

ペトリネットによる回路のモデル化と解析・ 検証手法の提案

渡辺, 隆司 / Dang, Ryo / Watanabe, Takashi / 檀, 良

(出版者 / Publisher)

法政大学工学部

(雑誌名 / Journal or Publication Title)

法政大学工学部研究集報 / 法政大学工学部研究集報

(巻 / Volume)

37

(開始ページ / Start Page)

5

(終了ページ / End Page)

10

(発行年 / Year)

2001-03

(URL)

<https://doi.org/10.15002/00003785>

ペトリネットによる回路のモデル化と解析・検証手法の提案

A METHOD FOR MODELING, ANALYZING AND VERIFYING A CIRCUIT USING PETRINET

渡辺 隆司* 檀 良**
Takashi WATANABE and Ryo DANG

At present, FSM(Finite State Machine) is used for modeling a circuit in Formal-Verification but FSM can not be used to express a parallel operation. To overcome this difficulty, we propose a method for modeling a circuit using Petrinet. Petrinet is the graph which can express a parallel operation and an asynchronous operation naturally. Examples are shown to prove the usefulness of our method. We also propose a method for analyzing and verifying using Petrinet properties.

Key Words : Formal-Verification, Petrinet, Model-Checking

1. はじめに

現在, 半導体素子の微細化により設計を行う回路は大規模化・複雑化してきている。それに伴い, 設計にかかる期間も長期化している。回路設計において, 検証にかかる期間が大部分を占めているのが現状である。そこで, Formal-Verification という技術が近年注目を集めている。Formal-Verification とは, 設計を行った回路をモデル化し, そのモデル化を行ったものに対し数学的操作を行うことにより証明し回路検証を行う技術である。数学的証明であるので, 従来の論理シミュレーションに比べ非常に高速かつエラーの見落とし無しに検証を行うことが可能である。

従来設計を行った回路のモデル化には FSM(Finite State Machine)が用いられているが, FSM はその解析を容易にするため厳しい制約がある。その制約上並列動作を表現することが不可能である。そこで, 本研究では並列動作・非同期動作を自然に表現することの出来るペトリネットを回路のモデル化に用いる。そして, Verilog-HDL によって設計された回路をペトリネットとしてモデル化する手法を提案する。

2. Formal-Verification

Formal-Verification とは, 設計を行った電子回路を組み合わせ論理回路については BDD(Binary Decision Diagram), 順序論理回路に関しては FSM(Finite State Machine)によりモデル化を行い, そのモデル化されたも

のに数学的操作を加え, 数学的証明を行うことにより検証を行う技術である。Formal-Verification により検証を行うことで, 従来のシミュレーションに比べ高速にまた質の高い検証が可能となるという利点がある。また, Formal-Verification には以下のような種類がある。

1) 等価性検証

等価性検証とは, 2 つの回路が同一の論理を有しているかどうかを検証するものである。

2) モデルチェッキング

モデルチェッキングとは, 設計を行った回路がある性質(仕様)を満たしているかどうかを検証する。

3) Theorem Proof

Theorem Proof とは, 設計を行った回路が仕様と一致しているかどうかを検証する。

本論では, 特にモデルチェッキングに着目する。

3. ペトリネット

本研究で回路のモデル化に用いたペトリネットについて述べる。ペトリネットとは, システムの動作を, 各時点で成り立つ“条件”とそれに基づいて生起し得る“事象”の因果関係によって特徴付けるモデルである。以下にペトリネットの定義, 図 1 にペトリネットの例を示す。

