

# 高水準ペトリネットを記述可能な援用ツール HiPS2 と 非同期回路検証への適用

## HiPS2 : An Application Tool for High-level Petri Net and Its Verification Case-study on Asynchronous Circuits

堀内 維作 †      和崎 克己 ††  
Isaku Horiuchi    Katsumi Wasaki

### 1 はじめに

デジタルシステムの信頼性確保は、システム全体の機能性を安定して維持するために必要不可欠である。デジタルシステムの多くは非同期回路を利用しており、特に統一されたクロック同期を必要としない非同期回路は、クロック・スキューの問題が発生せず、遅延時間を最小化できる利点がある。しかし、閉路上にラッチを有しない非同期ループが存在する場合、系全体の安定性はゲート素子の遅延時間に依存する欠点も有する [1]。非同期回路の安全性の検証は一般的に難しいとされている。非同期回路の安定性を検証する手段として、信号点の状態変化に着目したトレース理論を用いた方法などが提案されている。他方、信号点の状態変化 (H → L, L → H などのエッジトリガー) をメッセージとして後段のゲートへ順次送る、ペトリネットでのモデル化がある [2]。ペトリネットとは、並行的、非同期的、分散的なシステムを表現するためのグラフィカルな数学的モデルであり、それに時間の概念を加えたものが時間ペトリネットである [3, 4]。現実の論理ゲートには遅延時間が存在するため、時間の概念を付与した時間ペトリネットで非同期回路の振る舞いを検査する手法は有用である [2]。

筆者らは、基本的な時間 Place/Transition ネット (時間 P/T ネット) を階層的に記述し、シミュレーションやネット構造全体のインバリヤント、また可達性解析などを実行可能な援用ツール HiPS を開発し、種々の並列システムモデルに適用してきた [5]。しかしながら、基本的な時間 P/T ネットで規模の大きなシステムをモデル化する際、通常の P/T ネットでは同型な構造を有する部分ネットが数多く生成・配置する必要があるため、ネットが複雑かつ大規模になり、また記述性、可読性が低下してしまう。この問題を解決する手段として、トークンが情報を保持することができるカラーペトリネットがある。本報告では、高水準ペトリネット、具体的には制約付カラーペトリネットの記述に対応した拡張援用ツール HiPS2 の概要について述べる。トークンに定義可能なデータ型は、整数型、文字列型、ビット列型ならびに構造体である。拡張ツールの有用性を確認するため、非同期回路検証への適用を試みた。

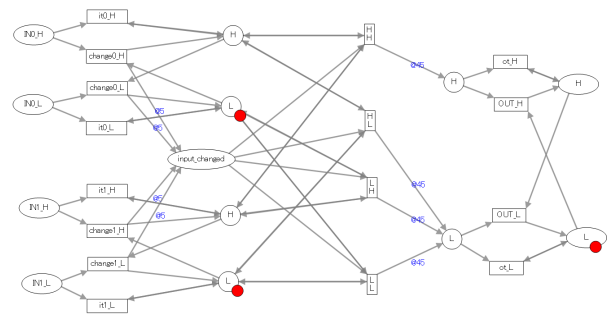


図1 2入力論理積ゲートのペトリネットモデル [5]

### 2 ペトリネット

ペトリネットとは、Carl Adam Petri によって提唱された、複数のプロセスからなる離散事象システムをグラフィカルに表現するモデル化手法である [3]。並行的、非同期的、分散的、並列的、非決定的、確率的な動作を特徴とするシステムの、記述・検証ツールとして有用である。システム構造の可視的な表現手段としての利用だけでなく、ペトリネット内の各プレースにトークンを配置し、事象生起に伴うトークン分布の遷移により、システムの振る舞いをシミュレートできる。

#### 2.1 時間 P/T ネット

時間 P/T ネットとは、上記で示したペトリネットに、時間の概念を付与したものである。時間に関するパラメータの導入の仕方により、大きく 2 つに分類される。一方は、(1) プレースに入力されたトークンが利用可能になるまでの遅延時間を導入する場合であり、プレース時間ペトリネットと呼ばれる。もう一方は、(2) トランジションの発火に要する時間を定めるものであり、トランジション時間ペトリネットと呼ばれ、内部では更にいくつかの手法が提案されている。本報告では、トランジションの発火に要する時間を定めたトランジション時間ペトリネットを時間 P/T ネットと呼ぶことにする。

#### 2.2 カラーペトリネット

カラーペトリネットとは、トークンに色 (情報) を属性として付与し、データ型、データ値、式を定義できる高水準ペトリネットである [4, 6]。プログラミング言語のような構文と意味論をネットに融合できるため、トークンフロー制御や発火制御、計算処理を行うことができる。これにより、同型な構造を有する部分ネットの情報を畳み込むことができ、モデルの規模を縮小できる。

† 信州大学大学院工学系研究科,  
Graduate School of Science and Technology, Shinshu University.  
†† 信州大学工学部情報工学科,  
Department of Information Engineering, Faculty of Engineering,  
Shinshu University.

