設 計 自 動 化 60－25
（1991．12．13）

最小・最大遅延モデルを用いた論理回路のタイミング検証について

加島 重見　　木村 晋二　　羽根田 博正

（神戸大学 工学部）

本稿では、最小・最大遅延モデルを用いた新しい論理回路のタイミング検証手法を提案する。最小・最大遅延モデルは、実際の素子の遅延時間のばらつきを考慮したモデルで遅延時間として最大値と最小値を指定し、実際にはその中のどれかの値になるとするモデルである。従来、最小と最大の間を不定値でシミュレーションする手法が用いられてきたが、不定値の部分が拡大し、あまり有効な方法とは言えなかった。ここでは、最小と最大の間のすべての遅延の値について、各遅延に対応する出力系列を系列集合として扱う手法を示す。これにより、正確なタイミング検証ができる。タイミング検証問題としてハザード検出と非同期順序回路の検証について考察し、いくつかの例題で本手法の有効性を示す。

Timing Verification of Logic Circuits Using Minimum Maximum Delay Model

Shigemi Kashima　Shinji Kimura　Hiromasa Haneda

Faculty of Engineering, Kobe University

In this paper, timing verification of logic circuits using minimum-maximum delay model is discussed. In the model, the delay time of each logic element is specified as an interval between the minimum and maximum delay times. In the usual simulation, the time variance is manipulated as an unknown value, and only the too pessimistic results can be obtained. Here we propose a method to manipulate the time variance by a string set, and show examples of hazard detections and verification of asynchronaous circuits. This method works well when we analyze small circuits.

1 はじめに

近年の集積回路の進歩に伴い、大規模な論理回路の実現が可能となった。それとともに、論理回路の設計の正しさを検証することが重要になってきた。大規模な論理回路は同期式回路として設計されることが多いが、同期回路間のインターフェースなど非同期的な動作を行う部分のタイミング検証が重要な問題となっている。

現在、タイミング検証の方法としては、立ち上がり立ち下がり遅延による論理シミュレーションが最も広く行われている。しかし、正確なタイミング検証を行うためには、素子の遅延のばらつきを考慮することが必要である。

最小・最大遅延モデルは素子に割り当てる遅延時間に幅を持たせる方式で、実際の素子の遅延時間のばらつきを考慮したモデルとなっている。これまでに、素子の出力を最小遅延時間から最大遅延時間までの期間、不定値としてシミュレーションする方法が提案されたが、シミュレーションにより、不定値の部分が拡大し、実際の回路よりも悲観的な結果を与えるという問題点がある。

素子の遅延時間を記号として、ある入力パターンに対する素子の遅延の長さと時間の関係を連立不等式の形で求める方法も提案されているが、すべての場合を尽くす能力に問題がある[2]。また、CTLモデルチェックにおいて、CTLの式に対して回路の検証を行うという方法がある[3]が、状態の指数爆発の問題点がある。

本稿では、これらの問題点を解決する1つ方法としてシミュレーションの過程で系列集合を用い、最小と最大の間のすべての遅延の値について回路動作の解析を行う方法を提案する。本方法は、すべての遅延の組み合わせを取るので、最悪の場合、最大遅延時間の指数に比例した計算量を必要とする。しかし、指定された入力集合のみに対して回路の動作を求めるもので、多くの場合本手法を適用できる。

以下本論文では、2章で正則表現とオートマトンについて、3章で系列集合を用いた取扱いについて、4章でシミュレーションアルゴリズムリズムについて、5章で検証例について述べる。

2 準備

本稿では、以下で用いる基本的な定義となる正則表現と有限オートマトンとについて文献[4]に従う。

2.1 正則表現

正則表現は、連接、和(+)、包含(*)により再帰的に定義されたもので、系列集合の表現方法である。

正則表現シミュレーション手法では系列の集合を正則表現を用いて表す。例えば、$01(0+010)=\{010,01010\}$、$0^*=\{\varepsilon,0,00,000\}$、$(01)^3=\{010101\}$、$01^3=0111$、$(0+01)^2=\{00,010,001,0101\}$など正則表現で系列集合を表す。$\varepsilon$は空語である。

2.2 有限オートマトンと論理回路

系列集合を表す正則表現と等価な表現方法として、有限オートマトンがある。有限オートマトンMは、6項組 $(Q, \Sigma, \Gamma, \delta, \lambda, q_0)$ で表される。ここで、

Q	：状態集合
Σ	；入力アルファベット
Γ	；出力アルファベット
$\delta: Q\times\Sigma\rightarrow Q$	；状態遷移関数
$\lambda: Q\times\Sigma\rightarrow\Gamma$	；出力関数
$q_0\ (\in Q)$	；初期状態

