

5P-2 組合せ論理回路に対するモデルベース検証について

平塚 聡 房岡 璋
立命館大学工学部情報学科

1. はじめに

本論文では、モデルベース推論に基づいて、組み合わせ論理回路の設計検証を行う方法について報告する。組み合わせ論理回路の設計誤りを特定し自動修正する方法に関しては、今まで多くの研究がおこなわれているが、test vector を用いたシミュレーションによる方法と形式検証による方法の2つが代表的なアプローチである [2]。前者は、test vector を用いて検証の対象であるゲート図 (Gate level Implimentation、以下 GI) の診断を行い、この結果から誤りの箇所を同定するものである。stuck-at 故障の診断をそのまま用いる方法も報告されているが [3]、一般には設計誤りに対する test vector 生成を如何に行うかが問題である。もとより、この方法では完全な設計検証は不可能である。

後者は、機能仕様 (Functional Specification 以下 FS) と GI の等価性の形式的検証を行い、この結果から設計誤りの箇所と種類を同定するものである。この場合、FS と GI の乖離が大きいと、実際は再設計と同じことになってしまい、誤りの同定や修正がきわめて難しくなる。設計誤りは、CAD 出力の部分修正や最適化など設計の手作業部分で発生するものであり、殆ど以下のパターンに該当するといわれている [3]。

- (1) Inverter の欠如、過剰
- (2) 結線 (wire) の欠如、過剰
- (3) Incorrect gate type (ANDgate と ORgate の取り違い等)

従って、これらのパターンを組み込んだ問題設定を行うことにより、より効率的な方法が得られる可能性がある。

ここでは、GI に上記の設計誤りのパターンを組み込んだ設計の形式モデル (System Description 以下 SD) を与え、これと FS に対してモデルベース推論を行うことにより誤りを特定する方法を与える。以下では、特に Inverter の欠如、過剰のみを取り上げる。

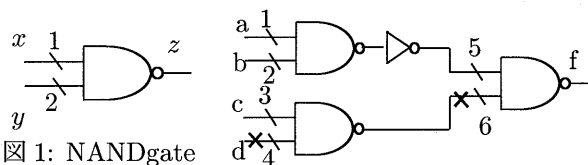


図 1: NANDgate

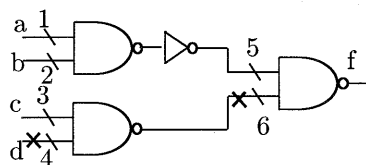


図 2: GI

2. 問題の記述

諸記号

以下では、次の記号を用いる

1. 入力変数 a, b, c, \dots 出力変数 f
2. 中間変数 $1, 2, 3, \dots$
3. Abnormality 述語 A_1, A_2, A_3, \dots

また、論理関数 $f(x_1, x_2, \dots)$ に対し、

$$f_{x_i}^1 = f(\dots x_{i-1}, 1, x_{i+1} \dots), \quad f_{x_i}^0 = f(\dots x_{i-1}, 0, x_{i+1} \dots)$$
$$\forall x f(x) = f_x^1 \cdot f_x^0, \quad \exists x f(x) = f_x^1 + f_x^0 \text{ と定義する。}$$

ゲート記述

以下では、GI は 2 入力 の NANDgate と Inverter のみから構成されていると仮定する。

GI のすべての結線 i に Abnormality 述語 A_i を付加し、 A_i が成り立つとき、この結線に Inverter が欠けていることを表すものとする。形式モデルは、各 A_i が成立する場合と不成立の場合の双方に対して動作を記述する論理式である。

図 1 のゲートに対し、入力線 1, 2 に Inverter の欠如がある場合、Abnormality 述語 A_1, A_2 は入力線 1, 2 に Inverter の欠如があることを示すが、以下では A_1, A_2 は同時に成り立たないと仮定する。すなわち、GI 全体としては誤りは複数箇所あってもよいが、各ゲートの入力線に対しては高々 1 個であると仮定する。

このとき、このゲートの特性関数は以下の節形式で与えられる。

$$\{A_1 + x + z\} \{\overline{A_1} + \bar{x} + z\} \{\overline{A_1} + x + \bar{y} + \bar{z}\}$$
$$\{A_2 + y + z\} \{\overline{A_2} + \bar{y} + z\} \{\overline{A_2} + \bar{x} + y + \bar{z}\}$$
$$\{A_1 + A_2 + \bar{x} + \bar{y} + \bar{z}\}$$

加えて、 $\{x + y + z\}$ が成り立つが、これを Sys-rule と呼ぶ。

システム記述

与えられた GI に対し、システム記述 SD は、GI の各

*On A Model-Based Verification For Combinatorial Boolean Network
Satoshi Hiratsuka, Akira Fusaoka
hira@muse.cs.ritsumeikan.ac.jp
Department of Computer Science, Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu Shiga 525-8577 Japan

ゲートに対するゲート記述の集合である。これを、

$\Sigma(a, b, c, \dots; f; 1, 2, \dots, n; A_1, \dots, A_n)$ と記述する。

仕様記述

仕様記述は入出力変数の論理関数であり、やはり節形式として与える。これを、

$\Psi(a, b, c, \dots; f)$ と記述する。

問題の記述

設計検証の問題は、任意の入力に対して、SD : Σ と FS : Ψ が同時に充足される時、Abnormal 述語 A_1, A_2, A_3, \dots に対して成り立つ論理関係を求めること、すなわち、

$(\forall a, b, c, \dots), (\exists f)(\exists 1, 2, 3, \dots)[\Sigma \wedge \Psi]$