<定義>

ペトリネットとは次に示す 4 項の組である。

$$N=[P,T,A,M]$$

ここで

P : プレースの有限集合

*大学院電気工学専攻

**電子情報学科

T: トランジションの有限集合
 A: 有向枝の有限集合
 M: 初期マーキング
 である。

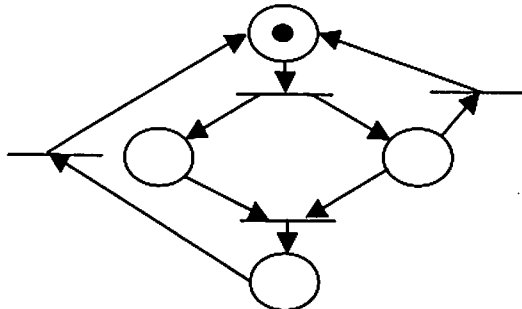


図1 ペトリネット

図1に示すように、ペトリネットは円と棒と有向枝で表現される。円はプレース(Place)と呼ばれ、システムの局所的な条件・状態を表し、FSM おけるステートにあたる。また、棒はトランジション(Transition)と呼ばれ、一定の条件が満たされたときに生じ得る事象を示し、FSM における遷移条件にあたる。また、プレース内に置かれている黒丸は、そのプレースに対応する条件が成立していることを表し、現状態を示す。初期マーキングとは、システムの初期状態のことである。マーキングは、トークンの配置を示し、マーキング変化の規則に従い変化する。このマーキングの変化によりシステム全体の遷移の様子を表現する。

4. ペトリネットによるモデル化手法

現在、回路のモデル化には FSM(Finite State Machine) が用いられている。しかし、FSM は解析を容易にするため非常に厳しい制約がある。その制約により並列動作を表現することが不可能である。しかし、設計を行う回路内の並列動作は増加している。そこで、本研究では Verilog-HDL により設計された並列動作を含む回路を 1 つのペトリネットとしてモデル化する手法を提案する。図2に本手法のフローを示す。

まず、設計を行った Verilog-HDL 記述の回路からそれぞれのプロセスの状態遷移表を作成する。この処理は、現在公開されているツールにより行うことができる。次の処理として、プレースの割り当てという処理を行う。これは状態遷移表におけるステートにそれぞれ対応するプレースを割り当てる。また、状態遷移表の遷移条件を、トランジション

として考え初期状態を初期マーキングにする。この次に遷移行列を作成する。遷移行列とは、各トランジションの入力プレース・出力プレースの関係を行列にした

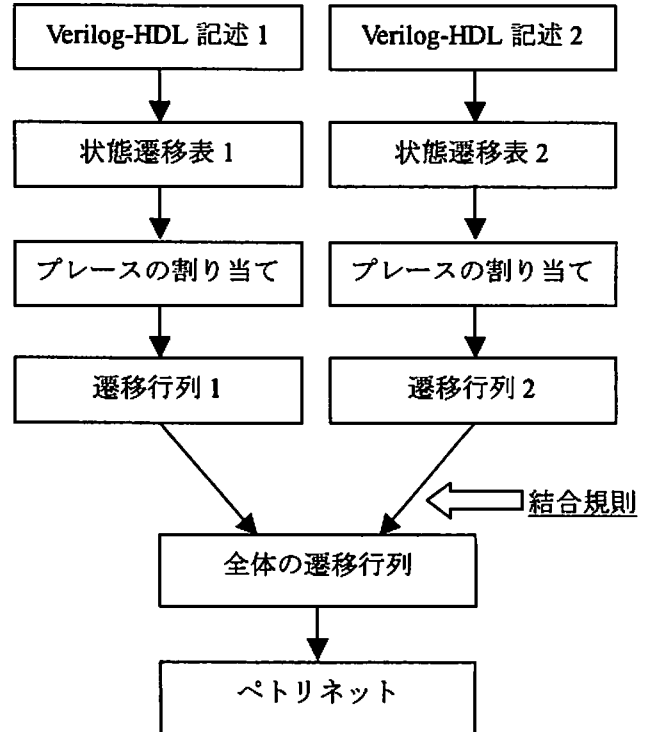


図2 モデル化のフロー

ものである。この遷移行列に変換した時点で FSM からペトリネットへ変換されたこととなる。次に各プロセスの遷移行列をシステム全体の遷移行列として結合を行う。この結合には、本研究にて検討を行った結合規則に従い結合を行う。この結合規則については後述する。結合後の遷移行列が最終的な結果となる。このシステム全体の遷移行列からペトリネットを作図することが出来る。

<結合規則>

- (1) 遷移条件(トランジション)ごとに入力プレースが属する FSM とは異なる出力プレースへの遷移を追加。
- (2) 複数の FSM に共通の遷移条件(トランジション)が存在しない場合、初期プレースを新たに追加。初期プレースから FSM での初期状態への遷移条件(トランジション)を追加。トークンを初期プレースへ配置。

