5N−7

時相論理に基づく順序回路検証システムの二分決定グラフを用いた改良

久木元 裕治　田中 英彦

東京大学工学部

1 はじめに

我々は、命題論理レベルのLinear Time Temporal Logic (LTTL)で仕様を与え、ディジタルシステムの制御部同期回路を検証するシステム[1]を既に開発しているが、論理表現として積和形(カバー表現)を用いたため、対象となる回路が大きくなると実行速度の点で問題となることがあった。近年コンパクトな論理表現法として注目されている二分決定グラフ(BDD)[2]を用いると、実用的な論理関数の多くが変数の数の多項式オーダの記憶領域で表現でき、論理演算がBDDのノードのサイズに比例した計算時間で実行できることが知られている。また、Branching Time Temporal Logicの一つであるCTLについて、BDDを用いて検証を効率化できることが報告[3]されている。本稿では、[3]で提案されているBDDによる順序回路表現を用いたLTTLに基づく検証システムの改良について述べる。

2 Linear Time Temporal Logic

Linear Time Temporal Logic(LTTL)とは通常の古典論理に以下に示す時相演算子を付け加えた論理である。

$\bigcirc P$	:	次の時刻にPが成り立つ
$\Box P$	:	現在から考えてすべての時刻でPが成り立つ
$\Diamond P$	:	現在から考えていつかはPが成り立つ
PUQ	:	現在から考えてQが成り立つまではPが成立する

これらの時相演算子を用いるとハードウェアの仕様記述に必要なさまざまな性質を記述できる。例えば、信号Pがactiveになると次の時刻に必ず信号Qがactiveになるという仕様は、時相演算子を用いて$\Box(P \rightarrow \bigcirc Q)$と表現することができる。

またLTTLでは公理より次の性質が成り立つ。$\Diamond$演算子の展開規則の中にある$\{P\}$はeventualityと呼ばれるもので、いつかはPが成立するという条件を表す。

$$\begin{aligned} \Box P &= P \wedge \bigcirc \Box P \\ \Diamond P &= P \vee (\sim P \wedge \bigcirc \Diamond P\{P\}) \\ PUQ &= Q \vee (P \wedge \sim Q \wedge \bigcirc (PUQ)) \end{aligned}$$

この性質を用いると、LTTLの論理式を現在に関する条件と次の時刻に関する条件に分解することによって、状態遷移表現を得ることができる。LTTLの時相論理式が充足可能であることを示すためには、この展開された状態遷移図上で無限長の状態遷移シーケンスが存在することをいえばよい。

以上で述べたことを用いると、設計された順序回路を表現する時相論理式(*Design*)、仕様の時相論理式(*Spec*)が与えられたときに、$Design \rightarrow Spec$が恒真であることを示せば設計が仕様に対して正しいことが検証できる。よって、背理法を用いて、この式の否定$Design \wedge \sim Spec$が充足できないことを示せばよいことになる。

3 検証アルゴリズム

仕様はLTTLで与え、その否定の論理式を状態遷移図の形に展開する。このとき各遷移条件をBDDで表現する。[4]

一方、設計はゲート回路で与える。各フリップフロップの出力を$O_1, O_2, \ldots, O_n$、回路の入力信号を$I_1, I_2, \ldots, I_m$とすると、ゲート間の接続情報を用いて、以下に示すような論理関数f_iが求まる。

$$\begin{aligned} \bigcirc O_1 &= f_1(I_1, \ldots, I_m, O_1, \ldots, O_n) \\ \bigcirc O_2 &= f_2(I_1, \ldots, I_m, O_1, \ldots, O_n) \\ &\vdots \\ \bigcirc O_n &= f_n(I_1, \ldots, I_m, O_1, \ldots, O_n) \end{aligned}$$

ここで、次のように論理関数$Design$を定義する。

$$Design(I_1, \ldots, I_m, O_1, \ldots, On, \bigcirc O_1, \ldots, \bigcirc O_n) =$$

$$\begin{cases} 1 & \text{if } \prod_{i=1}^{n} (\bigcirc O_i \cdot f_i(\ldots) + \overline{\bigcirc O_i} \cdot \overline{f_i(\ldots)}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

論理関数$Design$は、フリップフロップの出力値、回路の入力信号値、次状態フリップフロップの出力値の組み合わせが設計を満たしているときに限り論理値1をとる関数[3]である。BDDで表現するときには、フリップフロップの出力変数O_iと次の時刻の出力変数$\bigcirc O_i$は別な変数として区別する。

ここで、仕様の否定の状態遷移図からeventualityを満たしている無限長遷移シーケンスをとりだしてきて、次に示す手続きを行なう。

1. フリップフロップの初期状態を表現するBDD、無限長シーケンスの初めの遷移条件を表現するBDD、設計を表現するBDDに対して、AND演算を行なう。得られたBDDは、遷移条件と初期条件を満たしたときに次の時刻でフリップフロップがどのような状態をとり得るかを表現している。

[0] Improvement of Synchronous Circuit Verification System Based on Temporal Logic Using Binary Decision Diagrams
Yuji KUKIMOTO and Hidehiko TANAKA
Faculty of Engineering, The University of Tokyo