から、 A_1, A_2, A_3, \dots のみの論理関数を演繹することに帰着する。以下でその方法を与える。

3. 検証アルゴリズム

推論ルール

以下の4種類の推論規則のみを用いる。

1. **Sys-rule** 連結された2つのゲートに対し、下段ゲートの Sys-rule を用いて上段の特性関数を単純化する。
2. **Elim-rule** $\{L + i\}\{K + \bar{i}\} \Rightarrow \{L + K\}$
但し、 i は中間変数又は出力変数である。 $(\exists i)[(L + i) \cdot (K + \bar{i}) \equiv L + K]$ より得られる。すなわち、**Elim** は中間および出力変数消去のためのルールである。
3. **Gen-rule** $\{L + K\} \Rightarrow \{K\}$
但し、 L は入力変数のリテラルのみの論理和、 K は Abnormal 述語のリテラルのみの論理和である。
 $(\forall a)[L(a) + K(A_1, \dots)] \equiv K(A_1, \dots)$ より得られる。すなわち、**Gen** は入力変数を一般化するためのルールである。
4. **Simp-rule** 上記より得られた節を用いて resolution により、 Σ を単純化する。

アルゴリズムの概略

GI をゲート木と見なしたとき、ルート of 出力側ゲートより、スタートし、depth-first の順序で以下の手順を行う。

1. 出力変数から入力変数に至る path を1つ固定し、最下段のゲートの結合に対し Sys-rule を適用する。
2. Elim-rule を用いて中間変数を順次消去し、これを含まない節を導出する
3. 中間変数を含まない節に対し Ψ を用いて出力変数 f を消去する
4. Gen-rule を用いて Abnormal 述語のリテラルのみの節を導出する
5. これを用いて、 Σ の節を単純化する

6. この手続きを繰り返す

4. Example

図2を例にとって考える。

問題記述

FS: $\Psi \equiv \{\bar{f} + \bar{a} + b + \bar{c} + \bar{d}\}\{f + a\}\{f + b\}\{f + c\}\{f + \bar{d}\}$

SD: $\Sigma \equiv \{A_1 + a + 5\}\{\bar{A}_1 + \bar{a} + 5\}\{\bar{A}_1 + a + \bar{b} + 5\}$

$\{A_2 + b + 5\}\{\bar{A}_2 + \bar{b} + 5\}\{\bar{A}_2 + \bar{a} + b + 5\}$

$\{A_1 + A_2 + \bar{a} + \bar{b} + 5\}$

$\{A_3 + c + 6\}\{\bar{A}_3 + \bar{c} + 6\}\{\bar{A}_3 + c + \bar{d} + 6\}$

$\{A_4 + d + 6\}\{\bar{A}_4 + \bar{d} + 6\}\{\bar{A}_4 + \bar{c} + d + 6\}$

$\{A_3 + A_4 + \bar{c} + \bar{d} + 6\}$

$\{A_5 + 5 + f\}\{\bar{A}_5 + 5 + f\}\{\bar{A}_5 + 5 + \bar{b} + f\}$

$\{A_6 + 6 + 5\}\{\bar{A}_6 + \bar{b} + f\}\{\bar{A}_6 + 5 + 6 + f\}$

$\{A_5 + A_6 + 5 + \bar{b} + f\}$

推論手法 path $f - 5 - 1, 2:$

sys-rule $\{a + b + 5\}, \{c + d + 6\}$ を用いて

$\{\bar{A}_5 + a + b + c + d\}$, Gen-rule により、 $\{\bar{A}_5\}$

Simp-rule により、 $\{5 + f\}, \{A_6 + 5 + \bar{b} + f\}$

Elim-rule により 5 を消去して $\{\bar{A}_1 + \bar{a} + f\}\{\bar{A}_2 + \bar{b} + f\}$

f を消去して、 $\{\bar{A}_1 + \bar{a} + \bar{b} + \bar{c} + d\}, \{\bar{A}_2 + \bar{a} + \bar{b} + \bar{c} + d\}$

Gen-rule より、 $\{\bar{A}_1\}, \{\bar{A}_2\}$

Simp-rule より、 $\{\bar{a} + \bar{b} + 5\}, \{A_6 + \bar{a} + \bar{b} + \bar{b} + f\}$

path $f - 6 - 3, 4$ Elim-rule より \bar{b} を消去して、

$\{A_6 + A_3 + \bar{a} + \bar{b} + c + f\}\{A_6 + \bar{A}_3 + \bar{a} + \bar{b} + \bar{c} + f\}$

$\{\bar{A}_6 + \bar{A}_3 + \bar{c} + f\}\{\bar{A}_6 + A_4 + \bar{a} + \bar{b} + \bar{c} + \bar{d}\}$

Gen-rule より

$\{A_6 + A_3\}, \{A_6 + \bar{A}_3\}, \{\bar{A}_6 + \bar{A}_3\}\{A_6 + A_4\}$

Simp-rule より $\{A_6\}, \{\bar{A}_3\}, \{A_4\}$

従って、結線4および6に inverter を追加する必要がある。

5. むすび

ここでは、特に Inverter の欠如、過剰のみを取り上げたが、他の設計誤りに対しても同様の形式モデルを与え類似の推論により設計誤りを特定することができる。

参考文献

- [1] Dressler, O. and Struss, P. 1996. The Consistency-based Approach to Automated Diagnosis of Devices, G.Brewka eds *Principles of Knowledge Representation*, pp 267-331, CSLI Publication.
- [2] Chung, P., Wang, Y. and Hajj, I.N., 1994. Logic Design Error Diagnosis and Correction, *IEEE Trans. on VLSI Systems*, vol 2, pp 320-332.
- [3] Abadir, M.S. Ferguson, J. and Kirkland, T.E. 1988. Logic Design Verification via Test Generation, *IEEE Trans. on CAD*, vol 7, pp. 138-148.