一手法である符号化時間記号シミュレーション<sup>7)</sup>を基にしている。符号化時間記号シミュレーションでは, すべてのゲートの遅延のとりうるすべての値の組合せについてシミュレーションを行うことにより, 遅延のばらつきによる回路のふるまいを完全に把握する。そして, 類似した計算をまとめて行うために不確定な遅延を持つゲートのとりうる遅延の値の可能性を符号化し, ブール変数を用いて表現する。この遅延の符号化によって, 不確定な遅延を持つゲートを含むシミュレーションを通常の記号シミュレーションに帰着させる。確率的符号化時間記号シミュレーションでは, ゲートのとりうる遅延の値の可能性を符号化する際に, 遅延の値の分布状況を考慮する。すると, ある信号線のある時刻の信号値確率は, 記号シミュレーションの結果得られた論理関数の真理値表密度として求めることができるようになる。この方法では, 信号線の依存関係が論理関数の形で保

持されているので, 回路に再収れんが存在しても正確な結果を得ることができる。また, 論理関数の内部表現として共有二分決定グラフ<sup>5)</sup>を用いることにより, 多くの論理関数を効率良く記憶するとともに, 論理関数の真理値表密度を効率良く計算することができる。

以下, 2章では信号値確率を利用したタイミングエラー解析法とその問題点について述べる。3章では, 符号化時間記号シミュレーションの概念について述べる。4章で確率的符号化時間記号シミュレーションの詳細およびエラー確率の計算方法について述べ, 5章では確率的符号化時間記号シミュレーションの結果の正確さについて議論した後に性能評価の結果について述べる。

## 2. タイミングエラー確率の近似計算手法

### 2.1 最大最小遅延モデル

現実の論理素子の遅延の値は製造条件や使用環境の違いによって変化する。従って, 非同期回路のように, 微妙なタイミングの違いによって回路のふるまいが変化する場合には, 素子遅延の不確定性を考慮して検証を行わなければならない。

論理シミュレーションでは, この不確定性を最大/最小遅延モデルを用いて解析する<sup>1)</sup>。しかし, このモデルには, 回路中に再収れんが存在する場合にはシミュレーション結果が悲観的になる (必要以上の不定値を出力する) という欠点が存在する。図1 (b)は図1 (a)の回路を最大/最小遅延モデルによりシミュレ

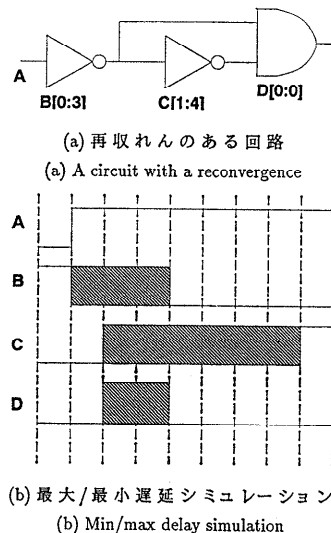


図1 最大/最小遅延シミュレーションの悲観性  
Fig. 1 Over pessimism in min/max delay simulation.

シミュレーションした結果である。信号値  $D$  中の不定値の存在はハザードの出現の可能性を示しているが、実際にはハザードの可能性はない。なぜならば、信号線  $B$  の立ち下がる時刻の方が、信号線  $C$  の立ち上がる時刻より常に早いからである。

2.2 タイミングエラー確率の近似計算手法

タイミングエラーの確率を解析する1つの方法として、信号値確率を計算する方法が考えられる。信号値確率は、信号値が1をとる確率と定義される。本節で述べる方法は、近似的な信号値確率を効率良く求めるものである。信号線  $a$  の時刻  $t$  における信号値確率を  $P_a[t]$  と表記し、ゲート  $f$  の入力信号線を  $x_1, x_2, \dots, x_n$  と表記すると、遅延値0のゲート  $f$  の出力信号線  $y$  の信号値確率は以下のように計算される。

$f$  が AND ゲートの時

$$P_y[t] = P_{x_1}[t] \times \dots \times P_{x_n}[t]$$

$f$  が NOT ゲートの時  $P_y[t] = 1 - P_{x_1}[t]$

上式の計算の結果は、回路中に再収れんが存在しなければ正確である。

遅延素子  $g$  の出力信号線  $y$  の信号値確率は以下のように計算できる。なお、 $d_{min}$  および  $d_{max}$  を  $g$  の遅延の最大値および最小値、 $\Delta$  を遅延のとりうる場合の数 ( $= d_{max} - d_{min} + 1$ )、 $p_d$  を遅延の値が  $d$  である確率とする。また、 $x$  を  $g$  の入力信号線とする。

$$P_y[t] = \sum_{d=d_{min}}^{d_{max}} p_d \times P_x[t-d].$$

上式により実際に信号値確率を計算するためには、遅延の値の分布を考慮する必要がある。遅延の値の分布として正規分布を仮定すると、 $p_d$  ( $d = d_{min}, d_{min} + 1, \dots, d_{max}$ ) は、以下の値をとる。

$$p_d = \frac{\Delta - 1}{2^{\Delta - 1}} \times \frac{C_{d-d_{min}}}{2^{d-d_{min}}}.$$

