

5 J-2

正規表現を用いたインタフェース仕様記述からの
プロトコル検証用テストベンチ生成法の提案時久卓也[†] 北嶋暁^{††} 武内良典[†] 今井正治[†] 鈴木敬^{†††}[†] 大阪大学大学院基礎工学研究科情報数理系専攻^{††} 大阪電気通信大学総合情報学部 ^{†††} 日立製作所中央研究所

1 はじめに

近年, 大規模な回路が製造可能になったため, テストベンチの開発の負担は大きくなってきており, テストベンチ自動生成の重要性が高まっている. [1]

本稿では, 正規表現を用いたインタフェース仕様記述 [2] とパラメータを与え, 対象モジュールのプロトコル検証を行うテストベクタを含んだテストベンチを自動生成する手法を提案する. ここで用いるインタフェース仕様記述は, 意味のある一連の信号変化を一つの記号で表し, 階層的にモジュール全体の機能を正規表現で表した記述である. 提案手法では, 機能が記号で表現できることを利用して, どのようなテストベクタを生成するかの指定を, 各機能の実行回数を表すパラメータにより行う. 本提案手法により, ある機能に着目したテストベンチを生成することが容易になる.

本手法を信号制御モジュールに適用することによって, 本手法が有効である見通しを得た.

2 正規表現を用いたインタフェース仕様記述

正規表現を用いたインタフェース記述は主に以下の項目で構成される.

port 定義 入出力の端子についての入出力方向, 幅と信号属性(クロック, 制御系, データ系), 名称の定義. ここで, 制御系とは, モジュールの制御を行う信号, データ系とは, アドレスや演算に用いるデータなどを表す.

alphabet 定義 モジュールの実行時に生じ得る, ある時刻における端子の信号値の組み合わせに対する名前.

word 定義 ある機能を実行したときに起こり得る信号系列に対する名前. alphabet, word を用いた正規表現で表す.

Automatic test bench generation for protocol verification from interface specification in regular expression

Takuya Tokihisa[†], Akira Kitajima^{††}, Yoshinori Takeuchi[†], Masaharu Imai[†], and Kei Suzuki^{†††}

[†]Department of Informatics and Computer Science, Graduate School of Engineering Science, Osaka University

^{††}Department of Engineering Informatics, Osaka Electro-Communication University

^{†††}Central Research Laboratory, Hitachi, Ltd.

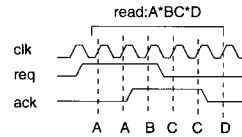


図 1 波形と正規表現の対応例

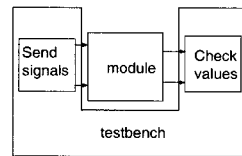


図 2 テストベンチの構造

sentence 定義 alphabet, word を用いた正規表現で表した, 機能モジュール全体の動作.

例えば, 図 1 のような信号変化をする機能では, A (req=1, ack=0), B (req=1, ack=1), C (req=0, ack=1), D (req=0, ack=0) が alphabet として定義される. そして, 'AABCCD' は, 'read' のパタンに含まれる. この場合, 'read' は 'A*BC*D' で表される word の一例である.

機能を表す word に着目することによって, 特定の機能に着目したテストベンチ生成が容易になる.

3 テストベンチ生成法

本手法は有限状態を持つ制御回路に対してテストベンチを自動生成する.

3.1 テストベンチのモデル

本手法で生成するテストベンチは図 2 の構造である.

本手法で生成するテストベンチは, 以下の動作を行う.

対象モジュールに対して, 入力値をクロック単位で与えて動作のシミュレーションを行い, クロック単位で出力値を期待値と比較して, 対象モジュールの誤りを検出する. 入力値と出力の期待値はテストベクタから得る.

3.2 提案手法

本稿で提案する手法の概要を図 3 に示す. 本手法はテストベクタ生成とそれ以外の部分の生成の二つに大きく分かれる.

