

セルフチェックング論理回路網の一構成法

1 D-1

畠中慎一 小野尾隆一 南谷 崇
東京工業大学 工学部

1 はじめに

セルフチェックング技術は、オンライン故障検出の一手法である。回路に故障が生じた時、出力に現れる最初の誤りは必ず非符号語であるというセルフチェックング技術の目指す目標はTSC目標と呼ばれ、SFS/SCD回路はTSC目標を満たす最大のクラスの回路として知られている。SFS性は出力に現れる最初の誤りは必ず非符号語であることを保証し、SCD性は回路に故障が生じて非符号語入力に対しては必ず非符号語を出力することを保証する。本研究では、以前に報告されたSFS/SCD組合せ回路構成法[1]をもとにして、まずSFS/SCD同期式順序回路構成法を提案する。そして、SFS/SCD論理回路を構成要素とする回路網を構築する際に生じる問題を取り上げ、その解決法についても述べる。

2 SFS/SCD同期式順序回路構成法

2.1 故障集合と誤りモデル

ここで提案するSFS/SCD順序回路は、図1に示されるようにPLA(N-MOS NOT-NOR-NOR型)、記憶回路、及び、2線式符号チェッカ[2]から構成される。

PLAに仮定する故障集合 F_{PLA} には、以下の故障を仮定する。

[F_{PLA}]

1. 並行する複数信号線の単方向固定故障
2. 並行する複数信号線の短絡故障
3. 断線故障
4. 単一交点付加故障、単一交点欠落故障

そして、この故障集合に対して、PLAの入力に全てのパターンが入力される時、PLAは誤るとすればその誤りは次の3つの誤りタイプのいずれかに属する[3]。

- E1. 正しい入力が正しく写像された出力に生じる単方向誤り
- E2. 単方向誤りの生じた入力が正しく写像されて生じる誤り
- E3. 単方向誤りの生じた入力が正しく写像された出力に生じる単方向誤り(但し、入力と出力の誤る方向は同じである。)

F_{PLA} は、Fuchsらが提案したPLA故障集合[3]から"直行する2本の信号線の短絡故障"を除いた集合となっている。これは、"直行する2本の信号線の短絡故障"が生じた場合、上記の誤りタイプに属さない誤りが生じることがあるからである[4]。

記憶回路に仮定する故障集合 F_{Memory} には、記憶回路の入出力線の単方向固定故障を仮定する。また、2線式符号チェッカに仮定する故障集合 F_{2-rail} には、回路を構成する論理ゲートの入出力線の単方向固定故障を仮定する。

2.2 SFS/SCD同期式順序回路

SFS/SCD順序回路を構成するにあたり、以下のことを仮定する。

- [仮定1] 2つ以上の故障が同時に生じることはない。
- [仮定2] 任意の状態に対して、全ての入力パターンが現れるような入力系列が入力される。

図1が、提案するSFS/SCD順序回路構成法であり、入力 X 、出力 Z 、状態 S 、状態遷移関数 δ 、出力関数 ω なるMealy型順序回路を実現する。入力 X 、出力 Z 、状態 S は完全非順序分離符号で符号

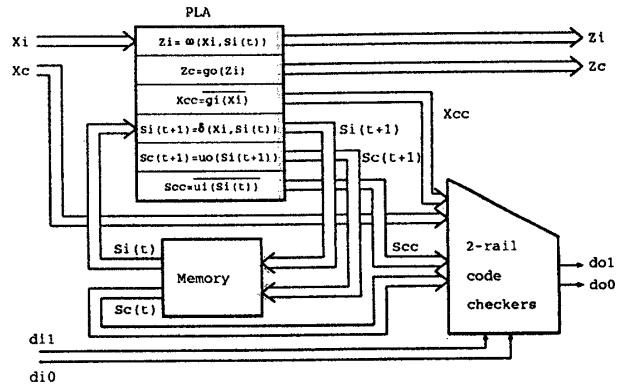


図1: SFS/SCD順序回路構成法

化され、添字の i は情報部を、 c は検査部を表す。入出力には、入出力のインターフェースを統一するために、更に1対の2線式符号(d_1, d_0)を付加する。関数 g_i を入力検査部生成関数、 g_o を出力検査部生成関数とすると、入力は $X_c = g_i(X_i)$ 、 $(d_{i1}, d_{i0}) = (1, 0) \text{ or } (0, 1)$ のとき、出力は $Z_c = g_o(Z_i)$ 、 $(d_{o1}, d_{o0}) = (1, 0) \text{ or } (0, 1)$ のとき、それぞれ符号語となる。また、関数 u_i を現状の検査部生成関数、 u_o を次状態の検査部生成関数とすると、現状は $S_c(t) = u_i(S_i(t))$ のとき、次状態は $S_c(t+1) = u_o(S_i(t+1))$ のとき、それぞれ符号語となる。