5. 対象回路

本手法を検証するために用いた回路を示す。4 つの回路に対して行った。それぞれタイプが違うのでそれぞれの FSM を示す。

<回路 1>

表 1 FSM1

現状態	遷移条件	次状態
Sa	C2	Sb
Sb	b	Sc
Sc	c,d	Sd
Sd	C1	Sa

表 2 FSM2

現状態	遷移条件	次状態
Se	C2	Sf
Sf	C1	Sg
Sg	f	Se

この回路は、2つの module で構成され、互いに並列動作を行う回路である。

<回路 2>

表 3 FSM1

現状態	遷移条件	次状態
Sa	t11	Sb
Sb	t12	Sa

表 4 FSM2

現状態	遷移条件	次状態
Sc	t22	Sd
Sd	t21	Sc

この回路は、並列動作ではないが並列動作を行わない回路についてもモデル化できる必要があるので行う。

<回路 3><回路 4>

回路 3,4 に関しては、回路が大規模になれば回路内に存在するプロセスの数が増加する。よって、2つのプロセスに対してだけモデル化できても有用ではない。そこで回路 1 に2つのタイプの FSM を追加することにより3つのプロセスの存在する回路を作成した。これらの回路がモデル化可能であれば、これ以上のプロセスが存在する場合についてもモデル化可能であることが確認できる。ここでは、回路 1 に追加する FSM のみを示す。

表 5 FSM3-3

現状態	遷移条件	次状態
Sh	f	Si
Si	g	Sh

表 6 FSM4-3

現状態	遷移条件	次状態
Sh	g,h	Si
Si	j	Sj
Sj	C2	Sk
Sk	C1	Sh

これらの回路を用いて実験を行う。

6. 実験結果

本手法により前述の対象回路のモデル化を行った結果を示す。ただし、詳細な結果までの流れは回路 1 についてのみ示す。

<回路 1>

まずはじめに、FSM1,2 それぞれのステートにプレースの割り当てを行う。この回路では、Sa~Sg の7つのステートがありそれぞれにプレース P1~P7 を割り当てる。また遷移条件はそのままトランジションとして考える。

この処理の後、各トランジションの入力プレース・出力プレースの関係を行列にすることにより遷移行列を作成する。作成した遷移行列を表 7,8 に示す。

表 7 回路 1 の遷移行列 1

入力/出力	1	3	4	5
①		C2		
3			b	
4				c,d
5	C1			

表 8 回路 1 の遷移行列 2

入力/出力	2	6	7
②		C2	
6			C1
7	f		

表 7,8 の遷移行列で①,②と円で囲まれたプレースは初期プレースを示し、FSM での初期状態に対応するプレースである。ここで、前述した結合規則の 1 に従い全体の遷移行列に結合する。結合後の遷移行列を表 9 に示す。またシステム全体の遷移行列より各トランジションの入力プレース・出力プレースが分かる。これよりペトリネットを作図することが出来る。回路 1 のペトリネットを図 4 に示す。

表 9 システム全体の遷移行列

入力/出力	1	2	3	4	5	6	7
①			C2			C2	
②			C2			C2	
3				b			
4					a, d		
5	C1						C1
6	C1						C1
7		f					

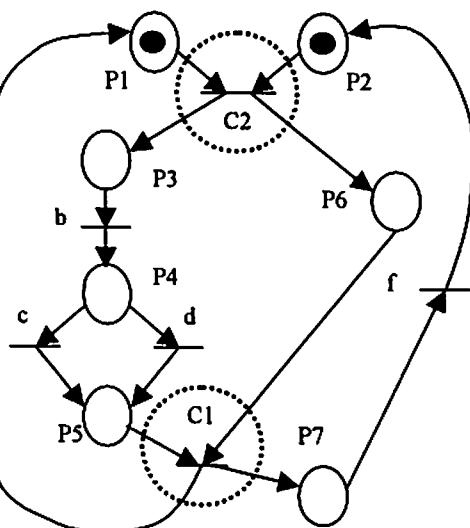


図 3 回路 1 のペトリネット

表 8, 図 4 において点線の円で囲まれたトランジションは, 結合時に結合規則により 2 つの入力プレースと出力プレースを持つトランジションに変換されたものである。FSM が並列動作を表現できない原因は, このようなトランジションが複数の入力プレースと出力プレースを持つことが許されないためである。本研究において提案した手法により並列動作を含む回路を 1 つのペトリネットとしてモデル化できることが確認できた。

