記号モデル検査システム「BINGO」の適用事例

Case Study of Applying Symbolic Model Checking System: BINGO

あらまし

論理装置の規模，複雑度が増大するにつれて，論理検証に要する工数，論理検証の網羅性が問題となってきている。従来論理検証の手段としてはシミュレーションが広く用いられてきたが，単にシミュレーションの速度を高めるだけでは論理検証の問題は解決できないと考えられている。

一方，形式的検証技術の進歩により，論理照合，記号モデル検査が実用レベルに達しつつある。これらの技術は，検証項目に対して100%の網羅性を保証できる代わりに，扱える装置の規模が限定される。したがって，形式的検証技術単独に検証すべてをまかせるのではなく，シミュレーションで洗いにくい部分に対して適用するという補完的な形が望ましい。

本稿では，記号モデル検査技術に関して，富士通研究所が行ってきた取組みを記すとともに，著者らが開発した記号モデル検査システム「BINGO」を実際の設計プロジェクトに適用した経験を通じて，記号モデル検査の効果的な利用方法と適用上の課題について述べる。

Abstract

Simulation has been widely used for logic verification of LSI. However, because of the increasing complexity of these devices it is now considered difficult to verify them within a reasonable number of person-hours and at an acceptably high level of coverage.

On the other hand, symbolic model checking is now becoming practical due to progress that has been achieved in formal verification technologies. This technology guarantees 100% coverage for specified verification properties, but restricts the manageable scale of the device. Therefore, symbolic model checking technology should be used as a supplement to verify the parts which a simulation has failed to cover, rather than for verifying an entire LSI.

This paper introduces the work being done by Fujitsu Laboratories Ltd. on symbolic model checking. It then describes a case study of a symbolic model checking system developed by the authors called “ BINGO.”In this study, several techniques for the effective application of symbolic model checking to large designs were used.

中田恒夫（なかた　つねお）

1986年東京大学工学系研究科情報工学専門課程了。同年(株)富士通研究所入社。以来ディジタル装置のCADの研究に従事。1993，1994年カリフォルニア大学バークレー校客員研究員。1998年より情報処理学会設計自動化研究会(現システムLSI設計技術研究会)幹事。工学博士。
コンピュータシステム研究所CAD研究部

岩下洋哲（いわした　ひろあき）

1991年大阪大学大学院工学研究科電子工学専攻修士課程了。同年(株)富士通研究所入社。以来論理検証CAD技術の研究に従事。
コンピュータシステム研究所CAD研究部

高山浩一郎（たかやま こういちろう）

1987年大阪大学大学院工学研究科電子工学専攻修士課程了。同年富士通入社。以来VLSI設計CADの研究開発に従事。1998年から米国富士通研究所。
Advanced CAD Research Department

記号モデル検査システム「BINGO」の適用事例

ま え が き

大規模設計では，ハードウェアを作り出す手間よりも検証する手間の方が大きい。しかも検証に要する手間は設計規模が増大するにつれて急速に増えており，危機的な状況に至っている。

設計検証ではシミュレーションベースの手法が広く用いられてきたが，これを補完し検証の危機に対抗するアプローチとして「形式的検証技術」が注目を集めている。形式的検証とは，設計の正しさを数学的に証明する手法である。現在，組合せ回路の「論理照合技術」と「記号モデル検査技術」が実用レベルに達している。

富士通研究所ではこれらの形式的検証技術の研究開発に取り組んでおり，実用レベルのツールを社内に提供している。本稿では記号モデル検査技術に対する著者らの取組みとともに，実際の設計現場に記号モデル検査ツールを持ち込むことで得られるメリット，解決すべき課題について報告する。

