

レジスタ転送 (RT) レベルの論理回路記述から機能検証用のテストパターンを自動的に生成するシステムについて述べる。システムは、RT レベルのモデルの機能動作を検証するために用いる新しいテストパターンを生成するものである。このシステムを用いることにより、従来人手で作成していたテストパターンと同程度の質のテストパターンを(半)自動的に生成することができる。さらに、このシステムにより RT レベルのモデルの誤りを発見することもできる。

This paper describes a new tool to generate stimuli for functional verification from a register transfer level (RTL) description with a path activation technique. Experiments have shown that stimuli generated for functional verification at the register transfer level are acceptable for practical designs.

Moreover, a new design approach for checking circuit models is addressed. That is, if stimuli for a desired behavior cannot be generated under certain constraints, then the design is incorrect.

1 まえがき

機能シミュレータと論理合成ツールが実用化されたことにより、現在では論理 LSI は次に示す手順で設計されるのが一般的である。

- (1) モデリング RT レベルのハードウェア記述言語を用いて作製したい論理 LSI のモデル (RT レベルの記述 (RTL 記述)) を作成する。
- (2) 機能検証 テストパターンを作成し機能シミュレーションを行い、モデルが正しいかどうかを検証する。
- (3) 論理合成 RT レベルの記述をゲートレベルの記述に自動的に変換する。

このうちもっとも時間のかかるステップは機能検証である。このため、機能検証を効率的に行うためのツールの開発への要求が高まっている。論理 LSI のゲート規模が増加、構成される論理が複雑化するにしたがい、とりわけ記述を検証するためのテストパターンを作成する時間が増加してきている。われわれは、機能設計の段階で機能検証用のテストパターンを効率よく生成することが機能検証の効率化に役だつと考え、RT レベルの自動テスト生成システム “FUNTASSI” の開発を行っている。

論理回路記述を用いてハードウェアの故障を検出するためのテストパターンをアルゴリズム的に生成する方法はいろいろ研究されている。目的は異なるが、“FUNTASSI” もアルゴリズム的な手法を用いてテストパターンを生成するシ

ステムである。論理回路記述を用いて機能検証用のテストパターンをアルゴリズム的に生成することにより、場合によってはモデルの誤りを見つけられる、という副次効果がある。なぜならば、テストパターンが得られないということは、モデリングの段階で作成した論理回路の記述に誤りがあることを示している場合があるためである。

2 テストパターン生成

“FUNTASSI” は論理回路のモデルの機能検証を行うためのテストパターンを生成する、または、論理回路のモデルの誤りを検出する。モデリングの段階で作成された RTL 記述はコンパイラにより文法のチェックが行われる。記述が文法的に正しい場合、RTL 記述は計算機での扱いを容易にするためにネットワークデータ構造へと変換される。“FUNTASSI” はこのネットワークデータにアクセスすることによりテストパターンを生成する。“FUNTASSI” では二通りの方法 (ノードの活性化、パスの活性化) でテストパターンを生成することができる。

2.1 ネットワークデータ

図 1 は論理回路の RTL 記述と対応するネットワークデータを表しており、図 1(a) に示されている記述はコンパイラにより図 1(b) へと変換される。実線の正方形はデータノードと呼ばれるものであり、RTL 記述中の入力端子、出力端子または内部端子に対応する。波線の正方形は条件ノードと呼ばれる