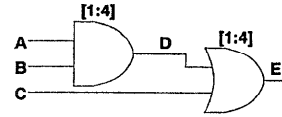
従って、 $P_y[t]$  は以下のように計算できる。

$$P_y[t] = \sum_{i=0}^{\Delta-1} \frac{\Delta-1}{2^{\Delta-1}} C_i P_x[t-d_{min}-i].$$

図2は1時刻めに信号線  $A$  が立ち上がった際の信号値確率の計算の例である。この例では、回路に再収れんが存在しないので、正確な結果が得られている。

しかし、この近似計算方法には次の2つの重大な問題が存在している。

- 1) 信号線間の信号値確率の依存関係を把握することができないため、回路中に再収れんが存在する際には、正確な信号値確率を求めることができない。
- 2) 時刻間での信号値確率の依存関係を把握できないため、正確な歩留りを計算することができない。



t	$P_A[t]$	$P_B[t]$	$P_C[t]$	$P_D[t]$	$P_E[t]$
0	0	1	0	0	0
1	1	1	0	0	0
2	1	1	0	0.125	0
3	1	1	0	0.5	0.016
4	1	1	0	0.875	0.109
5	1	1	0	1	0.344
6	1	1	0	1	0.656
7	1	1	0	1	0.890
8	1	1	0	1	0.984
9	1	1	0	1	1

図2 近似計算手法による信号値確率の計算  
Fig. 2 An example of approximate calculation.

本論文では、4章で正確な信号値確率およびエラー確率を計算する手法を提案し、5章では、近似計算手法による結果の不正確さについて議論する。

3. 符号化時間記号シミュレーション—CTSS

本章では、確率的符号化時間記号シミュレーション (*probabilistic CTSS*) の基礎となっている符号化時間記号シミュレーション (*coded time-symbolic simulation*, 以下 CTSS と略する) について簡単に述べる。

シミュレーションにおける時間のモデル化としては、離散時間と連続時間が考えられるが、CTSS および Probabilistic CTSS では、通常の論理シミュレーションと同様に離散時間モデルを扱う。非同期回路のタイミング検証において、離散時間モデルによる解析が十分であるかどうかに関しては確固たる結果は知られていないが、慣性遅延<sup>1)</sup>を考えない場合の二つの時間モデルの差は文献9)で議論されており、1) 離散時間モデルでは単位時間内に複数回信号値が変化する現象を正確に表現することができないため、連続時間モデルに比べ検証結果が不正確になる (一般的な計算法では悲観的となる) 例が存在すること、2) 離散時間モデルと連続時間モデルの精度を一致させるような単位時間幅が存在するが、これは極めて小さい、などの結果が示されている。しかし、慣性遅延<sup>1)</sup>の存在下では、ある時間幅以下の信号値変化は発生しないため、この連続時間モデルと離散時間モデルが一致する単位時間幅は2)の理論値よりもはるかに大きく、通常の論理シミュレーションでは、離散時間モデルによって十分な検証が行えるものと考えられる。

遅延の不確定性のモデルとしては、遅延の値は未知ではあるが変化しないものと仮定する。このような仮

定の下では、上下限を指定された遅延については、そのとりうる場合を数え上げることが可能になる。図1(a)の回路を例にとると、NOTゲートBの遅延の値は、{0, 1, 2, 3}の4つの値のうちの1つをとり、NOTゲートCの遅延の値は、{1, 2, 3, 4}の4つの値のうちの1つをとる。よって、16通りの場合、すなわち(Bのとりうる遅延の値の数)×(Cのとりうる遅延の値の数)通りの場合についてシミュレーションを行えば、遅延のばらつきによる回路のふるまいを完全に把握できる。シミュレーションしなければならない場合の数は、不確定な遅延を持つゲートの数に対して指数的に増加するため、すべての場合を個別に扱うことは非効率的である。そこで、類似した計算をまとめて行うことで、効率の良いシミュレーションを行うことを考える。図1(a)の回路を再び例にとると、本手法では、ゲートBとゲートCの遅延値をそれぞれ、 $delay_B=(b_1, b_0)$  および  $delay_C=(c_1, c_0)$  のように符号化する。16通りの信号線の信号値は4つのブール変数による1つの論理関数として表現される。この符号化によって不確実性を持つ遅延を含むシミュレーションを通常の記号シミュレーション<sup>8)</sup>に帰着することができる。これがCTSSの基本的な考え方である。