回路は、N-MOS NOT-NOR-NOR型のPLA、記憶回路、2線式符号チェッカから構成される。PLAは、入力情報部 X_i と現状情報部 S_i を入力として、次の6つの機能を実現する。

1. 出力情報部 $Z_i = \omega(X_i, S_i(t))$ の生成
2. 出力検査部 $Z_c = g_o(Z_i)$ の生成
3. 入力検査部のビット毎の否定 $X_{cc} = \overline{g_i(X_i)}$ の生成
4. 次状態情報部 $S_i(t+1) = \delta(X_i, S_i(t))$ の生成
5. 次状態検査部 $S_c(t+1) = u_o(S_i(t+1))$ の生成
6. 現状検査部のビット毎の否定 $S_{cc} = \overline{u_i(S_i(t))}$ の生成

記憶回路は、次状態 $S_i(t+1)$ 、 $S_c(t+1)$ を受けとり、次のクロックでそれらを現状として出力する。現状情報部 $S_i(t)$ はPLAに入力され、現状検査部 S_c は2線式符号チェッカに入力される。2線式符号チェッカは、2線式符号(X_c, X_{cc})、(S_c, S_{cc})、(d_{i1}, d_{i0})を受けとり、2線式符号(d_{o1}, d_{o0})を出力する。入力符号(X_c, X_{cc})、状態符号(S_c, S_{cc})は完全符号で符号化されているため、(d_{i1}, d_{i0})が入力符号、状態符号とそれぞれ独立ならば、2線式符号チェッカにはテストに必要な全パターンが入力される。

PLAの誤りモデルは、PLAの入力には全てのパターンが入力されるという条件のもとに成り立っていた。状態シンボルが 2^k ($k = 1, 2, \dots$)個のときは、状態情報部を k ビットで表現すれば、状態情報部空間に全てのパターンが現れる。よって、仮定2よりPLAの入力には全パターンが入力されることになる。一方、状態シンボルが 2^k ($k = 1, 2, \dots$)個ではないときは、状態情報部空間に現れないパターンが存在する。従って、仮定2を設けてもPLAの入力には全パターンが入力されない。このときは、次に述べる状態増加手続きを行ない、状態数を 2^k ($k = 1, 2, \dots$)個まで増加させ、PLAの入力に全パターンが現れるようにする。

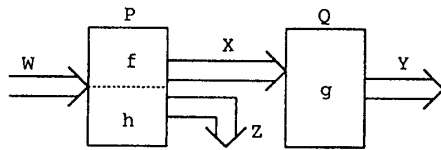


図2: モジュールの従属接続

2.3 状態増加手続き

入力、出力、状態シンボルはそれぞれ2進符号化されているものとする。 k_x ビット入力空間を X 、 k_z ビット出力空間を Z 、 k_s ビット状態空間を S で表す。入力ベクトル集合 I_X 、出力ベクトル集合 I_Z 、状態ベクトル集合 I_S 、出力関数 $\delta(I_X, I_S)$ 、状態遷移関数 $\omega(I_X, I_S)$ で表されるMealy型の順序機械 $M(I_X, I_Z, I_S, \delta, \omega)$ を考える。状態空間 S に全パターンが現れないとき、以下の手続きを行うことにより、動作は等価で状態数が1つだけ増加した別の順序機械 $M'(I_X, I_Z, I'_S, \delta', \omega')$ が得られる。

[状態増加手続き]

Step 0. I_S, I_X の全組み合わせに対して、 $\delta' = \delta, \omega' = \omega$ とする。

Step 1. $x \in I_X, s \in I_S$ に対して、 $s = \delta(x_i, s_j) = \delta(x_k, s_l) \in I_S, x_i \neq x_k \text{ or } s_j \neq s_l$ を満たす s を選ぶ。

Step 2. $s^+ \in S - I_S$ を選び、 I_S に加える。

Step 3. s^+ に対して、 $\delta'(x_m, s^+) = \delta(x_m, s)$ 、 $\omega'(x_m, s^+) = \omega(x_m, s)$ 、 $1 \leq m \leq 2^{k_x}$ を満たすように δ', ω' を定義する。

Step 4. $s = \delta(x_i, s_j)$ を満たす x_i, s_j に対して、 $s^+ = \delta'(x_i, s_j)$ を満たすように δ' を再定義する。

以上の手続きを状態空間 S に全パターンが現れるまで繰り返す。仮定2よりPLAの入力には全パターンが現れることになり、誤りモデルが適用できるようになる。そして、次の定理が成り立つ。

[定理]