回路 2 に関しては, 結合規則 2 に従い, 図 4 における点線の円で囲まれた部分が追加され, 初期プレース P1 から条件を満たした側に遷移していく様子がわかる。

<回路 2>

表 9 システム全体の遷移行列

入力/出力	1	2	3	4	5
①		t'11		t'21	
2			t11		
3		t12			
4					t22
5				t21	

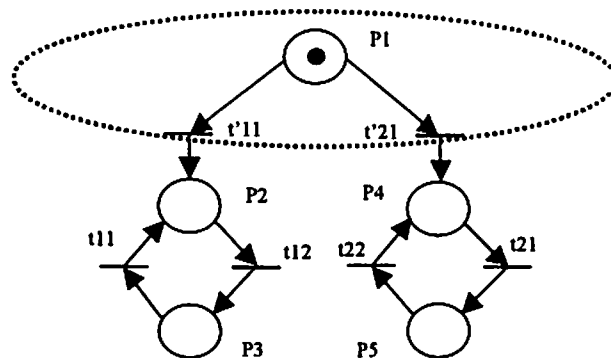


図 4 回路 2 のペトリネット

<回路 3>

表 10 システム全体の遷移行列

入力/出力	1	2	3	4	5	6	7	8	9
①			C2			C2			
②			C2			C2			
3				b					
4					a, d				
5	C1						C1		
6	C1						C1		
7		f							f
⑧		f							f
9								g	

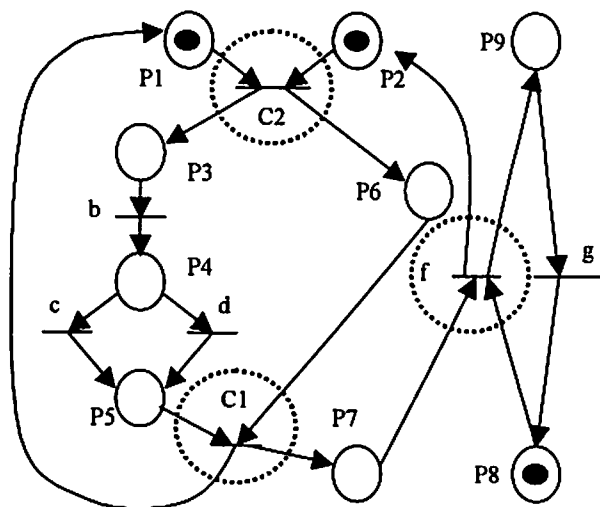


図 5 回路 3 のペトリネット

回路 3 に対して, 表 10, 図 6 にて点線の円で囲まれた 3 つのトランジションが 2 つの入力プレース・出力プレースを持ち, 3 つの並列動作を行う回路に対してもモデル化可能であることが確認できる。

<回路 4>

表 11 システム全体の遷移行列

入力/ 出力	1	2	3	4	5	6	7	8	9	10	11
①			C2			C2					C2
②			C2			C2					C2
3				b							
4					c,d						
5	C1						C1	C1			
6	C1						C1	C1			
7		f									
8									g,h		
9										j	
⑩			C2			C2					C2
11	C1						C1	C1			

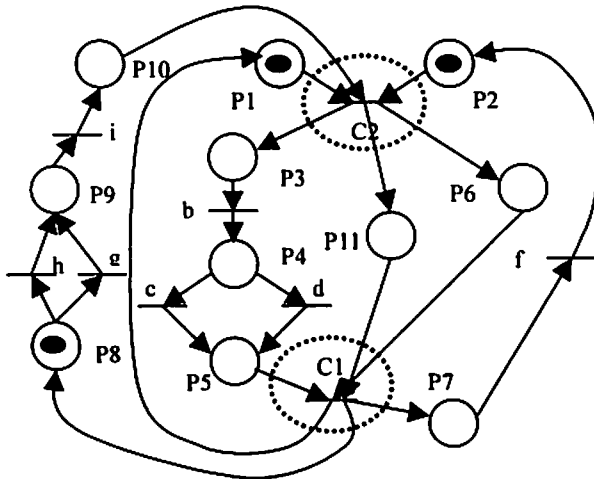


図 6 回路 4 のペトリネット

回路 4 に関して、表 11 では網掛け、図 6 で点線の円で囲まれているトランジション C1, C2 が 3 つの入力プレース・出力プレースを持っていることが分かる。よって、このようなタイプの回路についてもモデル化可能であることが確認できた。