記号モデル検査

記号モデル検査技術[1),2)]は，設計が特定の性質（以下，プロパティ）を満足していることを数学的に証明するために，つぎの手順をとるものである。

(1) 設計モデルを有限状態機械で表現する。
(2) プロパティを時相論理(注)で表現する。
(3) 状態空間を探索して，設計がプロパティを満足するかどうかを調べる。
(4) 有限状態機械を論理式で表現し，二分決定グラフ[3)]（以下，BDD）を用いたアルゴリズムで状態空間探索を実現する。

プロパティを時相論理で表現することから，記号モデル検査は信号間の時間的依存関係を検証するのに適している。記号モデル検査を用いて検証されてきたプロパティの例を列挙する。

・デッドロックが起こらない。
・リクエストが送られてきたら12サイクル以内に応答を返す。
・あるレジスタに値が書き込まれたら，そのデータが使われる前に上書きされることはない。

記号モデル検査のアルゴリズムに関しては，参考文献(1)，(2)に譲り，ここでは重要な点を指摘するに留める。記号モデル検査では起こり得るすべての事象を調べてこれらのプロパティが成り立つことを証明する。これに対し，シミュレーションベース検証では事象を人手で，あるいは乱数で作らねばならない。一般に起こり得る事象の数は非常に多く，人間では列挙しきれないし，逆に人間が見落とすような事象こそが設計誤りの原因となっている。起こり得る事象を網羅的にチェックできる点が記号モデル検査の特徴である。

いわば記号モデル検査は全数パターンチェックを効率的に行う手法と見ることもできる。一方，計算コストはシミュレーションと比較するとはるかに大きくなる。現時点で，記号モデル検査を適用できる装置の規模はレジスタ数で300程度が限界である。さらに，設計によってはレジスタ数30程度でも扱えない場合があり，扱えるか扱えないかが事前に推測できないという問題は依然として解決されていない。また，基本的に記号モデル検査の計算量はレジスタ数に対して指数的に増大し，アルゴリズム上の改善で扱える装置の規模を劇的に改善することは難しい。

最近では，検証能力に制約をかけて計算量を落とす手法,[4)] BDD以外に自動テスト生成や充足可能性チェックを用いた記号モデル検査[5),6)]といった実用指向の技術が提案されている。

富士通研究所での記号モデル検査研究

記号モデル検査が発表されたのは1990年であったが，富士通研究所ではいち早くその可能性に着目し適用実験を試みた。そこではATMスイッチの動作モデルに大学が開発したツールを適用して記号モデル検査の可能性を調査した。[7)] その後，1994年に同様の適用実験を試みようとしたところ，ある程度以上の規模の装置に対しては，提案されているアルゴリズムがうまく働かないことが判明した。このため，著者らは実際の回路モデルが持つ特徴を見極めて新規アルゴリズムForward Model Checking[8),9)]を考案するとともに，これに基づく検証ツール“BINGO”を開発した。Forward Model Checkingで検証可能なプロパティは，記号モデル検査のバリエーションである言語包含検査[10)]と同等であり，従来の記号モデル検査とは若干異なる。このため，BINGOでは両方のアルゴリズムを使えるようにして幅広いプロパティに対応している。BINGOの最大の特徴は実行速度である。実際の論理装置に対してForward Model Checkingは効果的に働き，従来手法と比較して100倍以上の高速性を誇る。Forward Model Checkingは確立されたアルゴリズムの一つとして，UC Berkeleyとコロラド大が共同開発している記号モ

（注） 命題論理に時間の概念を加えたもの。

デル検査ツールVIS[11]に採り入れられている。製品化された記号モデル検査ツールの中でも同様のアルゴリズムが採用されている模様である。

また，BINGOのもう一つの特徴はハードウェア記述言語(以下，HDL)フロントエンドの部分にある。通常の記号モデル検査ツールにおいてHDLフロントエンドは単にHDL記述をツール用のデータベースに変換する。これに対し，BINGOのフロントエンドではレジスタ転送レベル(以下，RTL)の記述に対する様々な変換処理をサポートしている。記号モデル検査ツールを現実の設計に適用する上での最大の問題点は設計の規模である。すでに述べたように記号モデル検査ツールが扱える規模は300レジスタ程度であり，現実の装置の規模を大きく下回る。このため，今まで学会などで報告されている適用事例はほとんどすべて動作レベルのモデルを検証していた。これに対し，著者らはRTL記述への適用を想定し，そのためにRTL記述を縮小する技術を開発し，これをツール化した。