である。また、Mの入力系列 $a_1a_2\ldots a_n\ (a_i\in\Sigma)$ に対する動作を、状態$q_i=\delta(q_{i-1},a_i)$を満たす状態の並び$q_0q_1\ldots q_n$で定義する。そしてそのときの出力系列は、各入力に対する出力$o_i=\lambda(q_{i-1},a_i)$の並び$o_1o_2\ldots o_n$で表される。

以下の例では、状態遷移関数および出力関数を(q_1,q_2,a,b) $(\in Q\times Q\times\Sigma\times\Gamma)$ の集合で表すことができる。ただし$\delta(q_1,a)=q_2$かつ$\lambda(q_1,a)=b$とする。ここで、(q_1,q_2,a,b)を状態遷移の枝という。

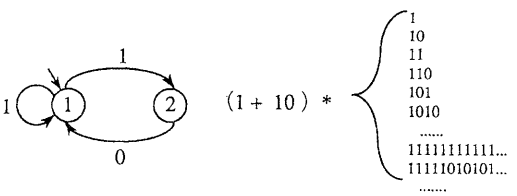


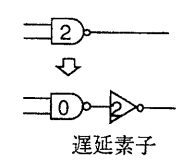
図1 正則表現と有限オートマトン

2.3 論理回路のモデル

論理回路は、回路中の素子の集合V、結線の集合E$(\subseteq V\times V)$、およびVの各要素の入出力関係を表す有限オートマトンの集合{M}とVから{M}への関数により表される。

外部入力とはV中の(w,v)なる結線のない素子vのことをいう。長さkの経路とは、結線の並びで$(v_1,v_2),(v_2,v_3),\ldots,(v_{k-1},v_k)$が含まれるとき$v_1,v_2,\ldots,v_k$のことをいう。閉路とは、$v_1,v_2,\ldots,v_{k-1},v_1$なる経路のことをいう。

論理素子の入出力端子の値として{0,1,X}の３値を使用する。ｎ入力１出力素子の入力出力関数は、$\{0,1,X\}^n$を入力アルファベットとし、{0,1,X}を出力アルファベットとする有限オートマトンで表される[1]。また素子モデルについては、後置モデルを使用する（図２）。これは素子を遅延のない論理素子と遅延素子とに分け、遅延素子を後ろに置いて表したものである。遅延のない論理素子や種々の遅延素子は有限オートマトンで表す。



遅延素子

図２　後置モデル

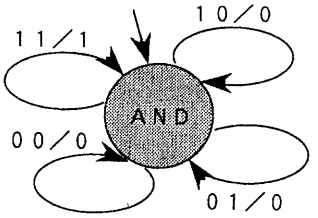


図３　ＡＮＤを表すオートマトン

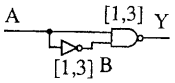
３　系列集合を用いた取扱い

3.1最小・最大遅延モデル

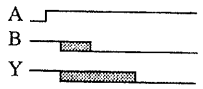
実際の論理素子は、製造条件や使用環境により素子の遅延時間にばらつきがでてくる。このため、タイミングに関する仕様がきびしい回路では、タイミング誤りを起こし易くなり良い検証方法が必要になってくる。このような素子の遅延のばらつきを考慮したモデルが最小・最大モデルである。このモデルでは、遅延時間を最大値（最大遅延時間）と最小値（最小遅延時間）で指定し、実際にはその中のどの値になるかわからないとしている。以下、MINが最小遅延時間、MAXが最大遅延時間として素子の遅延時間を[MIN,MAX]と表す。

従来、最小・最大遅延モデルを最小遅延時間から最大遅延時間までの期間の出力を不定値として扱うことが多かった。例えば、図４(a)の回路では立ち上がり信号を付加するとBの素子の遅延によりYにはパルスを発生するが、図４(b)の様に不定値の部分に吸収されている。さらに不定値の部分が伝搬し拡大していることがわかる。このように不定値を用いた方式では、実際の回路よりも悲観的なシミュレーション結果になる。一方、図４(c)のように最小遅延時間から最大遅延時間のすべての遅延の値の場合を考えることもできる。

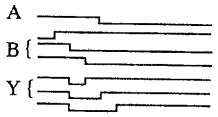
このとき、すべての遅延の値の場合を考えるので複数の時系列が現れる。この方式では、不定値の拡大は生じないので、正確なタイミング検証が行える。回路中の素子に現れる時系列の集合を扱うために、ここでは正則表現論理シミュレーション手法[1]を用いる。この手法は、入力信号系列の系列集合に対して素子の動作のシミュレーションを行うことができる方法である。