CTSSでは、論理回路は入力信号値から出力信号値を計算する機能を持つ遅延値0の論理ゲートと、入力信号値を遅延値だけ遅らせる機能を持つ遅延ゲートからなるものとする。入出力端子に遅延のあるゲートは、論理ゲートの入出力に遅延ゲートを付加したものとする。また、配線やネットの遅延は、ゲート間の結線に遅延ゲートを挿入することにより扱う。信号線sのt時刻の信号値をs[t]と表記する。gの関数をf<sub>g</sub>、ゲートgの入力信号線をx<sub>1</sub>, x<sub>2</sub>, ..., x<sub>k</sub>と表記すると、論理ゲートgの出力信号線yの信号値は以下のように計算される。

$$y[t] = f_g(x_1[t], x_2[t], \dots, x_k[t]).$$

遅延ゲートに関しては、遅延の符号化を図3のように考える。ここで、ブール変数b<sub>1</sub>, b<sub>0</sub>は、セレクタの制御入力に相当し、4つの遅延の値のうちの1つを選択する機能を持っている。遅延の符号化を(b<sub>1</sub>, b<sub>0</sub>)のように行ったらとすると、遅延ゲートの出力信号線yの信号値は以下のようにして計算される。

$$y[t] = \bar{b}_1 \cdot \bar{b}_0 \cdot x[t] + \bar{b}_1 \cdot b_0 \cdot x[t-1] + b_1 \cdot \bar{b}_0 \cdot x[t-2] + b_1 \cdot b_0 \cdot x[t-3].$$

図4は、図1(a)の回路に対してシミュレーションを行った例である。本例では、遅延はゲートの出力の

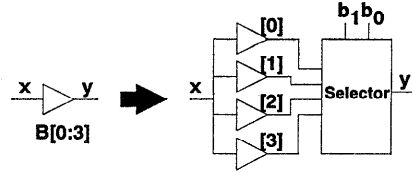


図3 セレクタを用いた不確定な遅延のモデル化  
Fig. 3 Modeling of an uncertain delay in CTSS.

b <sub>1</sub>	b <sub>0</sub>	delay <sub>B</sub>	c <sub>1</sub>	c <sub>0</sub>	delay <sub>C</sub>
0	0	0	0	0	1
0	1	1	0	1	2
1	0	2	1	0	3
1	1	3	1	1	4

(a) 遅延の値の符号化  
(a) Encoding delay values

t	A[t]	B[t]	C[t]	D[t]
0	0	1	0	0
1	1	b <sub>1</sub> + b <sub>0</sub>	0	0
2	1	b <sub>1</sub>	b <sub>1</sub> + b <sub>0</sub> + c <sub>1</sub> + c <sub>0</sub>	0
3	1	b <sub>1</sub> · b <sub>0</sub>	b <sub>1</sub> + c <sub>1</sub> · b <sub>0</sub> · c <sub>0</sub>	0
4	1	0	b <sub>1</sub> + c <sub>1</sub> + b <sub>0</sub> + c <sub>0</sub> · b <sub>1</sub> · c <sub>1</sub>	0
5	1	0	b <sub>1</sub> + c <sub>1</sub> + b <sub>1</sub> + b <sub>0</sub> + c <sub>1</sub> + c <sub>0</sub> + b <sub>0</sub> + c <sub>0</sub> · b <sub>1</sub> · c <sub>1</sub>	0
6	1	0	b <sub>1</sub> + c <sub>1</sub> + b <sub>1</sub> · c <sub>1</sub>	0
7	1	0	b <sub>1</sub> · b <sub>0</sub> · c <sub>1</sub> · c <sub>0</sub>	0
8	1	0	1	0

(b) シミュレーション結果  
(b) Simulation results

図4 符号化時間記号シミュレーションによるシミュレーション例  
Fig. 4 An example of CTSS.