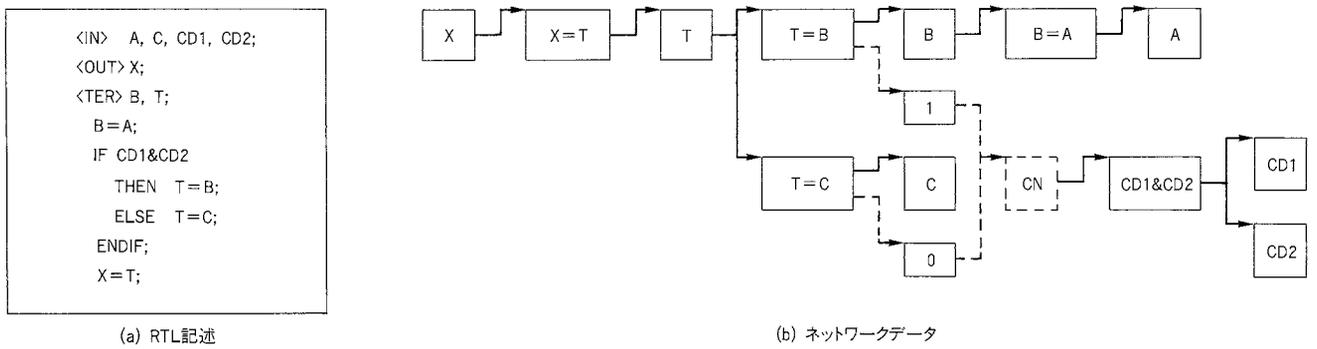


図1. RTL 記述とそのネットワークデータ RTL 記述(a)はコンパイラによりネットワークデータ(b)に変換される。
RTL description and its network data

ものであり、条件文の出力を表す仮想的なノードである。長方形はRTL 記述中の文に対応する。実線の矢印は出力側から入力側への接続を表す。例えばノード X から文 X=T を経由しノード T へ至る接続は、T に存在する信号が文 X=T を経て X へ流れることを表している。文 T=B から条件ノード CN への接続 (波線の矢印で示されている) は、CN の値が1 のとき文 T=B が実行されることを表している。

2.2 ノードの活性化

(N, v) はノード N を値 v に設定することを意味するものとする。“FUNTASSI” が (N, v) という要求を受けると、N の値を v にするための入力系列を求めるを試みる。N が入力端子ではないとき、N を出力としてもつ文 S が選択され、N の値が v となるような S の入力側のノードの値を求める。値が得られたら、入力側のおおののノードに対して上述の処理を繰り返す。S がある条件の下で実行される場合 (すなわち S が条件文への接続をもつ場合)、さらに条件値の設定を試みる。条件値の設定も通常のノードの活性化の処理とほぼ同様である。ノードの活性化アルゴリズムを図 2 に示す。

2.3 パスの活性化

P を信号が伝搬する回路中のパスとする。“FUNTASSI” が P を受けると P を活性化し、すなわち P の入力に存在するデータを P の出力まで伝搬させる入力系列を求めるを試みる。P は RTL 記述の文の列として次のように表される。

$$P = S_0 S_1 \dots S_n$$

P を活性化することは、各 S_i に対して S_i がある条件の下で実行される場合、条件値を適切な値に設定することである。したがって、パスの活性化は複数のノードを同時に活性化することによって実現できる。