RTL記述縮小の目的は，検証に必要ないと考えられる回路部分を削除することにある。記号モデル検査で検証すべき内容は，信号間の時間的な依存関係という制御部分である。このため，多くの場合はデータパス部を削除しても検証上影響がない。一般の論理装置ではデータパス部分が回路の大部分を占めるため，これを削除することでモデルを扱える範囲に落とし込める可能性が高くなる。ただし，削除前後では当然モデルの動作が異なってくる。このため，削除の手順は慎重でなければならないし，削除が誤りなく行われねばならない。このうち，BINGOでは，削除すべき場所と削除の仕方を指定するとHDL記述を解析して誤りなく削除するツールを実現している。この削除作業を自動化することが望ましいが，これは極めて難しい問題であり，少なくとも５年は解決されないと予想している。当面は検証担当者の責任の下でモデル縮小が行われなければならない。

BINGOでサポートされている変換処理の一部を以下に示す。著者らがいくつかの設計にBINGOを適用する際に人手で行ってきた変換処理を分析し，必要な処理を抽出した結果である。

・信号への定数代入，および伝播
・信号のビット幅縮小
・信号の外部入力化
・列挙型タイプの項削減

定数代入・伝播は，信号の値を強制的に定数値に固定し，それを伝えて論理を縮小する処理である。装置に決まった動作モードがあり，動作中はモードが変化しないことが保証されている場合に用いる。信号のビット幅縮小は主にデータパスに適用する。複数ビットのデータパスに載る値が検証において本質的でない場合にはビット幅を１ないし０にするなどして関連するレジスタを除去する。また，パイプライン処理が行われている部分のように，レジスタの値が完全に外部から制御可能であるとすると，その信号を切断して外部入力として扱って構わない。列挙型タイプの削減は，例えば装置に与えられるコマンドを見て，検証する上で等価なコマンドを統合することに相当する。例えばプロセッサのパイプライン制御を検証する際には，加減算命令，論理演算命令を個々に区別する必要はない。命令が列挙型タイプで定義されているとするとこれらを統合してモデルの状態数を削減する。

設計への適用事例

今までに著者らがBINGOを実際の設計に適用した事例を表-1に示す。他で報告されているのと同様の動作レベルへの適用だけでなく，RTL記述に適用している点が大きな特徴である。本章ではマルチメディアプロセッサにBINGOを適用した事例を報告する。[12]

プロセッサのブロック図を図-1，全体的な適用手順を図-2に示す。本設計は数万のレジスタを持ち，記号モデル検査のアルゴリズムで扱える規模を大きく超えている。単に検証ができないだけでなく，記号モデル検査で用いる有限状態機械のデータ構造も作れない。このため，検証の対象とする部分を限定する必要がある。

そのためには，まず何を検証するかを定める。記号モデル検査の特徴は，複雑な動作を網羅的に洗える点である。したがって，記号モデル検査に適しているのは複雑な制御部分，とくに並行動作にかかわる制御部分であると言える。今回のモデルではバスの動作に着目する。本

表-1　BINGOの適用事例

事　例	設計レベル	検証内容	検証結果
マルチプロセッサシステム	動作	キャッシュプロトコル	設計誤りの修正がプロトコル上正しく行われたことを保証
サーバ用ネットワーク	動作	バスプロトコル	プロトコル上ライブロックが起こらないことを保証
マルチメディアプロセッサ	RTL	バス調停制御	バス調停における設計誤りの修正が不完全であることを発見
交換機サブシステム	RTL	クロック同期制御	初期化の際に不具合があること，それ以降問題ないことを証明
組込み用プロセッサ	RTL	メモリ，バス制御	あるレジスタで意図しない値の上書きが起こり得ることを発見
VLIWプロセッサ	RTL	キャッシュ制御	命令がキャンセルされたのに終了通知が出る場合があることを発見