7. ペトリネットの解析

ペトリネットの解析を行う際には、マーキングと呼ばれるものが重要となる。マーキングとは各プレースに配置されているトークンの数を示すものであり、このマーキングの変化によりシステムの遷移を表現する。以下にマーキングの例を示す。

プレース数 7, プレース 2, 5 にそれぞれトークンが 1 つ配置されている場合

マーキング → [0, 1, 0, 0, 1, 1, 0]

図 7 マーキングの例

マーキングは、以下に示す規則に従い変化する。

- (1) 任意のトランジション t に関し、トランジション t の入力プレースすべてにトークンが 1 つ以上存在するとき、 t は発火可能(enabled)である。発火とは、ペトリネットにおいて状態の遷移のことを示す。
- (2) トランジション t は発火可能であるときのみ発火しようとする
- (3) 一時に発火するトランジションは 1 つのみである。
- (4) トランジション t が発火すると、マーキング $M=(M(p_1), M(p_2), \dots, M(p_m))$ は次に定められる $M'=(M'(p_1), M'(p_2), \dots, M'(p_m))$ に変化する。 $i=1, 2, \dots, m$ に対して

- ① 任意のプレース p がトランジションの出力プレースでありかつ入力プレースでない時、
 $M'(p)=M(p)+1$
- ② 任意のプレース p がトランジションの入力プレースでありかつ出力プレースでない時、
 $M'(p)=M(p)-1$
- ③ その他の時、 $M'(p)=M(p)$

以上の規則をふまへ、到達可能木と呼ばれるものを作成することにより解析を行う。到達可能木とはマーキングに対応する節点とトランジションに対応する枝から構成されるものであり、以下に到達可能木を作成するフローを示す。