(a) シミュレーションする回路



(b) 従来のシミュレーション結果



(c)本手法のシミュレーション結果

図４　最小最大遅延

3.2　遅延を表すオートマトン

ここでは、[m,n]の最小・最大遅延（m<n）を表すオートマトンを示す。

最小・最大遅延については２通りのモデルが考えられる。一つはあいまいな最小・最大遅延モデルで、連続した入力の変化に対していつも[m,n]のすべての場合を尽くすものである。他方は、連続した入力の変化に対して[m,n]の間のある決まった値で処理を行うものである。

あいまいな最小・最大遅延は、状態集合$Q=\{0,1,X\}^n$、入力集合$\Sigma=\{0,1,X\}$、出力集合$\Gamma=\{0,1,X\}$、状態遷移関数$\delta_{mn}: Q\times\Sigma\rightarrow 2^Q$を、

$$\delta_{mn}((a_n a_{n-1} a_{n-2}\cdots a_1),a)=\{(a_{n-1}a_{n-2}\cdots a_1 a),\ (a_{n-2}a_{n-2}a_{n-3}\cdots a_1 a),\ \cdots\ (a_{m-1}\cdots a_{m-1}a_{m-2}a_{m-3}\cdots a_2 a_1 a)\}$$

出力関数$\lambda_{mn}: Q\times\Sigma\rightarrow\Gamma$を、

$$\lambda_{mn}((a_n a_{n-1} a_{n-2}\cdots a_1),a)=a_n$$

として表すことができる。

例）

[3,5]のあいまいな最小・最大遅延は、

入力 a　状態 $(a_5a_4a_3a_2a_1)$、

に対する遷移と出力は、

遷移 $\delta'((a_5a_4a_3a_2a_1),a)=\{(a_4a_3a_2a_1a),$ 5遅延
$(a_3a_3a_2a_1a),$ 4遅延
$(a_2a_2a_2a_1a)\}$ 3遅延

出力 $\lambda'((a_5a_4a_3a_2a_1),a)=a_5$

となる。

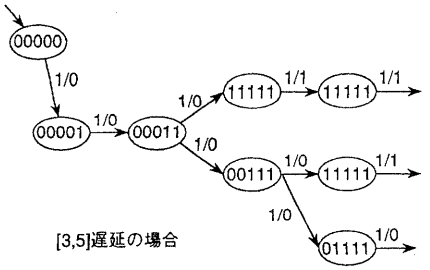


図５　[3,5]遅延の場合

一方、あいまいでない最小・最大遅延は、[m,m],[m+1, m+1],...,[n,n]の遅延のオートマトンの直和で、全体の初期状態から非決定的に各オートマトンへ遷移させたものとなる。ここで[i,i]の遅延オートマトンは、状態集合$Q=\{0,1,X\}^i$、入力集合$\Sigma=\{0, 1, X\}$、出力集合$\Gamma=\{0,1,X\}$、状態遷移関数$\delta_i: Q\times\Sigma\rightarrow Q$を$\delta_i((a_ia_{i-1}a_{i-2}\ldots a_1),a)=(a_{i-1}a_{i-2}\ldots a_1a)$、出力関数$\lambda_i: Q\times\Sigma\rightarrow\Gamma$を$\lambda((a_ia_{i-1}a_{i-2}\ldots a_1),a)=a_i$としたものである。

例えば、[10,10]の遅延素子は、過去10単位時刻の入力を記憶していて10単位時刻前の値を出力するので、状態集合として$\{0,1,X\}^{10}$を用い、$(a,a_i\in\{0,1,X\})$

入力 a、状態 $(a_{10}a_9a_8\ldots a_1)$、
に対する遷移と出力は、

遷移 $\delta((a_{10},a_9a_8\ldots a_1),a)=(a_9a_8\ldots a_1a)$、

出力 $\lambda((a_{10}a_9a_8\ldots a_1),a)=a_{10}$

となる。

4　シミュレーションアルゴリズムリズム

4.1　組み合わせ回路のシミュレーション

ここでは、組み合わせ回路のシミュレーション手法について述べる。本シミュレーションにより求められるのは、外部入力を入力系列とし、各端子の出力を出力アルファベットとするオートマトンである。組み合わせ回路の場合、各素子の出力は外部入力のみにより決まるので、外部入力に近い素子から順次、出力系列集合を求めて行けばよい。なお、閉路のある回路は内部を切断して組み合わせ回路として扱われる[1]。

例えば、図６(a)のような組み合わせ回路の場合を考える。図の様な入力に対し端子aに{0011,0111}、端子bには{0001,0011}という出力系列の集合が表れたとする。このとき、問題となるのが、素子の要素間に相関関係がある場合である。端子dのNANDゲートの遅延がなかったときの出力は、２通り×２通りの４通りの場合が考えられる。しかし、この回路は端子eで分岐しているためそれぞれの要素間に相関関係があることになる。この相関関係を表すために図６(b)のように各時刻での外部入力値をそのまま使用している。