みに存在すると仮定している。図4(a)は遅延の符号化方法を表し、図4(b)は信号線Aが1時刻めに立ち上がった際のシミュレーション結果、すなわち各時刻における各信号線の信号値を表している。2時刻めのCの論理関数  $\bar{b}_1 + b_0 + c_1 + c_0$  は、 $b_1 = b_0 = c_1 = c_0 = 0$ 、すなわち  $delay_B = 0$  かつ  $delay_C = 1$  であるときのみ1となり、その他の場合においては0となることを表している。信号線Dの信号値は0のまま変化していないが、これは最大/最小遅延モデルでは求めることができなかった正確な結果である。本手法では信号線の信号値はすべて論理関数の形で表現されているので、論理関数の内部表現が効率の良いシミュレーションを行うための鍵になる。我々の実現では、共有二分決定グラフ<sup>9)</sup>を用いている。

#### 4. 確率的符号化時間記号シミュレーション

3章で述べてきたように、CTSSはエラーの有無の判定だけを問題にする検証には有効である。しかし、実際には、タイミングエラーは歩留りと深い関連があ

るので、エラーの存在のみならずエラーの起こる確率を求めることも重要な問題であると考えられる。遅延のばらつきによるエラー確率の増大は歩留りの低下となって現れる。一方、遅延のばらつきによるエラーの可能性が存在しても、その確率が十分小さければ、歩留りに対してほとんど影響を与えないので無視することができる。2.2節で述べたように、タイミングエラーの確率を正確に計算することは難しい問題である。本章では、タイミングエラーの確率を正確に求める新しい手法として、確率的符号化時間記号シミュレーション (probabilistic CTSS) を提案する。

4.1 確率的符号化時間記号シミュレーションの概念

Probabilistic CTSSは、正確な信号値確率を求めることによってタイミングエラーの確率の解析を行う。

CTSSでは、図3のようにセレクタにより遅延の値を選択する回路を用いて遅延のモデル化を行っていた。このモデル化では1つの遅延の出力を1つのセレクタのデータ入力に割り当てている。Probabilistic CTSSにおける遅延のモデル化では、図5のように1つの遅延の出力を複数のデータ入力に割り当てることを許す。データ入力の重複は遅延値のばらつきに対応する。図5の例は、遅延値0, 1, 2, 3における重複度は、それぞれ1, 3, 3, 1であり、その遅延値をとる確率はそれぞれ1/8, 3/8, 3/8, 1/8であることを表現している。このような回路に対して記号シミュレーションを行うと、得られる論理関数の真理値表密度は信号値確率を表すようになる。Probabilistic CTSSは、信号値を表現する論理関数により、回路中の再収れんによる信号線の依存関係に関する情報を保持しているので、回路中に再収れんが存在しても正確な結果を得ることができる。

4.2 遅延の符号化および論理関数の表現

遅延のとりうる場合の数を  $\Delta (=d_{\max} - d_{\min} + 1)$  と表記する。遅延の値の分布として二項分布を仮定する

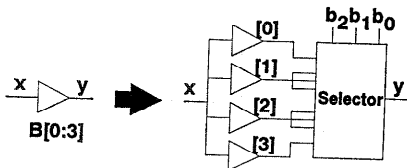
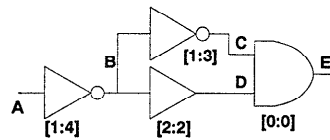


図5 確率的符号化時間記号シミュレーションにおける不確定な遅延のモデル化  
Fig. 5 Modeling of an uncertain delay in probabilistic CTSS.

と、 $i$  番目の遅延の値に関する重複度は  $i-1C_i$  となる。重複度の合計は  $2^{\Delta-1}$  となるので遅延の符号化には  $\Delta-1$  個の変数を要する。

符号語の割り当てについては、図6(b)のように重みが  $i$  の符号語を  $i$  番目の遅延値に割り当てる。この符号化方法は、図7のように、遅延の最小値が0, 最大値が  $N$  である遅延ゲートを、遅延の値が0か1である遅延ゲートが  $N$  個つながった回路に置き換えてシミュレーションすることを意味している。すなわち、図7(a)の回路の Probabilistic CTSS によるシミュレーション結果は、図7(b)の回路の CTSS



(a) 再収れんのある回路  
(a) A circuit with a reconvergence

$b_2$	$b_1$	$b_0$	delay <sub>B</sub>
0	0	0	1
0	0	1	2
0	1	0	2
1	0	0	2
0	1	1	3
1	0	1	3
1	1	0	3
1	1	1	4

$c_1$	$c_0$	delay <sub>C</sub>
0	0	1
0	1	2
1	0	2
1	1	3

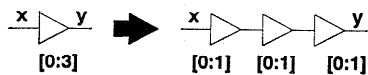
(b) 遅延の値の符号化  
(b) Encoding delay values