### 3 ペトリネット援用ツール HiPS

HiPSは、既存のペトリネットツールの記述性、操作性、再利用性の問題を解決するために本学で開発されたペトリネット援用ツールである[5]。図2に、HiPSの編集画面を示す。直感的で一般的な操作方法のGUIと、ペトリネットの階層化並びに時間ペトリネットに対応し、作成したペトリネットモデルをシミュレートし、挙動を観察することが可能である。また、T-インバリエント解析や可達解析などの解析機能が実装されており、現在も解析機能の追加が行われている。本ツールは著者らの研究グループによって、主に非同期回路設計・検証などに応用されている[5, 7]。

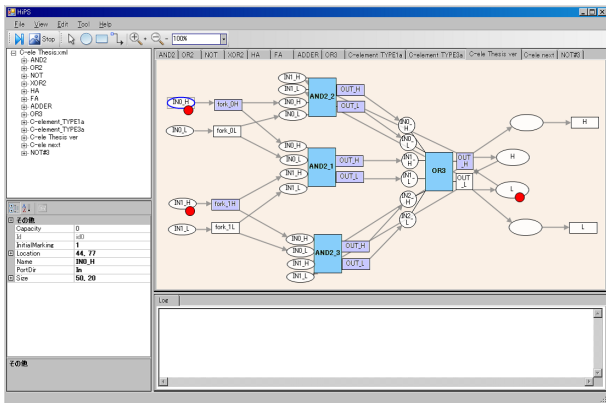


図2 HiPSの編集画面

### 4 制約付きカラーペトリネット援用ツール HiPS2

HiPS2は、HiPSをカラーペトリネットの記述に対応させた拡張援用ツールである。HiPSは上記で示した通り、階層化に対応しており、作成済みモデルのモジュール化により、設計における負担の軽減を図っている。データ値の表現を行うと、モデル構造が複雑かつ大規模になってしまう。そのため、トークンにデータ値を付与し、各種演算をネットに融合させることで、抽象度の高い設計を可能とするカラーペトリネットへの機能拡張をHiPSに行った。以下に、定義可能なデータ型とトランジション発火論理条件式について説明する。

#### 4.1 データ型

本ツールで定義可能なデータ型を表1に示す。各種データ型を用意することで、システムに適したデータ値の表現が可能となり、ネット構造の簡略化を実現する。

整数型	- Int8, Int16, Int32
文字列型	- String
ビット列型	- BitArray
論理型	- Boolean
列挙型	- Enum
構造体	- Struct

表1 宣言可能なデータ型

構造体は、一つ、または複数のデータ値をまとめて格納できるデータ型である。構造体の利点は、ユーザが各システムに適したデータ型を定義できる点とシステムの情報をトークンに多く付与することにより、さらにネット規模が縮小できる点である。データ型宣言ダイアログを図3に示す。このダイアログで変数宣言、変数名の変更、削除などデータ型の管理を行う。宣言された変数は、アーク、トランジション発火条件論理式で利用される。

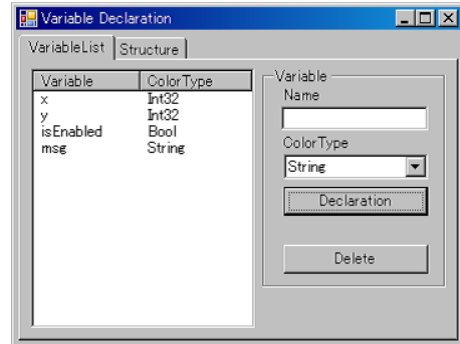


図3 データ宣言ダイアログ

#### 4.2 トランジション発火論理条件式

トランジション発火論理条件式は、条件式や計算式をトランジションに付与する式である。この条件式により、トランジションが発火可能か判定する発火制御、また、発火可能であれば、条件によって異なる処理をトークンのデータ値を行うトークンフロー制御が可能となる。具体的には、入力プレースの保持するトークンのデータ値をアークにラベル付けされた変数に代入し、トランジション発火論理条件式で処理を行う。入力アークの変数またはトランジション発火論理条件式で処理された変数が、出力アークにラベル付けされていればその変数のデータ値を有するトークンが出力される。存在しない場合、トークンは出力されない。