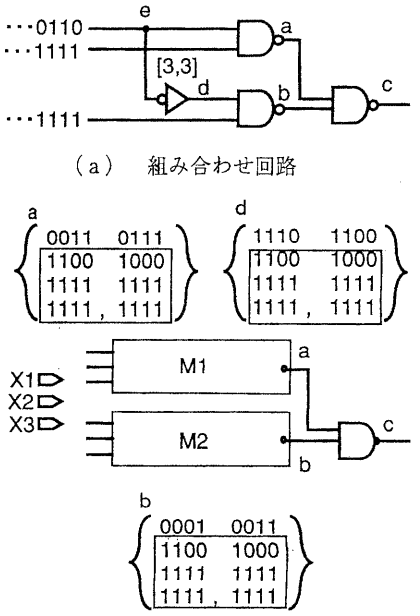


図６　シミュレーション手法

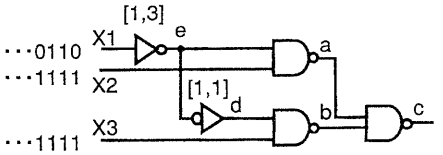


図７　最小・最大遅延を用いたときのシミュレーション回路例

次に図７のように最小・最大遅延[1,3]を使用した場合、端子eには１遅延、２遅延、３遅延の３通りの出力{0110,0011, 0001}が現れる。端子aでは、{0110,0011, 0001}の３通り、端子bも同様に{0011,0001,0000}の３通りになる。このままでは端子cには３×３の９通り場合が考えられるが、先と同じように端子eで分岐しているため、要素間に相関関係がある。よってそれぞれの系列集合に分岐点情報をとして、分岐点eでの出力値を付加する必要がある。これは、新たにeも外部入力と同様に扱い、４入力の回路と見ることに等しい。

4.2 論理回路シミュレーションアルゴリズム

論理回路シミュレーションは基本的に2つの有限オートマトンM_1、M_2を入力とする。そしてこのM_1とM_2の直積をとり、できあがった直積オートマトンM_{12}を出力とする。M_1とM_2からM_{12}を構成するアルゴリズムについて説明する。

4.2.1 論理演算の処理

論理演算した結果生成されるオートマトンの構成方法ついて述べる。

$\times_{para}$演算：$M_{12}=M_1\times_{para}M_2$

ここで、入力される2つのオートマトンを、$M_1=\{(p_1,p_2,I_1,o_1)\}$,$M_2=\{(q_1,q_2,I_2,o_2)\}$とする。ここで、$I_1,I_2$:外部入力、$o_1,o_2$:はオートマトンM1、M2の出力値である。また、$M_{12}$は$\times_{para}$演算により構成されたオートマトンである。

1) 空のキュー(FIFO)を用意する。

2) M_1の初期状態p_0とM_2の初期状態q_0を取り出し順序対(p_0,q_0)をキューに登録する。(p_0,q_0)を状態対と呼ぶことにする。

3) キューから状態対(p,q)を取り出す。キューが空なら終了する。

4) M_1の状態がp、M_2の状態がqであるすべての状態遷移の枝(p,p_n,I_1,o_1)、(q,q_n,I_2,o_2)の組み合わせに対し、I_1とI_2が等しい場合のみ以下の処理をする。

5) M_{12}の状態遷移の枝として$((p,q),(p_n,q_n),I_1,(o_1,o_2))$を$M_{12}$の状態遷移の枝の集合に付加する。

6) (p_n,q_n)がこれまでに現れていなければキューに登録する。3)へ。

$\times_{para}$で構成されたオートマトンの出力の対(o_1,o_2)のo_1とo_2にAND,OR,EXOR,NAND,NOR,EXNORなどのの演算を施せば、2つの正則集合を入力をした時の論理演算結果の正則集合となる。

外部入力の系列集合が決定性有限オートマトンとして表される場合、外部入力とその他の端子の出力との$\times_{para}$演算は以下の様に簡単化できる。

$M_1=\{(p_i,q_i',i_{i'1}a_{I'2}\cdots I_{i'k},I_{i'n})\}$、

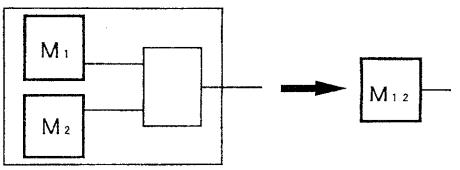
$M_2=\{(p_j,q_j',a_{j'1}I_{j'2}\cdots I_{j'k},o_{j'n})\}$とする。ここでi,j,k,n=1,2,3,…、Iは外部入力値の並び、oは出力値、p,qは状態を表す。$I_{i'n}$は外部入力の一つである。ここで$M_{12}=M_1\times_{para}M_2$としたときの$M_{12}$は、$M_{12}=\{(p_i,r_j),(q_i',s_j'),I_{j1'}I_{j'2}\cdots I_{j'k},(I_{j'n},o_{j'n})\}$であるが、$M_{12}=\{p_j,q_j',I_{j'1}I_{j'2}\cdots I_{j'k},(I_{j'n},o_{j'n})\}$と簡単化できる。

