

エミュレーションシステムによるシステムレベル検証

2D-08

清野 克幸† 横浜 宏紀‡ 花島 育雄‡ 佐藤 孝一‡

†NEC通信システム 技術本部 第三技術部

‡NECネットワークス ネットワークシステム事業本部 ネットワークノード事業部 設計技術部

1. はじめに

近年の通信系システムの機能の多様化／複雑化に伴いシステムの核となるLSIの大規模化が進み、LSI単体レベルの検証のみで設計品質を保つのが困難な状況になりつつある。その対策としてエミュレータを用いたシステムレベルの検証手法がある。エミュレータのゲート規模／メモリ容量の制約に対して有効な検証モデル化の方法について述べる。

2. 現状の課題

通信系システムに対してエミュレータ手法によるシステムレベルの検証を実施する場合、少なくとも数10MGのゲート容量・数100MB以上のメモリ容量を搭載したエミュレータが必要となる。しかしながら、エミュレータが高額である事から現実的には、その数分の1に満たない容量であり、その環境下で有効な検証を行わなくてはならない。

また、エミュレータのコンパイル時間は1つの検証モデル当たり約半日もかかり、検証中の解析・修正を含めると実質1日に実行可能なコンパイル回数はせいぜい1～2回が限度の状況である。限られたコンパイル回数の中、設計論理バグ以外でもライブラリ論理ミス等のモデル化ミスによる後戻りインパクトは非常に大きく、モデル化においてはミスが許されない。

現状の課題

- ①検証規模制限への対応
- ②モデル化品質の向上

3. 課題への対策

- (1) 検証規模制限への対応

エミュレータの規模制限内の有効運用として次の対策を検討した。

- ①検証ブロック構成管理
- ②外付け汎用メモリ

ここでは①検証ブロック構成管理について述べる。

①検証ブロック構成管理

課題でも述べたが現在のエミュレータの容量制限上、システムレベルのH/Wモデル全てをエミュレータには展開出来ない。検証規模の制限内で検証項目に応じて検証モデルの組み合わせの構成を容易に管理・作成出来る仕組みを検討した。

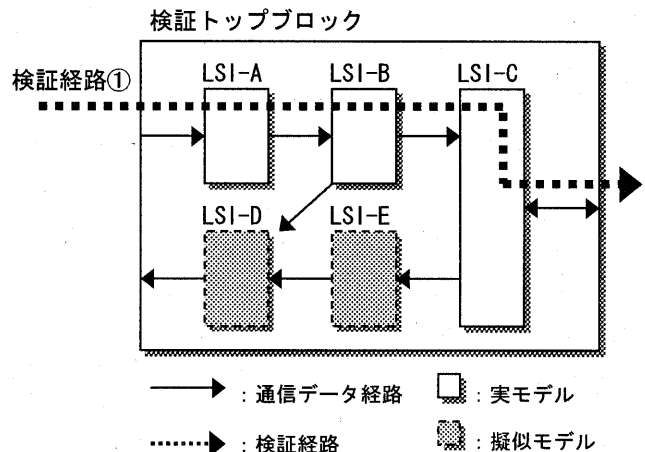


図1 検証経路①のトップブロック構成

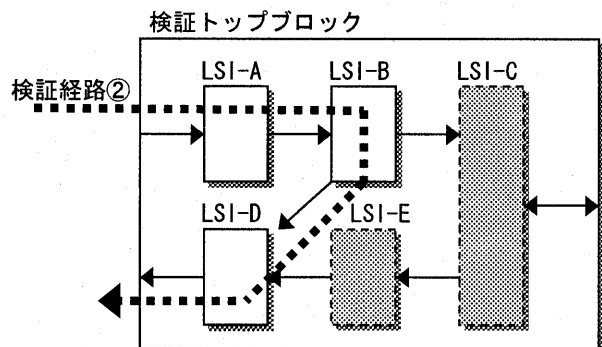


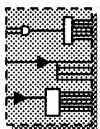
図2 検証経路②のトップブロック構成

図1・図2の様に同じ検証トップブロックの構成でも検証経路によって検証構成に必要なモデルと必要でないモデル（擬似モデルによる代替可能）が出てくる。その検証ブロック構成の検証モデルの組み合わせを検証項目単位に図3による表形式で情報管理する。