$t$	$A[t]$	$E[t]$	density
0	0	0	0
1	1	0	0
2	1	0	0
3	1	$b_2 + b_1 + b_0 + c_1 + c_0$	0.031
4	1	$\frac{c_1 + c_0 \cdot b_2 \cdot b_1 + b_2 \cdot b_0}{c_1 + c_0 \cdot b_1 \cdot b_0}$	0.094
5	1	$\frac{c_1 + c_0 \cdot (b_2 \cdot b_1 + b_2 \cdot b_0)}{c_1 + c_0 \cdot b_1 \cdot b_0}$	0.094
6	1	$c_1 + c_0 \cdot b_2 \cdot b_1 \cdot b_0$	0.031
7	1	0	0
8	1	0	0
yield		$c_1 + c_0$	0.75

(c) シミュレーション結果  
(c) Simulation results

図6 確率的符号化時間記号シミュレーションによるシミュレーション例

Fig. 6 An example of probabilistic CTSS.



(a) 遅延素子 (b) 遅延素子の連結  
(a) A delay gate (b) A cascade of delay gates

図7 不確定な遅延のモデル化

Fig. 7 An interpretation of an uncertain delay.

によるシミュレーション結果に等しくなる。この符号化方法は、CTSS において共有二分決定グラフが少ない記憶量で表現されていることを考えると、もっとも良い符号化であると思われる。また、我々が使用している共有二分決定グラフに基づく論理関数処理プログラム<sup>5)</sup>は、論理関数の真理値表密度をその論理関数の表現の大きさに比例する時間で求めることができるので、信号値確率を効率良く算出することができる。

図6はシミュレーションの例である。図6(c)は1時刻にAが立ち上がった時のシミュレーション結果である。表中の欄  $y$  が論理関数を、 $den$  が真理値表密度、すなわち信号値確率を表している。3時刻めから6時刻めまでの信号値確率は、ハザードが発生する可能性があることを示している。

#### 4.3 エラー確率の解析

前節のシミュレーション結果によれば、回路に出力の3時刻めから6時刻めまでの信号値確率はそれぞれ0.031, 0.094, 0.094, 0.031である。これらの確率は互いに依存関係を持っているので、ハザードが生じる確率をこれらの確率から単純に計算することはできない。しかし、論理関数  $E[3]$ ,  $E[4]$ ,  $E[5]$ ,  $E[6]$  は、これらの確率の間に存在する依存関係に関する情報を保持しているので、これらの論理関数からハザードが生じる確率を求めることができる。CTSSの結果解析手法として以前に提案した、有限オートマトンの記号シミュレーションによりシミュレーション結果と設計者の期待値とを照合する結果解析手法<sup>5)</sup>を、Probabilistic CTSSの確率解析に応用する。

まず、設計者の要求を正則表現で記述し、それをもとにして設計者の要求を満たす時に限り1を出力するような有限オートマトンを作成する。そして、Probabilistic CTSSの結果として出力された、各時刻の回路の出力信号線の論理関数をこの有限オートマトンに入力する。シミュレーションが終了した時点でオートマトンから出力される論理関数の真理値表密度が、回路が正常に動作する確率を示している。すなわち、得られた論理関数の真理値表密度が1であることは、その回路は素子の遅延の値に依存せずに正常に動作することを意味し、真理値表密度が1でない場合には、遅延の値の組合せによっては誤動作の可能性があることを意味している。

前節の例についてこの解析手法を適用すると、以下のようなになる。ハザードが生じないという仕様を表現したオートマトンは図8のようなになる。この2状態の

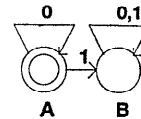


図8 仕様を表現するオートマトン  
Fig. 8 Automata to represent the specification.

オートマトンから、次のような遷移関数および出力  $ok$  を持った順序機械を作成できる。ここで、 $y$  は状態変数を、 $e$  は回路の出力信号値、すなわち順序機械の入力を表している。

$$(y', ok) = (y + e, \bar{y})$$

この順序機械を回路とともにシミュレーションすることにより、図6(c)の  $yield$  の欄に見られる論理関数を得ることができる。この論理関数の真理値表密度がハザードが起こらない正確な確率である。

正則表現を用いれば、本例のハザードが生じないという仕様ばかりでなく、信号値の変化が指定回数以下である、指定時間を越えるようなハザードは生じないなどの仕様も容易に記述でき、これを用いて回路が仕様どおりに動作する確率を求めることが可能である。