すなわちM_1の状態がp_i、M_2の状態がq_jである様な全ての枝の組み合わせについて考える必要がなくM_{12}の枝の外部入力の一つと出力値の順序対$(I_{j'n},o_{j'n})$を作ればM_{12}を構成できる。

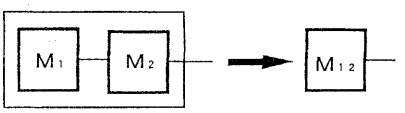
上の性質を利用し、出力値を外部入力の一つとする操作を以下の様に定義する。

$M=\{p_i,p_i',I,o\}$としたとき$M_r=\{p_i,q_i',I\cdot o,\varepsilon\}$とする。ここで・は連接を表す。

決定性オートマトンで表される系列の正則集合を扱うとき、この操作を行い、先の様に外部入力と、ある端子の出力とでオートマトン$\times_{para}$を構成するときにシミュレーションの高速化を図ることができる。



（a） X_{para}構成



（b） X_{cas}構成

図8 直積オートマトンの構成

4.2.2 遅延の処理

遅延の処理を行った結果生成されるオートマトンの構成方法について述べる。

$\times_{cas}$演算：$M_{12}=M_1\times_{cas}M_2$

1) 空のキュー(FIFO)を用意する。

2) M_1の初期状態p_0とM_2の初期状態q_0を取り出し状態対(p_0,q_0)をキューに登録する。

3) キューから状態対(p,q)を取り出す。キューが空なら終了する。

4) M_1の状態がp、M_2の状態がqであるすべての状態遷移の枝(p,p_n,I_1,o_1)、(q,q_n,I_2,o_2)の組み合わせに対しo_1とI_2が等しい場合のみ以下の処理をする。

5) M_{12}の状態遷移の枝として$((p,q),(p_n,q_n),I_1,o_2))$を$M_{12}$の状態遷移の枝の集合に付加する。

6) (p_n,q_n)がこれまでに現れていなければキューに登録する。3)へ。

4.3　シミュレーションの流れ

図7の回路のシミュレーションの処理を示す。

1)　外部入力を出力とする様なオートマトンオートマトンM_{x1}、M_{x2}、M_{x3}を構成する。外部入力は、出力のないオートマトンとして与えられるので、それぞれの入力端子に入力される値を出力値とすればよい。

2)　端子eでの出力を表すオートマトンM_eは、まず、M_{x1}の出力値にNOT演算を行い、オートマトンM'_{x1}を構成する。次に、M'_{x1}と[1,3]遅延を表すオートマトン$M_{[1,3]}$に$\times_{cas}$演算を行い、M_eを構成する。すなわち、$M_e = M'_{x1} \times_{cas} M_{[1,3]}$である。

3)　同様にして、$M_d = M'_e \times_{cas} M_{[1,1]}$より$M_d$を構成する。

4)　M_eとM_{x2}の$\times_{para}$演算を行い、オートマトンを構成する。このとき出力の順序対の要素にNAND演算を行いM_aとする。

$M_a = (M_e \times_{para} M_{x2})_{NAND}$と記述することにする。

5)　同様に、$M_b = (M_d \times_{para} M_{x3})_{NAND}$により$M_b$を構成する。

6)　同様に、$M_c = (M_a \times_{para} M_b)_{NAND}$により$M_c$を構成する。

以上の様に各端子での出力を表すオートマトンが構成されて行く。

5　最小最大遅延を用いたタイミング検証

以下では、ハザード検出と非同期回路の検証について述べる。

5.1ハザード検出

ハザード検出は、許容されるすべての入力変化について出力パルスが発生するかどうかを判定する問題である。すべての入力パターンを$P_1, P_2, P_3, \cdots, P_m$とすると、$(P_1+P_2+P_3+\cdots+P_m)(P_1^*+P_2^*+P_3^*+\cdots+P_m^*)$に対するシミュレーションと、その結果得られるオートマトンの解析結果からハザードが検出できる。ここでは、単一の入力変化に対して、出力が2回以上変化したとき、ハザードであると考える。入力信号が長さ1単位時間の後、変化するのは、遅延素子の初期値を指定できることによる。以下に、出力されたオートマトンからハザード検出のためのオートマトンの変形方法について述べる。