#### 条件式

記述可能な条件式は、以下の通りである。

if-else 文
switch-case 文

表2 トランジション発火論理式に記述可能な条件式

#### 演算子

記述可能な演算子は、以下の通りである。

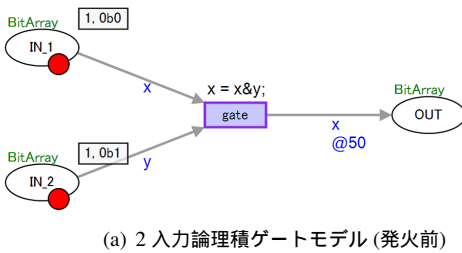
#### 4.3 記述例

図4(a)はHiPS2で記述した2入力論理積ゲートモデルである。プレース上部に表示されている“BitArray”は、プレースが保持できるトークンのデータ型であり、プレース右上部は“トークン数、データ値”を表している。アークにラベル付けされた $x, y$ はデータ宣言されたBitArray型変数であり、トランジションの出力アークにラベル付けされた“@50”は遅延時間を表している。また、トランジション(gate)上部に表示されている $x = x \& y;$ は、トランジション発火条件論理式であり、出力アークにラベル付けされた変数を用いて、実現し

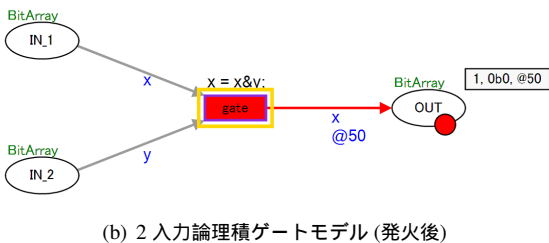
算術演算子	+ , - , * , / , %
関係演算子	== , != , < , > , <= , >=
	eq , ne
論理演算子	! ,    , &&
ビット演算子	^ ,   , ~ , &
ビットシフト演算子	<< , >>
複合代入演算子	+= , -= , *= , /= , %= ,  =
	^= , <<= , >>=
三項演算子	<条件式> ? <真式> : <偽式>

表3 トランジション発火論理式に記述可能な演算子

たい処理を記述する。初期マーキングとして、プレース IN\_1 はデータ値“0b0”，つまり High のトークンを1つ、IN\_2 はデータ値“0b1”，つまり Low のトークンを1つ保持している。発火後のモデルを図4(b)に示す。IN\_1 のデータ値“0b0”がアークラベル  $x$  に代入され、IN\_2 のデータ値“0b1”がアークラベル  $y$  に代入され、トランジション発火論理条件式  $x = x \& y$ ; が実行される。プレース OUT にデータ値“0b0”のトークンが出力され、論理積が実行された。



(a) 2入力論理積ゲートモデル (発火前)



(b) 2入力論理積ゲートモデル (発火後)

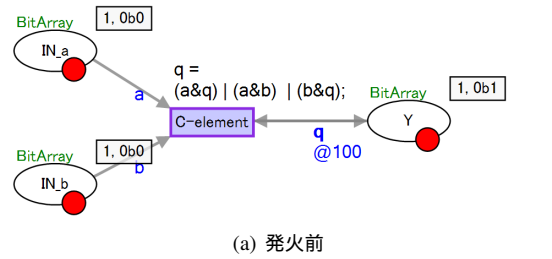
図4 HiPS2 で記述した2入力論理積ゲートモデル

## 5 非同期回路のモデル化と検証

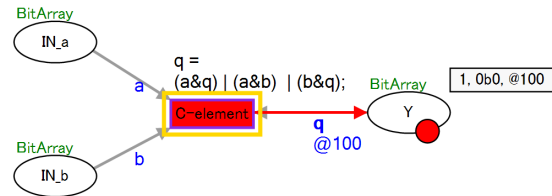
制約付きカラーペトリネット援用ツール HiPS2 の有用性を確認するため、非同期回路のモデル化と検証を試みた。