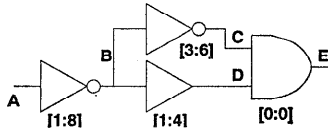
## 5. 考察および性能評価

### 5.1 シミュレーション結果の正確さについて

本節では、2.2節で述べた近似計算手法が不正確な結果を出力するのに対して、Probabilistic CTSSが正確な結果を得ることができる3つの例を示す。まず、近似計算方法によって求められる信号値確率が悲観的である例を、次に近似計算方法によって求められる信号値確率が楽観的である例を示す。最後に近似計算手法によって計算されたエラー確率の誤差が大きくなる例について示す。なお、本節の例では、遅延はゲートの出力のみに存在すると仮定している。

#### 5.1.1 近似計算手法では悲観的な信号値確率を出力する例

図9は、近似計算手法では信号値確率を過大評価してしまう例である。表中の欄  $conv.$  が近似計算手法によって求めた信号値確率を、PCTSSがProbabilistic CTSSによって求めた正確な信号値確率を表している。近似計算手法で計算された回路出力の信号値確率は、Probabilistic CTSSで求めることのできる正確な値に比べて、約10倍となっている。この悲観性は、再取れんによって生じる信号線Cと信号線Dの間の依存関係を考慮できないことに原因がある。この悲観性は、最大/最小遅延モデルによるシミュレーション



t ·	A	E	
		Conv.	PCTSS
0	1	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0.001	0
6	0	0.009	0.001
7	0	0.034	0.003
8	0	0.065	0.004
9	0	0.065	0.004
10	0	0.034	0.003
11	0	0.009	0.001
12	0	0.001	0
13	0	0	0
yield		0.799	0.968

図 9 近似計算手法による悲観的な解析結果

Fig. 9 Pessimistic result given by the conventional technique.

結果の悲観性と密接な関係にある。

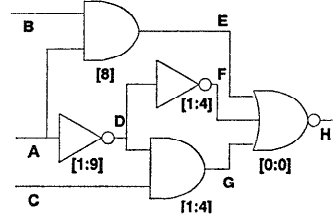
5.1.2 近似計算手法では楽観的な信号値確率を出力する例

信号値確率を過大評価することは、望まれることではないが、安全である\*。しかし、近似計算手法による解析では、信号値確率を過小評価してしまう場合が存在する。図 10 の例では、回路の出力の 9 時刻の信号値確率は、Probabilistic CTSS では 0.506 と正確に計算されるのに対して、近似計算手法では 0.376 という、正確な値に比べて小さい値が出力される。この楽観性も、信号線 E と F の信号値確率が独立であるという仮定で計算をしていることに原因がある。Probabilistic CTSS では、論理関数によって各信号線の信号値確率間に存在する依存関係を正確に把握することができるので、正確な信号値確率を求めることができる。

5.1.3 エラー確率の計算

エラー確率の計算においては、近似計算手法によって求めた値の誤差はさらに大きくなる。Probabilistic CTSS では信号線間の依存関係のみならず、同一信号線中の時刻間の依存関係も正確に把握できるので、4.3 節で述べた、オートマトンの記号シミュレーションに基づく解析手法を用いることにより正確なエラー

\* 本節の例はすべて信号値 1 が出力されることがエラーと考えられる例であるため、信号値確率の過大評価は悲観的、過小評価は楽観的となる。



t	A	B	C	H	
				Conv.	PCTSS
0	0	0	1	0	0
1	1	0	1	0	0
2	1	1	1	0	0
3	1	1	1	0	0
4	1	1	1	0	0.001
5	1	1	1	0	0.007
6	1	1	1	0.005	0.036
7	1	1	1	0.038	0.112
8	1	1	1	0.150	0.281
9	1	1	1	0.376	0.506
10	1	1	1	0	0

図 10 近似計算手法による楽観的な解析結果

Fig. 10 Optimistic result given by the conventional technique.

確率を求めることができる。一方、近似計算手法においては、異なる時刻間の依存関係を把握することは非常に困難であるので、近似計算手法によって求めたエラー確率は悲観的にならざるを得ない。5.1 節の図 9 は、近似計算手法による値の誤差が大きい例である。yield の欄は、回路の出力信号線 E にハザードが起こらない確率を表している。Probabilistic CTSS では、97% という値を計算するのに対して、近似計算手法では、各時刻において出力信号値が 0 となる確率の積とみなして計算するので、79% と計算される。