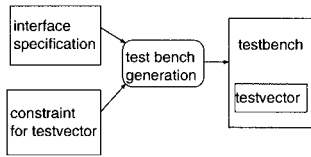


図3 提案手法の概要

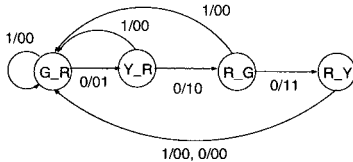


図4 信号機の動作を表した状態遷移図

テストベクタ生成では対象モジュールのインタフェース仕様記述とパラメータを入力として与える。ここで、パラメータとは、一つのテストベクタの長さ（実行サイクル数）の上限と下限、一つのテストベクタに現れる機能の出現回数の上限である。与えられた正規表現を用いたインタフェース記述からトンプソンの構成法[3]を応用して、状態遷移図を word ごとに階層的に作成する。次に、与えられたパラメータを満たすように状態遷移図を網羅的に辿ることによって、テストベクタを生成する。

テストベンチのテストベクタ以外の部分は、インタフェース仕様記述の port 定義より生成する。

4 適用例

例として信号機の制御部を考える。

信号機の制御部は、初期状態が G_R であり、以下の状態を持つ。

G_R: 道路 main が青, 道路 sub が赤, 出力は '00'

Y_R: 道路 main が黄, 道路 sub が赤, 出力は '01'

R_G: 道路 main が赤, 道路 sub が青, 出力は '10'

R_Y: 道路 main が赤, 道路 sub が黄, 出力は '11'

図4はこの信号機が正しい動作をする場合の状態遷移図である。初期状態の G_R から、reset 信号が '0' である限り、Y_R, R_G, R_Y, G_R と遷移し、再び Y_R へ遷移する。reset 信号が '1' であると、G_R へ遷移する。

ここで、誤って図5で表される制御部を設計したとする。このとき、状態 R_Y で、reset 信号 '0' が入力されたとき、本来は G_R へ遷移しなければならないが、R_Y へ遷移するという誤りが生じる。

図5の動作を行うモジュールに対し、図4を表した仕

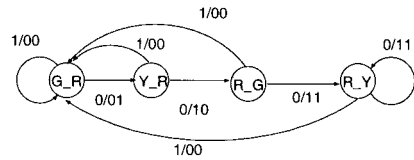


図5 誤りを含んだ信号機の動作を表した状態遷移図

様記述を用いて、提案アルゴリズムを適用する。G_R, Y_R, R_G, R_Y, G_R と遷移する動作が一つの word として定義されており、この word の出現回数の上限を1回、その他の word の出現回数の上限を0回、テストベクタの長さの上限と下限をともに4というパラメータを与えると、図5の誤りを検出できるテストベクタを持つテストベンチが生成される。

5 考察

本手法を利用すると、パラメータを適切に与えることによって、例えば以下の動作を行うテストベンチを作成することが可能である。

- 各機能を最低1回実行する
- 特定の機能を組み合わせる
- 特定の頻度で特定の機能を実行する

上記のように、一定の方針のもとにテストを行う場合は、対応するパラメータを自動的に導出することが可能である。このように、典型的なテスト戦略に対して自動化を進めることにより、より効率的な機能検証が行えると考えられる。

6 おわりに

本手法を例題に適用した結果より、パラメータを適切に与えることにより、プロトコル検証を行うことができる見通しを得た。

今後の課題としては、制御部だけでなくデータパスの検証も行うことのできるテストベクタを生成する方法の考案が挙げられる。

参考文献

- [1] Julia Dushina, Mike Benjamin, and Daniel Geist. Semi-formal test generation with genevieve. In *Proceedings of the 38th Design Automation Conference*, pp. 617-622, Las Vegas, NV, June 2001. ACM.
- [2] 荒宏視, 鈴木敬, 矢野和男. IP 再利用を促進するインタフェース記述言語 Owl. 情報処理学会研究報告 2000-SLDM-97, Vol. 2000, No. 97, pp. 25-32, September 2000.
- [3] A.V. エイホ, R セシイ, J.D. ウルマン. コンパイラ I 原理・技法・ツール. サイエンス社, 1990. 原田賢一 訳.