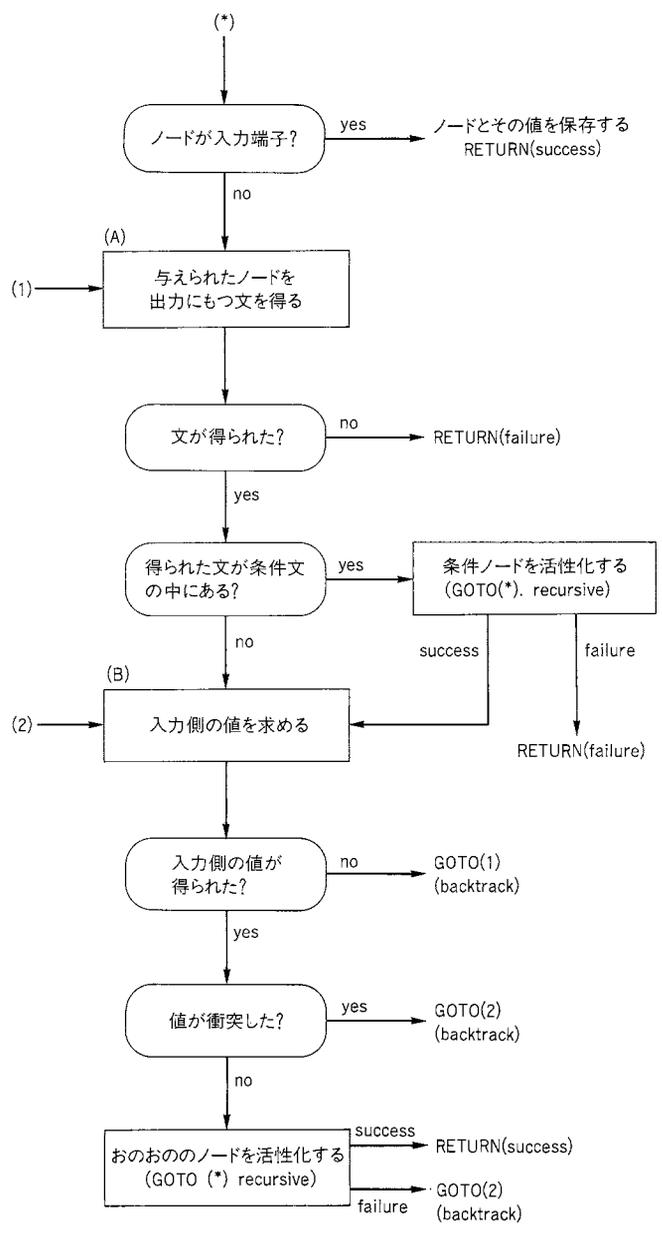


図2. ノードの活性化のアルゴリズム “FUNTASSI” はネットワークデータの接続をたどりながら、このアルゴリズムを用いてテストパターンを生成していく。
Algorithm of node activation

3 機能動作のチェック

簡単な RTL 記述の例を用いて記述の誤りを調べる方法を述べておく。

(1) “FUNTASSI” で得られたテストパターンを用いた機能シミュレーションによる誤りの発見 要求された動作を実現するテストパターンが得られたからといって、その動作が正しくモデリングされているとは限らない。図3は信号の衝突が起こる例である。“FUNTASSI” にパス A → X を与えると、(CND3, 1) という入力系列を生成する。しかし、この入力系列は文 X=B をも活性化してしまうため、X において信号の衝突が起こる。モデリングの正しさを確認するためにはシミュレーションを実行する必要がある。

```
CND1 = CND3;
CND2 = CND3;

if CND1 then X = A; endif;

if CND2 then X = B; endif;
```

図3. 信号の衝突が起こる RTL 記述の例 (CND3, 1) を入力すると A の信号値と B の信号値が X で衝突してしまう。
Example of RTL description in which conflict is caused

(2) テストパターンが得られないことによる誤りの発見 図4は与えられた動作に対するテストパターンが得られない記述例である。“FUNTASSI” にパス A → T → X を与えると、A → T を活性化するために (CND3, 1), T → X を活性化するためには (CND3, 0) がそれぞれ必要とな

```
CND1 = CND3;
CND2 = not(CND3);

if CND1 then T = A; endif;

if CND2 then X = T; endif;
```

図4. 入力系列を生成することができない RTL 記述の例 A から T を経由して X へ信号を伝搬させることができない。
Example of RTL description in which stimulus for propagating data signal cannot be generated

るため、値が衝突してしまいテストパターンが得られない。値の衝突が起こりテストパターン生成ができなかった場合には、どこで衝突が起こったかを示す情報が出力される。この情報は、記述の誤りを修正するときに役にたつ。

4 適用例

“FUNTASSI” を浮動小数点演算回路の RTL 記述に適用し、機能検証用のテストパターン生成を行ってみた。この回路は二つの正規化された浮動小数点数を同一の指数をもつ形に変換(けた合せ)し、加算または減算を行い、計算結果を正規化して出力するものである。けた合せの処理および正規化の処理は、それぞれ、入力となる二つの浮動小数点数の指数の差、加算または減算後の仮数部の値により異なる。これらの処理は論理回路中の一つのパスに対応する。そこで、おのおのパスに対してパスの活性化を実行し、処理動作を検証するためのテストパターン生成を行った。その結果人手で作成したのと同程度の質のテストパターンを生成することができている。

5 あとがき

RT レベルの論理回路記述から機能検証用のテストパターンを自動的に生成するシステムのプロトタイプを開発した。このシステムを用いることにより RTL 記述の誤りを発見することもできる。また、“FUNTASSI” に実現不可能な動作を与え、それを実現する入力系列が得られたことにより記述に誤りがあると結論づける、という使いかたも考えられる。

さらに、このシステムのコアであるテスト生成部は、機能レベルの記述を用いたハードウェアの故障検出用のテスト生成にも適用可能であり、現在その検討も行っている。



河野 和義 Kazuyoshi Kohno

1986 年入社。機能論理設計 CAD の研究・開発に従事。現在、研究開発センター ULSI 研究所研究主務。
ULSI Reserch Labs.



関根 優年 Masatoshi Sekine

1976 年入社。バイポーラメモリ、64 K ビット C RAM、VLCAD 機能設計 CAD の研究・開発に従事。現在、研究開発センター ULSI 研究所研究主幹。
ULSI Reserch Labs.