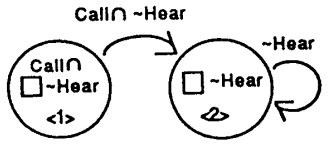

図 1: 仕様の否定の状態遷移図

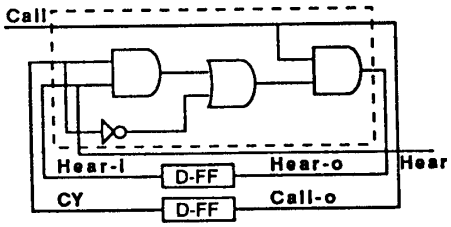

図 2: Reciever 回路

2. 1で得られた BDD から次の時刻のフリップフロップの状態を表現する BDD をとりだす。

3. とりだした BDD、次の遷移条件を表現する BDD、設計を表現する BDD に対して AND 演算を行なう。以下、ループが検出されるまで 2 と 3 を繰り返す。

フリップフロップの次状態を表現する BDD が途中で論理値 0 となったときは、与えられた遷移を設計が許さないことになり、その遷移シーケンスは $Design \wedge \sim Spec$ を満たし得ないことになる。また、得られた BDD が前状態の BDD と一致し、ループが検出されると、その無限長シーケンスは誤設計の例となる。すべての無限長遷移シーケンスについて、設計上で遷移を実現することが不可能であることがわかれば、設計は与えられた仕様を満たしていることが検証できたことになる。

4 検証例

ハンドシェーク・プロトコルを用いてデータ転送を行なうレシーバ回路を例にとって検証の流れを説明する。検証する仕様は $\Box(Call \rightarrow \Diamond Hear)$ で、信号 Call が active になればいつか信号 Hear が active になることを意味する。仕様の否定の状態遷移図と検証対象のゲート回路をそれぞれ図 1、図 2に示した。この遷移図上で考えられる無限長遷移シーケンスは $\{Call \wedge \sim Hear, \sim Hear, \sim Hear, \ldots\}$ である。初期条件としてはフリップフロップがいずれもリセットされていることを仮定する。ゲート間の接続情報より次の関係式が成り立つことを利用して、設計の BDD が得られる。

$$\bigcirc Hear = (\overline{CY} + Hear \cdot CY) \cdot Call$$
$$\bigcirc CY = Call$$

図 3に検証のプロセスを示した。この場合、2 番目の遷移条件 $\sim Hear$ を適用した結果、次状態 BDD が 0 になり仕様を満たしていることがわかる。

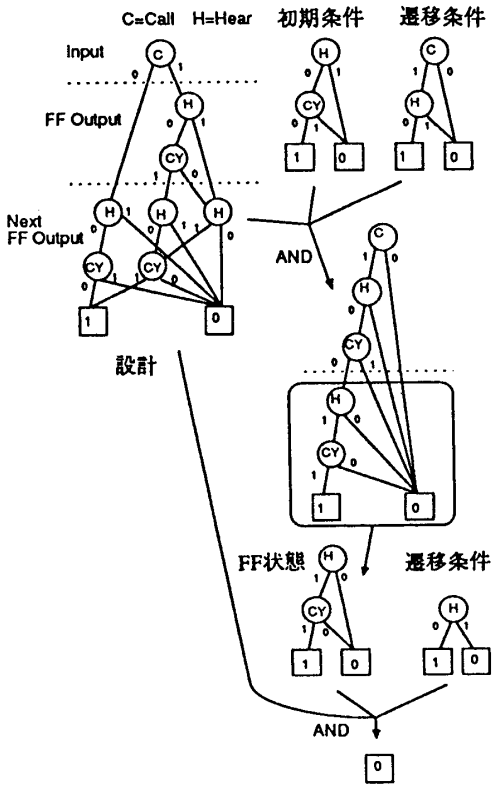

図 3: BDD による検証例

5 おわりに

本稿では 2 分決定グラフを用いた LTTL に基づく順序回路の検証法の改良について述べた。今後はこの検証システムを実装して、実際に回路の検証を行なう予定である。また、仕様を LTTL 以上の表現力をもつオートマトンの形で与えることも検討している。

謝辞 日頃御討論頂く富士通研究所藤田昌宏博士、筑波大学中村宏博士に感謝致します。

参考文献

[1] 中村, 藤田, 河野, 田中: 時相論理に基づく論理回路検証システム , 情報処理学会論文誌 *Vol.30, No.6*, 1989

[2] R.E.Bryant: "Graph-Based Algorithms for Boolean Function Manipulation", *IEEE Trans. Computer, Vol.C-35, No.8*, 1986

[3] J.R.Burch, E.M.Clarke, K.L.McMillan and D.L.Dill: "Sequential Circuit Verification Using Symbolic Model Checking", *Proc. of the 27th Design Automation Conference*, 1990

[4] M.Fujita and H.Fujisawa: "Specification, Verification and Synthesis of Control Circuits with Propositional Temporal Logic", *Proc. of the 9th CHDL*, 1989