### 5.1 Muller's C-element のモデル化

Muller's C-element は、非同期回路の構成要素の一つで、記憶素子の役割を果たす、AND2 ゲートを3つと OR3 で構成される2入力1出力の論理回路である。2つの入力値が同じであればその値を出力し、異なる場合の出力は変化しないという、ラッチの振る舞いをする。C-element のカラーペトリネットモデルを図5に示す。HiPS2 ではトランジション発火条件論理式で構成要素



(a) 発火前



(b) 発火後

図5 C-element のカラーペトリネットモデル

を表現できるので、ネット構造のみで表現する時間 P/T ネット [5] に比べて、大幅な簡略化が実現できる。このように HiPS2 では抽象度の選択に幅を持たせることができ、設計者は、検証したい事象に適した抽象度でモデルを作成できる。

### 5.2 2bit データレジスタのモデル化と検証

FIFO メモリの構成要素として使用する 2bit データレジスタの回路図ならびにカラーペトリネットモデルを、それぞれ図6、図7、図8に示す。C-element1, 2, 3, および XOR1, 2 はデータフロー制御部であり、D-FF1, 2 はレジスタである。制御部の XOR ゲートから出力される制御信号が“0b1”であるとき、D-FF は Full であり、“0b0”であるとき D-FF は Empty である。今回のモデルにおける初期状態は、すべての出力を“0b0”，つまり Low とし、レジスタは Empty，つまりデータ数ゼロとする。まず、C-element1 の入力 IN\_A1 が“0b0”から“0b1”に立ち上がると、IN\_B1 への入力は“0b0”の否定入力“0b1”であり、C-element1 の出力 Q1 は“0b1”となる。C-element2 の出力 Q2 は“0b0”であるので、図7(a)のように、XOR1 は“0b1”に立ち上がり、制御信号を D-FF1 へ伝播する。すると、D-FF1 は Full になり、データ“0b1”を保持する。次段の C-element2 の入力 IN\_A2 には、“0b1”が入力され、前段の C-element1 と同じ動作を行い、XOR2 を“0b1”へ立ち上げ、D-FF2 は D-FF1 の出力を保持する。ここで、XOR1 の入力 Q1, Q2 とともに“0b1”になるので、D-FF1 は Empty となる。次に C-element3 の出力 Q3 が“0b1”になるので、XOR2 の入力 Q2, Q3 とともに“0b1”になるので、D-FF2 は Empty となる。また、C-element1 の入力 IN\_A1 が“0b1”から“0b0”立ち下がると、C-element1 から逐次立ち下がっていく、制御信号“0b1”を D-FF へ伝播する。立ち上がり時と同じ動作をし、レジスタはデータを保持する。HiPS2 による 2bit データレジスタのモデル化とシミュレーションによって、時間 P/T ネットに比べ容易に記述でき、FIFO 動作をより視覚的、直感的に確認できた。



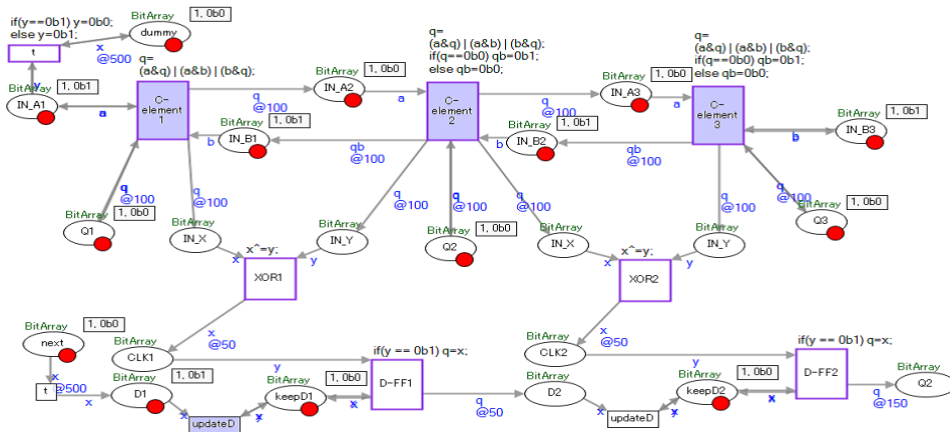
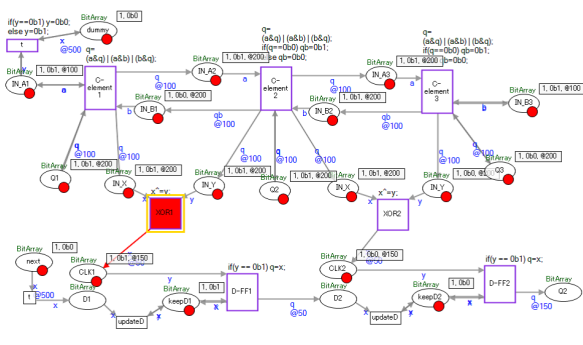
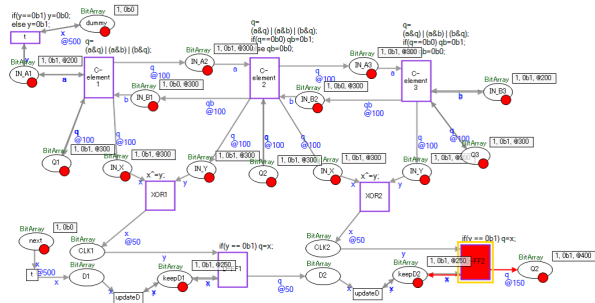