出力されたオートマトンの状態qに入る枝の集合を$E_{in}=\{(q_{in}, q, I_{in}, O_{in})\}$、状態qからでる枝の集合を$E_{out}=\{(q, q_{out}, I_{out}, O_{out})\}$とする。ただし、Iは外部入力、O(={0,1,X})は出力値とする、。このとき、枝の集合の直積$E_{in} \times E_{out}$をとる。出力値の変化が問題であるから、でき上がったすべての要素$((q_{in}, q, I_{in}, O_{in}),\ (q, q_{out}, I_{out}, O_{out}))$を$(O_{in}, q_{in\ out}, O_{out})$という3項組で表す。出力されたオートマトンを初期状態から順に上の3項組で表していく。これは、図9のようにオートマトンを変形することになる。

上の操作で変形されたオートマトンは、初期状態を除いて、状態から出る枝が1つの場合からなる。よって初期状態から枝をたどり次状態への枝がないかループするまで遷移の前後の出力値（O_{in}とO_{out}）の変化を調べ、変化が2回以上あればハザード検出されたことになる。

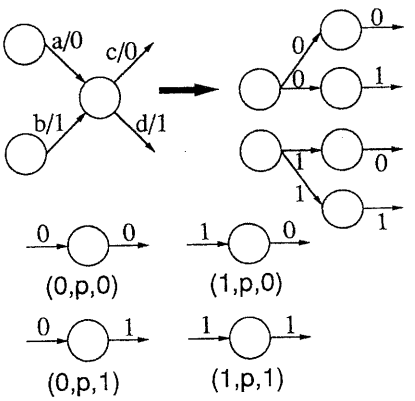


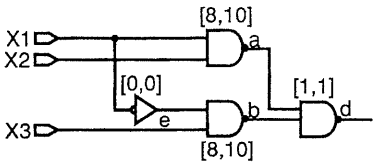
図9 オートマトンの変形

例として図10(a)の回路において、入力が(1,1,1)から(0,1,1)に変化した時、出力は関数的には1のままである。しかし、遅延の違いにより図10(c)のようにハザードが生じることが本方法で検出できる。外部入力に$(1,1,1)^{20}(0,1,1)^*$の入力系列が入ったとき、図のゲートaでは[8,10]の遅延をもつため、NAND演算した結果、図10(c)のタイミングチャートのように遅延8、9、10に対応して3通りの出力がでる。bについても3通りの出力がでる。dでは、$1^{20}(1^*+01^*+001^*+101^*)$という系列集合を出力し、ハザードが生じていることが分かる。一方、、図10(b)の回路は図10(a)にゲートを一つ付加したハザードを生じない回路である。実際、

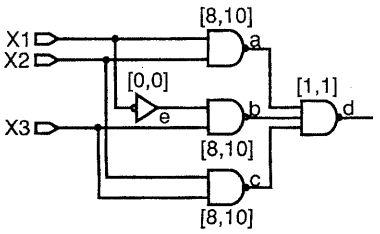
$\{(0,0,0)^{20} + (0,0,1)^{20} + \ldots + (1,1,1)^{20}\}$

$\{(0,0,0)^* + (0,0,1)^* + \ldots + (1,1,1)^*\}$

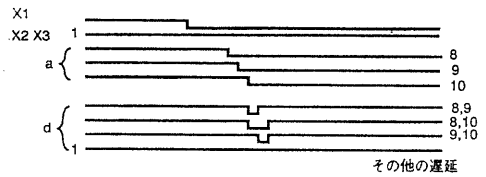
という入力に関してシミュレーションを行い、ハザードが発生しないことを確認した。



(a)



(b)