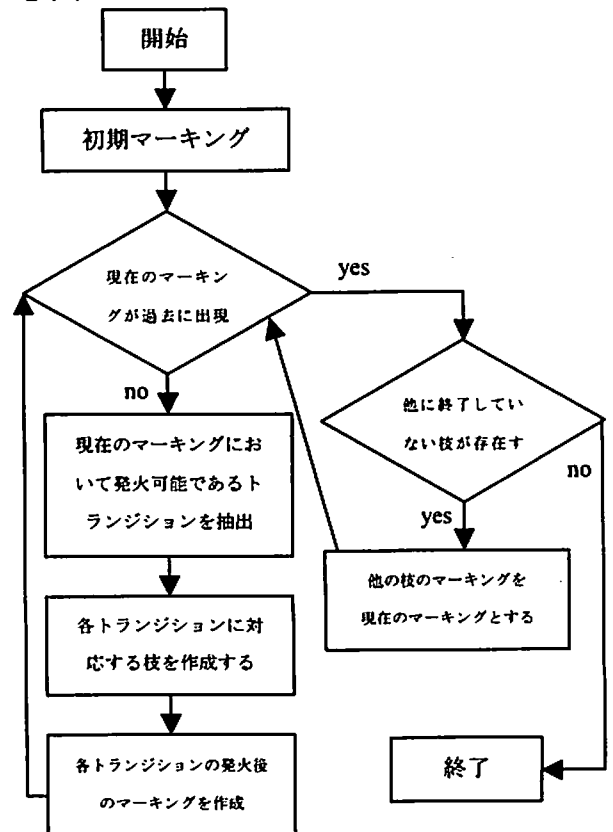


図 8 到達可能木作成フロー

以上のフローに従い、前述の対象回路の回路 1 から作成した到達可能木を示す。

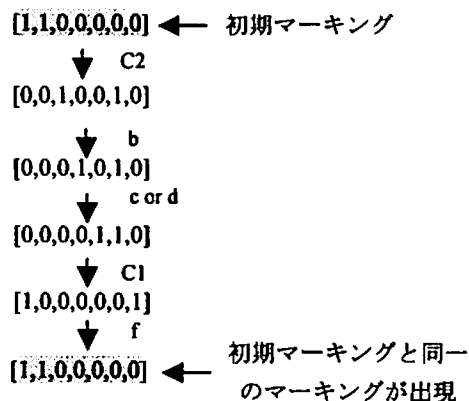


図 9 回路 1 の到達可能木

図 9 は、回路 1 の解析結果である到達可能木である。初期マーキングである $[1,1,0,0,0,0,0]$ よりマーキング変化の規則に従い変化していき、6 番目のマーキングが初期マーキングと同一のものが出現し、終了となる。矢印の横には遷移時に発火するトランジションが示されている。今回は、回路 1 のみの解析結果のみを示したが、他の回路に関しても到達可能木の作成が可能であり、解析可能であることが確認できた。

8. ペトリネットの検証

前節にて作成した到達可能木を基に、ペトリネットの持つ性質を用いることにより回路の検証を行う。まず、ペトリネットの持つ性質について述べる。

<有界性(k-bounded)>

ペトリネットが k -有界であるとは、任意のプレース p にトークンが k 以上存在することがないことである。ペトリネットの有界性が成立することにより、プレースを実現するハードウェア(スタック・レジスタや FIFO レジスタ)の記憶容量が有界であることが証明される。

<安全性(safe)>

安全性とは、プレース p が 1-有界であることをいい、これは各プレースに複数のトークンが存在することは決しないことである。安全性が成立することにより、例としてレジスタに書き込まれたデータが読み出される前に新しいデータが書き込まれてしまうようなことが決して起こらないことを保証する。

<生きている(live)>

ペトリネットが生きているとは、ペトリネット内のすべてのトランジションが決して発火不能に陥ることがない

ことをいう。ペトリネットが生きているということは、並列処理の問題であるデッドロックと関係する。システムのデッドロックとはすべてのトランジションが発火不能に陥る状況であり、ペトリネットが生きていれば、デッドロックが決して生じないことを保証する。

以上の性質を用い検証を行う。図 9 に示した回路 1 の到達可能木よりペトリネットの性質を用い検証を行うと以下になる。

- ・任意のプレースにおいて、置かれるトークンの数はすべて 1 であることから、回路 1 のペトリネットは有界性を持つことが確認できる。

- ・前述のことから回路 1 のペトリネットは 1-有界であることも確認できる。よって、安全性も持つことが確認できる

- ・初期マーキングから始まり、6 番目に初期マーキングと同一のマーキングが出現するまで到達可能木が途切れることがない。これは、このペトリネットにおいてすべてのトランジションが発火不可能に陥ることが決してないことを示し、この並列動作においてデッドロックが決して生じないことが保証される。

9. まとめ

本研究にて提案した手法により、並列動作を含む回路をペトリネットによりモデル化可能であるか 4 つのタイプの対象回路を用いて実験を行った。その結果、4 つすべて本手法により正しくモデル化可能であることが確認できた。この 4 種類のタイプの回路がモデル化可能であることより、本手法によってどのような回路であってもモデル化可能であると考えられる。

また、並列動作を含む回路をモデル化したペトリネットを到達可能木を作成することにより解析する手法を示し、ペトリネットの持つ性質、有界性・安全性・生きている、を到達可能木から対象のペトリネットが持ち、デッドロックなどが生じないことを保証することにより検証を行う手法を示した。しかし、本研究にて提案した検証手法についてはペトリネットを完全に検証することは不可能であり、現在、FSM において行われている CTL(Computation Tree Logic)により記述された仕様との比較等がペトリネットに対しても可能となることが望まれ、今後の課題である。

参考文献

- 1) 当麻,内藤,南谷,“順序機械”,岩波講座 情報科学 13,pp244-265,1983
- 2) Xiaohan Zhu, Bill Lin,“Hardware Compilation for FPGA-based Configurable Computing Machine”, AC—M/IEEE Design Automation Conference, 1999