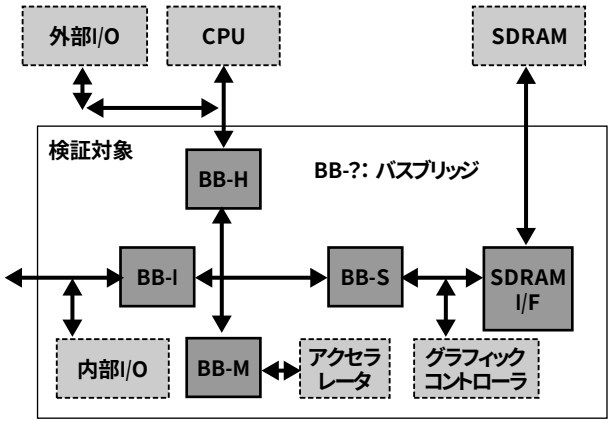

図-1　マルチメディアプロセッサの構成
Fig.1-Configuration of multimedia processor.

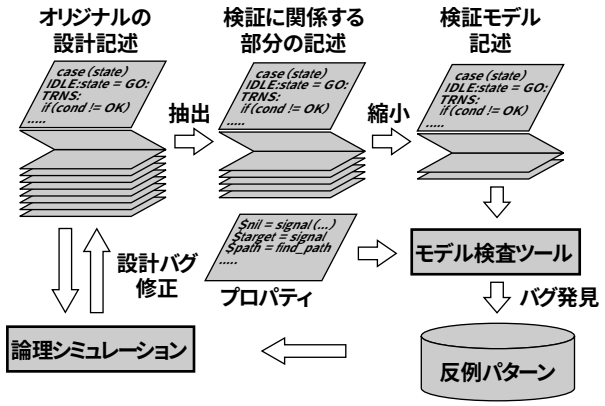

図-2　検証フロー
Fig.2-Verification flow.

プロセッサは複雑なバス構成を採り，ブリッジが複数含まれる。ブリッジ間の調停制御を検証する場合，検証モデルは図-1で四角に囲われた部分に限定できる。これはブリッジの内側から見て，外側のユニット群は単にランダムにバストランザクションを発生させる装置と見てよいからである。この限定処理は前章で示した「信号の外部入力化」を適用する。

つぎにこの検証モデルを縮小する。今回の検証で注目しているのはバス制御であるため，データパス部は検証に無関係である。このため，データパス部に含まれる信号，レジスタを除去する。このほかにも検証するプロパティに関係ない部分を除去する。これは前章における「ビット幅縮小」処理によって実現される。

モデル縮小と並行して検証項目の抽出，プロパティ作成を行う。検証項目としてはバスに対するアクセスのパターンから８種類を抽出した。例えば，プロセッサからSDRAMへのアクセス要求が出たとすると，必ずデータが返ってくるか，などである。検証項目抽出はシミュレーションベース検証におけるパターン生成でも行う作業である。シミュレーションベース検証では，この後項目に該当するパターンを細かく作成する必要があるが，記号モデル検査ではこれをそのまま時相論理で記述すればよい。細かいパターンに対応する動作は状態空間探索の際に自然に調べられる。

BINGOではプロパティ，検証の実行制御，結果のフォーマッティングをスクリプト言語Perlを用いて行う。通常は，このPerlスクリプトを実行することで検証結果として，プロパティが成り立つかどうか，成り立たない場合は反例となる入力列を生成する。