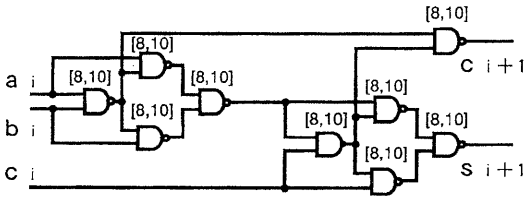
(c) タイミングチャート

図10　ハザード検出

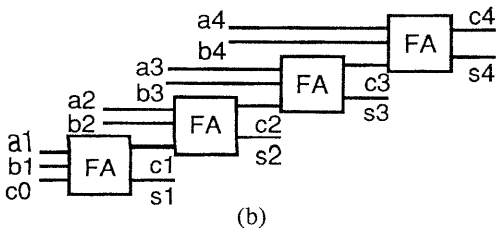
5.2　4ビット加算器のハザード検出

5.2.1　順次桁上げ加算器

現在図11(a)の様な全加算器に対してシミュレーションを行った。入力(a,b,c)として
(1,0,X)(0,1,X)*、(0,0,X)*(1,1,X)*、
(0,1,X)(1,0,X)*、(1,1,X)*(0,0,X)*、
{(0,0,0)+(1,1,0)}{(1,0,1)*+(0,1,1)*}、
{(1,0,1)+(0,1,1)}{(0,0,0)*+(1,1,0)*}
{(0,0,1)+(1,1,1)}{(1,0,0)*+(0,1,0)*}、
{(1,0,0)+(0,1,0)}{(0,0,1)*+(1,1,1)*}
についてハザードがでる。この全加算器を4段従属接続して4ビット順次桁上げ加算器を構成するが、すべての素子のばらつきを考えているので、状態数は最悪の場合、$((2^{10})^9)^4$ ($\fallingdotseq 10^{36}$)状態になるが、単一ハザード検出のための入力パターンについてシミュレーションを行った結果、4ビット目を表す有限オートマトンの状態数が130826状態になった。



(a) 全加算器(FA)



(b)

図11　4ビット全加算器

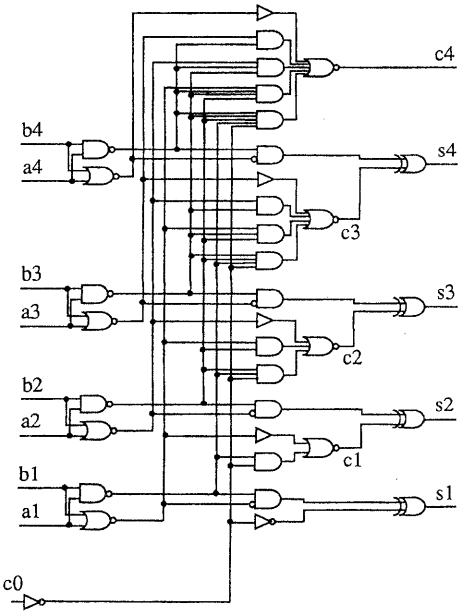


図12　先見回路付き4ビット全加算器

5.2.2 桁上げ先見加算器

また、図12のような桁上げ先見（4ビット）加算器(TTL 74SN283)についてもシミュレーションを行っている。ゲートの遅延はすべて[8,10]である。同じ構成方法でに従う2ビットの桁上げ先見加算器のハザード検出を、全入力パターン（状態数34状態96）枝に対し、SPARC-LC AS1000（東芝）上で行ったところ、71秒で終了し、その時の最終的なキャリー(c_2)の出力状態数は6万状態程度になった。

5.3非同期回路の検証について

回路にループが含まれる場合、ループを適当な所で切断して、シミュレーションの繰り返しにより、切断点の動作を求める。ここでは、例として図13のJKフリップ-フロップの検証を行った。この場合、回路の出力点Qを切断しており、Qの値は仕様として与えられるので、切断点の入力に仕様の値を与えシミュレーションした結果と与えた値を比較することにより、検証を行える。一部、仕様で0でも1でもない値としてある部分については、シミュレーションの繰り返し（前回にシミュレーションで得られたQの値を入力としてもう一度シミュレーションを行う。）により求めている。

ゲートD_1とD_2の遅延の値を変えて、J=K=Preset(PRS)=1かつ幅10のクロックパルスを入れた時の検証を行った。D_1、D_2の遅延が共に[4,5]とし他のゲートの遅延は[1,1]とした。このとき図14のような結果になった。正しくQの値が反転していることが分かる。

一方、遅延が[3,5]の時には、図14のように発振する場合があり、正しく動作しないことがわかる。図15は連続する状態遷移を示したもので、各行は(現状態)（次状態）(入力値)（遅延情報)(出力)を表す。図には、クロック(CK)の立ち下がり(状態55)後の3つの場合を示している。各場合はD_2の遅延が3、4、5の場合に対応している。なお、[4,5]の場合に比較し、[3,5]の場合には状態数が非常に大きくなった。

以上の例題では、あいまいな最小最大遅延を用いたが、あいまいでないモデルを用いても同様な結果が得られる。

6．おわりに

ここでは、最小遅延時間から最大遅延時間の期間のすべての遅延の場合について考えた最小最大遅延モデルを用いたシミュレーションについて述べた。またハザード検出およびフリップフロップなどの非同期回路の検証を行い、本手法の有効生性を示した。

本手法は最大遅延時間から最小遅延時間までのすべての場合を考えるので出力系列のパターンが指数級数的に増える。しかし、多くの場合は本手法で扱えるのではないかと考えられる。