5.2 性能評価

プログラムを Sun SPARCstation 2 上に C 言語で実現し、性能評価を行った。

実験結果を表 1 に示す。circuit 欄の addern は、n ビット桁上げ加算器を、multn は、n ビット配列型乗算器を、dec 8 は 8 ビットデータセレクタを、enc 8 は 8 ビットプライオリティエンコーダを表し、N は回路のゲート数を表している。CTSS の欄は CTSS による結果を、PCTSS は、Probabilistic CTSS による結果を表している。ノード数の欄はシミュレーションに要した共有二分決定グラフのノード数を表し、CPU 時間はシミュレーションに要した CPU 時間 (単位は秒) を表している。遅延はゲートの出力のみに存在すると仮定し、全遅延について最小値を 1、最大値を Δ とした。回路の外部入力にランダムなパターン対を入力した際のシミュレーションを、回路中の信号値の

表 1 実験結果  
Table 1 Experimental results.

回路	N	A	CTSS		PCTSS	
			ノード数	CPU (sec)	ノード数	CPU (sec)
adder 1	6	4	7.2	1.09	15.1	1.08
		8	20.8	1.11	100.2	1.11
		16	54.2	1.21	597.0	1.57
adder 2	12	4	25.1	1.14	45.4	1.15
		8	111.4	1.21	400.1	1.30
		16	535.9	1.47	4342.9	2.47
adder 4	24	4	48.9	1.25	81.0	1.27
		8	183.6	1.40	559.3	1.58
		16	733.9	1.99	4793.1	3.79
adder 8	48	4	138.4	1.54	219.3	1.59
		8	633.9	2.00	1851.1	2.56
		16	10250.1	4.19	66125.2	15.87
adder 16	96	4	350.1	2.18	545.0	2.33
		8	1656.3	3.48	4862.5	5.23
		16	33486.3	10.67	—	—
mult 2	16	4	36.5	1.16	60.1	1.18
		8	140.8	1.26	500.0	1.35
		16	776.4	1.55	7300.1	2.71
mult 4	88	4	—	—	—	—
		4	243.7	1.21	418.5	1.22
		8	6768.0	1.49	23342.0	3.52
dec 8	17	16	—	—	—	—
		4	238.5	1.22	425.5	1.26
		8	3626.6	1.46	13240.5	2.13
enc 8	22	16	61823.5	5.09	—	—

変化が存在しなくなる時刻まで行った。表中の数は、10回のシミュレーションを行った際の平均値である。共有二分決定グラフのノード数は500,000ノード(約11MB)を上限とし、それ以上のノードが必要な場合にはシミュレーションを途中で打ち切った(表中の“—”で示す)。

この表から、加算器などの性質の良い回路においては、回路規模の大きさにかかわらず、Probabilistic CTSSは、実行時間、必要な記憶量ともにCTSSの2倍弱しか必要としないことがわかる。また、すべての配線に遅延を仮定した回路では、ゲートの出力のみに遅延を定義した場合と比較すると、扱える回路の規模が半分程度になると予想される。

遅延の幅の大きさに対しては、必要な記憶量は遅延符号化の際に要した変数の数にはほぼ比例することがわかる。実行時間は、符号化時間記号シミュレーションと比較して長時間かかっているが、イベント駆動方式を採用するなど、実現上の工夫で短縮は可能であると思われる。

## 6. おわりに

本稿ではエラー確率を正確に求める手法として、確率的符号化時間記号シミュレーションを提案した。本手法は、符号化時間記号シミュレーションを基にしており、再収れんの存在にかかわらず正確な結果を得ることができる。確率的符号化時間記号シミュレーションは、信号線間および時刻間の依存関係を論理関数の形で保持することにより正確な結果を得る。また、オートマトンの記号シミュレーションによって、正確なエラー確率を効率良く計算することができる。

本手法をそのまま現実規模の回路に適用することは、計算量の面から考えて困難であると思われる。現実規模の回路への対応法を見出すことは、符号化時間記号シミュレーションと共通の課題であるが、現在の2つのアプローチを考えている。

1つは、本手法の適用範囲を限定する、あるいは回路を分割し部分回路ごとに本手法を適用するというものである。通常の大規模回路の大部分は同期式順序回路として設計されているため、複雑な再収れんによるタイミング誤りの有無を正確に検証する必要のある回路部分は限られると考えられる。本手株の適用はこのような部分に限定し、残りの部分は従来の最大/最小遅延モデルや信号値確率に基づく手法で処理することで、現実規模回路を扱う手法が考えられる<sup>10)</sup>。

もう1つの方法は、本手法で大きな計算量を必要としている共有二分決定グラフによる論理関数処理部分に近似計算法<sup>11)</sup>を適用するというものである。