この検証では，設計誤りが見つかった部分の修正が正しく行われたことをチェックした。その結果，一つの例を除いて正しく修正されていること，「一つの例」では直したはずの部分で違うパターンの不具合があることを見つけ出した。この検証で探索した状態数は10^{13}を超えており，乱数パターンのシミュレーションではほとんど発見不可能であった。

本検証に要した工数を大まかに分けると，設計の理解に２か月，設計モデルの縮小に２か月，検証作業に２か月を要した。検証作業を行った時点では先に述べた設計モデル縮小ツールは存在せず，すべて人手で縮小作業を行った。ツールを使うことで，この工数は２～３週間程度に抑えられると考えている。検証作業は設計修正への対応を含んでいる。

以上のように，大規模なRTL記述に記号モデル検査を適用するには相当の工数を要する。しかし，作業のほとんどは設計の理解に費やされ，この部分を除けば1.5か月程度の期間で複雑な制御の検証が行える。とくにバス制御のように，複数のユニットが関係するような動作に対しては，人間が起こり得るすべての状況を列挙することは不可能に近い。このような動作の検証には記号モデル検査が大きな威力を発揮する。

む　す　び

形式的検証技術の一つである記号モデル検査技術に関して，技術の概観，富士通研究所の取組み，設計への適用方法に関して説明した。記号モデル検査技術を用いた検証は，組合せで生じる複雑な動作を確実に洗えるという点で，シミュレーションベースの検証にない特徴を有する。

一方，BDDベースの記号モデル検査技術はここ数年でほぼ成熟してきている。記号モデル検査技術の持つパ

ワーを実際の設計検証に生かすためには，技術の特徴や限界をふまえ，補完的な技術と組み合わせて，安定した性能を発揮するツールを提供することが必要である。その際に必要な技術としては，HDLの処理技術，自動テスト生成や充足可能性チェック技術を応用した記号モデル検査技術が挙げられる。

今後，富士通研究所では記号モデル検査技術の高度化に関する研究を進めるとともに，検証工数の急激な増大を抑えるような設計手法についても検討していきたいと考えている。

参考文献

(1) Burch et al.：Sequential Circuit Verification Using Symbolic Model Checking. Proc. 27th Design Automation Conference, pp.46-51, 1990.

(2) McMillan：Symbolic Model Checking. Kluwer Academic Publishers, 1993.

(3) Bryant：Graph Based Algorithm for Boolean Function Manipulation. *IEEE Transactions on Computers*, C-35(8), pp.677-691(1986).

(4) Biere et al.：Symbolic Model Checking without BDDs. TACAS '99, 1999.

(5) Boppana et al.：Model Checking Based on Sequential ATPG. CAV '97 Computer-Aided Verification, pp.418-430, Springer, 1999.

(6) Biere et al.：Symbolic Model Checking using SAT Procedures Instead of BDDs. Proc. 36th Design Automation Conference, pp.317-326, 1999.

(7) Chen et al.：Bug Identification of a Real Chip Design by Symbolic Model Checking. Proc. European Design and Test Conference, pp.132-136, 1994.

(8) H. Iwashita et al.：CTL Model Checking Based on Forward State Traversal. Proc. Int'l Conf. Computer-Aided Design, pp.82-87, 1996.

(9) H. Iwashita et al.：Forward Model Checking Techniques Oriented to Buggy Designs. Proc. Int'l Conf. Computer-Aided Design, pp.400-404, 1997.

(10) Touati et al.：Testing Language Containment for ω-automata Using BDD's. Proc. 1991 International Workshop on Formal Methods in VLSI Design, 1991.

(11) The VIS Group：VIS：A System for Verification and Synthesis. Proc. 8th Conference on Computer Aided Verification, pp.428-432, 1996.

(12) K. Takayama et al.：An Approach to Verify a Large Scale System-on-a-Chip Using Symbolic Model Checking. Proc. ICCD-98, 1998.