以上の例題では、あいまいな最小最大遅延を用いたが、あいまいでないモデルを用いても同様な結果が得られる。

今後、データの圧縮法について研究を行い、より大規模な回路に対しても適用して行きたい。

謝辞　日頃から御討論頂く太田有三助教授はじめ、羽根田研究室の皆様に感謝します。

参考文献
[1] 木村，羽根田，矢島：”正則表現論理シミュレーション手法に基づく非同期順序回路の検証”，電子情報通信学会論文誌(D)，Vol.J71-D,No.9,pp.1786-1796(Sep.1988).
[2] 石浦，高橋，矢島：”論理回路の正確なタイミング検証のための時間記号シミュレーション”，情報処理学会論文誌，Vol.31,No.12,pp.1832-1839(Dec.1990).
[3] 濱口，平石，矢島：”二分決定グラフによる順序機械の設計検証について”，９１’夏のＬＡシンポジウム，１９９１．
[4] Ｊ．ホップクロフト，Ｊ．ウルマン共著，野崎，高橋，町田，山崎共訳：”オートマトン 言語理論 計算論Ｉ”，サイエンス社（昭和59）．
[5] Harry R.Lewis：”A Logic of Concrete Time Intervals”，Proc. of LICS '90, pp.380-389 (Jun.1990).

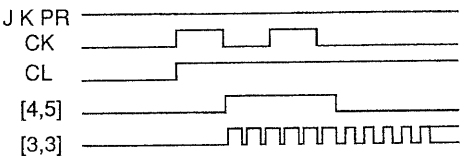
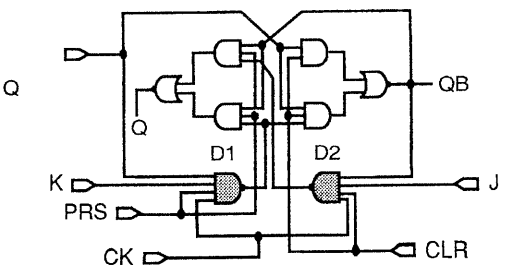


図１４　ＪＫ-ＦＦのタイミングチャート



D1,D2に最小最大遅延を適用する.
その他の素子は遅延[1,1]とした.

図１３　ＪＫ-ＦＦ

##	s1 --> s2	J	K	Clk	Prs	Clr	D1	Qb	D2	Q
52	53	1	1	1	1	1	1	0	0	
53	54	1	1	1	1	1	1	0	0	
54	55	1	1	1	1	1	1	0	0	
55	56	1	1	0	1	1	1	0	0	
56	58	1	1	0	1	1	1	0	0	
58	61	1	1	0	1	1	1	0	1	
61	67	1	1	0	1	1	1	1	1	
67	76	1	1	0	1	1	1	0	1	
76	82	1	1	0	1	1	1	0	1	0
82	88	1	1	0	1	1	1	0	1	1
88	94	1	1	0	1	1	1	1	1	1
94	100	1	1	0	1	1	1	0	1	1
100	106	1	1	1	1	1	1	0	1	0
106	112	1	1	1	1	1	1	0	1	1
112	118	1	1	1	1	1	1	1	1	1
118	124	1	1	1	1	1	1	0	1	1
124	136	1	1	1	1	1	1	0	1	0
136	154	1	1	1	1	1	0	0	0	1
154	166	1	1	1	1	1	0	1	1	1
166	181	1	1	1	1	1	1	0	1	1
181	193	1	1	1	1	1	0	0	1	0

##	s1 --> s2	J	K	Clk	Prs	Clr	D1	Qb	D2	Q
55	57	1	1	0	1	1	1	1	0	0
57	59	1	1	0	1	1	1	1	0	0
59	64	1	1	0	1	1	1	1	0	1
64	71	1	1	0	1	1	1	1	0	1
71	79	1	1	0	1	1	1	1	0	1
79	85	1	1	0	1	1	1	1	0	1
85	91	1	1	0	1	1	1	1	0	1
91	97	1	1	0	1	1	1	1	0	1
97	103	1	1	0	1	1	1	1	0	1
103	109	1	1	0	1	1	1	1	0	1
109	115	1	1	1	1	1	1	1	0	1
115	121	1	1	1	1	1	1	1	0	1
121	130	1	1	1	1	1	1	1	0	1
130	146	1	1	1	1	1	1	0	0	1
146	161	1	1	1	1	1	1	0	0	1
161	175	1	1	1	1	1	1	0	0	1
175	188	1	1	1	1	1	1	0	0	1
188	202	1	1	1	1	1	1	0	0	1
202	218	1	1	1	1	1	1	0	1	1
218	235	1	1	1	1	1	1	0	0	1

##	s1 --> s2	J	K	Clk	Prs	Clr	D1	Qb	D2	Q
55	57	1	1	0	1	1	1	1	0	0
57	59	1	1	0	1	1	1	1	0	0
59	64	1	1	0	1	1	1	1	0	1
64	72	1	1	0	1	1	1	1	0	1
72	79	1	1	0	1	1	1	1	0	1
79	85	1	1	0	1	1	1	1	0	1

図１５　[3,5]の遅延の時のQの出力