図6 2bit データレジスタのカラーペトリネットモデル



(a) 制御部から制御信号“0b1”が出力された様子



(b) D-FF2 からデータが出力された様子

図7 2bit データレジスタモデルのシミュレーションの様子

## 6 まとめと今後の課題

時間 P/T ネット援用ツール HiPS からカラーペトリネットを記述可能な援用ツール HiPS2 へと拡張し、非同期回路のモデル化と検証を行った。モデル表現の抽象度が高く、ネット規模の縮小とモデル設計の簡略化により、記述性、可読性が向上したことを確認した。

今後は、試作した引用ツールをベースとし、階層型時間ネットを用いて、比較的大規模な実回路のモデル化と検証を目標とする。設計・検証済の機能コンポーネントをライブラリとして援用ツールへ登録し、より規模の大きな回路システムの構成定義を援用ツール内で完結する。また、設計・検証済の回路モデルから、実装系への接続を検討する。具体的には、HiPS2 で定義したデータ型を、例えば形式仕様記述言語 LOTOS の抽象データ型 (ADT) として利用可能とし、ネット定義を LOTOS のブ

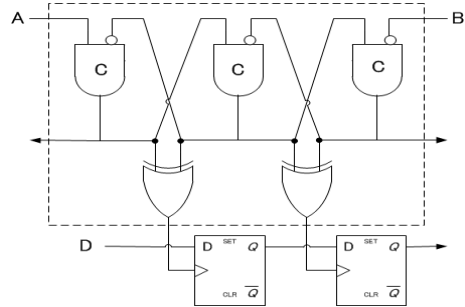


図8 2bit データレジスタの回路図

ロセス定義へ帰着させるコード生成器を準備する。その後、LOTOS 処理系から得られたプロセストレーシンググラフを解析するモデル検査を実施する。以上の流れで動的解析を行うことで、上流設計に対するシステムの挙動や特徴の抽出などの一貫した検査系の構築を目指す。

謝辞 本研究の一部は科学研究費(23500174)の助成を受けたものである。

## 参考文献

- [1] J.A.Brzozowski, C.J.H.Seger, “Asynchronous Circuits”, Springer-Verlag, 1995.
- [2] T.Yoneda, B.Zhou, B.H.Schlingloff, “Verification of Bounded Delay Asynchronous Circuits with Timed Traces”, Proc. of Int’l Conf. on Algebraic Methodology and Software Technology, Springer, LNCS-1548, pp.59-73, 1999.
- [3] T.Murata, “Petri Nets: Properties, Analysis and Applications”, Proc.IEEE, Vol.77, No.4, pp.541-580, 1989.
- [4] 熊谷, 薦田, “ペトリネットによる離散事象システム論”, コロナ社, 1995.
- [5] 松山, 和崎, “Time-Petri Net を用いた非同期回路のモデル化と階層化設計”, FIT2009 講演論文集, 1, 523-526, 2009.
- [6] K.Jensen, “Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use, Vol.1, Basic Concepts, 2nd ed.”, Springer-Verlag, 1996.
- [7] 堀内, 松山, 和崎, “ペトリネットツール HiPS におけるインバリエント解析機能の回路検証への適用”, IEICE 信越支部大会講演論文集, 39, 2010.