●：実モデル ○：擬似モデル

検証項目	検証モデル				
	LSI-A	LSI-B	LSI-C	LSI-D	LSI-E
検証経路①	●	●	●	○	○
検証経路②	●	●	○	●	○
⋮	⋮	⋮	⋮	⋮	⋮

図3 検証ブロック構成



擬似モデルは実モデルと接続構成した時に検証論理に影響しない簡易的な論理モデル（信号スルーや固定値出力のモデル）。このモデルにより上位階層の接続関係を変えずにモデル構成を変更可能とする。

図4 擬似モデル

この検証ブロック構成管理情報を次の項目で説明する検証用モデル自動生成の制御ファイルに記述する事で容易に検証項目に応じた検証モデルの作成を可能とする。

(2) モデル化品質の向上

課題に示す通りエミュレータのコンパイルは約1日がかりの作業となる為、検証モデル化の品質確保には十分に注意しなくてはならない。その対応として次の対策を検討した。

- ①検証用モデル自動生成
- ②単体モデル期待値動作確認
- ③部品ライブラリー一元管理
- ④モデル化作業チェックシート

ここでは①検証用モデル自動生成について述べる。

①検証用モデル自動生成

通常、LSI結合レベルの検証を行う場合、実際のボード設計回路とは別にアナログ部品等の検証に必要な回路を取り除いた検証用の結合ブロック回路を作成し、その検証用回路を用いて検証を行う。しかし、この回路の二重作成によって設計回路と検証回路との等価性の問題や回路修正・更新によるデータ

の二重管理の問題が出てくる。そこで検証用の結合ブロック回路を実際のボード設計回路を入力として自動生成するツールを開発した。

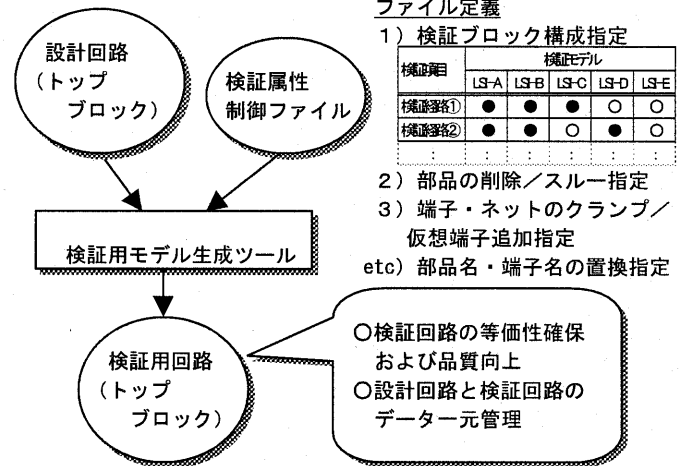


図5 自動生成ツールの入出力データ

ツールの主な機能は次の通りである。

- 1) 検証構成情報による実モデル/擬似モデル指定
- 2) 抵抗・コンデンサ等のアナログ部品削除
- 3) 空き端子・フローティングネット処理

上記の処理定義を制御ファイルに記述する事によりツールの処理機能を実現する。

この自動生成ツールの実現により、モデル化の品質確保やデータの一元管理化を図る。

4. 結果

課題対策の実施により次の結果が得られた。

①検証規模制限への対応

→ システムの検証構成により適用は限られるが、全体構成規模に対し1/3～1/5のエミュレータ規模での有効検証が可能となった。

②モデル化品質の向上

→ ツール化・チェック強化等によりモデル化ミス30%削減の品質向上が図れた。

5. おわりに

今回の結果では、モデル化品質の向上に挙げている全ての項目実施に至っていない。今後も対策実施継続でさらなる品質強化を図って行く予定である。