その他、正規分布でない遅延の値の分布への適応や、動的遅延モデル(遅延の値がシミュレーション中に変化することを可能とするモデル)への適応なども今後の課題としたい。

謝辞 共有二分決定グラフに基づく論理関数処理プログラムを提供して頂いた湊 真一氏(現在 NTT)に深謝します。また、有益な御議論を頂いた京都大学高木直史助教授、濱口清治博士、澤田宏氏に深謝いたします。

## 参考文献

- 1) Breuer, M. A. and Friedman, A. D.: *Diagnosis & Reliable Design of Digital Systems*, p. 308, Computer Science Press, California (1976).
- 2) Yoneda, T., Nakade, K. and Tohma, Y.: A Fast Timing Verification Method Based on the Independence of Units, *Proc. IEEE 19th Int.*

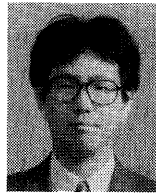


*Symposium on Fault-Tolerant Computing*, pp. 134-141 (1989).

- 3) Ishiura, N., Takahashi, M. and Yajima, S.: Time-Symbolic Simulation for Accurate Timing Verification of Asynchronous Behavior of Logic Circuits, *Proc. ACM/IEEE 26th Design Automation Conf.*, pp. 497-502 (1989).
- 4) Cerny, E., Rioux, P. and Berthet, C.: Comparison of Specification and Implementation for Asynchronous Circuits with Arbitrary Delays, *Proc. IFIP International Workshop on Applied Formal Methods for Correct VLSI Design*, pp. 704-720 (1989).
- 5) Minato, S., Ishiura, N. and Yajima, S.: Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation, *Proc. ACM/IEEE 27th Design Automation Conf.*, pp. 52-57 (1990).
- 6) Bryant, R.: Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Trans. Comput.*, Vol. C-35, No. 8, pp. 677-691 (1986).
- 7) Ishiura, N., Deguchi, Y. and Yajima, S.: Coded Time-Symbolic Simulation Using Shared Binary Decision Diagram, *Proc. ACM/IEEE 27th Design Automation Conf.*, pp. 130-135 (1990).
- 8) Carter, W. C., Joyner, W. H. and Brand, D.: Symbolic Simulation for Correct Machine Design, *Proc. ACM/IEEE 16th Design Automation Conf.*, pp. 280-286 (1979).
- 9) 石浦菜岐佐, 安浦寛人: 時間モデルとハザード検出問題の計算量の関係について, 電子情報通信学会技術研究報告, COMP 88-21 (1988).
- 10) 出口 豊, 石浦菜岐佐, 矢島脩三: 符号化時間記号シミュレーションと最大/最小遅延モデルとの結合手法, 情報処理学会全国大会論文集, 6R-1 (1991).
- 11) 石浦菜岐佐, 一宮敬弘, 白川 功: 論理関数処理の近似計算法について, 電子情報通信学会秋期大会, A-52 (1992).
- 12) 南谷 崇: 同期式プロセッサの限界と非同期式プロセッサの課題, 電子情報通信学会技術研究報告, FTS 90-45 (1990).

(平成4年6月4日受付)

(平成5年2月12日採録)



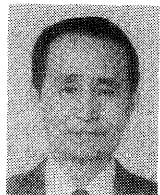
出口 豊 (正会員)

1967年生。1990年京都大学工学部情報工学科卒業。1992年同大学院修士課程修了。在学中に論理回路のタイミング検証の研究に従事。同年(株)東芝関西研究所に入社。現在はマルチメディア情報処理の研究に従事。



石浦菜岐佐 (正会員)

昭和36年生。昭和59年京都大学工学部情報工学科卒業。昭和61年同大学院修士課程修了。昭和62年1月京都大学工学部助手。平成3年3月京都大学工学博士。平成3年5月より大阪大学工学部情報システム工学科講師。論理回路の設計検証, テスト, 論理合成の研究に従事。電子情報通信学会, IEEE 各会員。



矢島 脩三 (正会員)

昭和8年生。昭和31年京都大学工学部電気工学科卒業。同大学院博士課程修了。工学博士。昭和36年より京都大学工学部に勤務, 昭和46年情報工学科教授。昭和35年京都大学第一号計算機 KDC-1 を設計稼働, 以来, 計算機, 論理設計, オートマトン等の研究教育に従事。著書は「電子計算機の機能と構造」(岩波書店, 57年)等。本学会元常務理事, 元会誌編集委員(地方), 元JIP編集委員。電子情報通信学会元評議員およびオートマトンと言語研専元委員長, North-Holland 出版元 IPL 編集委員, IEEE Senior Member。