入力、出力、状態が完全非順序分離符号で符号化され、かつ、2線式符号 (d_{i1}, d_{i0}) が入力符号、状態符号とそれぞれ独立ならば、図1に示した構成法は $F_{PLA} \cup F_{Memory} \cup F_{2-rail}$ に対してSFS/SCD同期式順序回路を実現する。

3 セルフチェック論理回路網の一構成法

SFS/SCD回路モジュールは、各モジュール間に符号チェッカを配置することなく従属接続をすることが出来る。従って、小規模な論理回路モジュールを構成要素とした大規模な回路網をセルフチェック化するとき、小規模な論理回路モジュールをまずセルフチェック化し、それらをただ単に接続するだけで大規模なセルフチェック回路網を構築することが出来る。論理回路モジュールが組合せ回路のときは[1]の構成法で、順序回路のときはここで提案した構成法で各モジュールをセルフチェック化することが出来る。しかし、回路網に組み込まれたモジュールは、前段のモジュールの実現する機能によっては、入力に全パターンが現れないことがある。すると、PLAの誤りモデルを用いることが出来なくなり、両構成法とも適用できなくなるという問題が生じる。

この問題を解決する一方法として、モジュールの機能を実現する関数の写像を変換し、回路網内部の全てのベクトル空間に全パターンが現れるようにする方法を提案する。これにより、各モジュールの入力には全パターンが現れることになり、PLAの誤りモデルを用いることが可能となる。以下に、その方法を述べる。

3.1 写像変換方法

図2に示すように、モジュールP,Qの従属接続を考える。モジュールPは X に対しては f を、 Z に対しては h を、モジュールQは g を実現する。モジュールPの k_w ビット入力空間を W 、 k_x ビット出力空間

を X 、モジュールQの k_y ビット出力空間を Y とする。 $k_w \geq k_x \geq k_y$ とし、 W には全パターンが現れるとする。このとき、 X に現れるベクトルの集合を I_X とする。目的は、 $Y = g(f(W)) = g'(f'(W))$ を満たし、ベクトル空間 X に全パターンが現れるような関数 f', g' を求めることである。以下に述べる空間ベクトル増加手続きを行なうことにより、1つだけ空間内のベクトルを増加させることが出来る。

[空間ベクトル増加手続き]

Step 0. $f' = f, g' = g$ とする。

Step 1. $x \in I_X, w \in W$ に対して、 $x = f(w_i) = f(w_j), w_i \neq w_j$ を満たす x を選ぶ。

Step 2. $x^+ \in X - I_X$ を選び、 I_X に加える。

Step 3. x^+ に対して、 $g'(x^+) = g(x)$ を満たすように g' を定義する。

Step 4. $x = f(w_i)$ を満たす w_i に対して、 $x^+ = f'(w_i)$ を満たすように f' を再定義する。

以上の手続きをベクトル空間 X に全パターンが現れるまで繰り返す。そして、回路網内部の全パターンが現れない全てのベクトル空間にこの手続きを適用すれば、各モジュールの入力には全パターンが現れるようになる。よって、PLAの誤りモデルを用いることが出来るので、SFS/SCD組合せ回路構成法、SFS/SCD順序回路構成法を適用することが出来る。ただし、空間ベクトル増加手続きは $k_w \geq k_x \geq k_y$ という仮定からわかるように、モジュールの入力空間が出力空間よりも小さいときは適用できない。そのような時は、モジュールの入力空間を拡張するために、モジュールに状態ビットを付加するという方法を用いればよい。その詳しい方法に関しては、ここでは述べない。

4 まとめ

本研究では、まずSFS/SCD同期式順序回路構成法を提案した。そして、セルフチェック論理回路網を構築する一方法として、小規模な論理回路モジュールをまずセルフチェック化し、それらを従属接続する方法を提案した。そして、各モジュールの入力には全パターンが現れるという仮定を満たすために、モジュールの機能を実現する関数の写像を変換する方法も提案した。これにより、複数のモジュールからなる回路網を簡単にセルフチェック化することが出来るようになる。現在、この構成法にもとづいたセルフチェック論理回路自動合成システムを開発中である。

本研究の一部は、科学研究費補助金(一般研究B:課題番号02452156)によるものである。

参考文献

- [1] S. Hatakenaka and T. Nanya. A design method of SFS and SCD combinational circuits. In *Proc. Pacific Rim Int. Symp. on Fault Tolerant Systems*, pp. 168-173, September 1991.
- [2] W. C. Carter and P. R. Schneider. Design of dynamically checked computers. In *Proc. IFIP '68*, pp. 878-883, August 1968.
- [3] W. K. Fuchs and J. A. Abraham. A unified approach to concurrent error detection. In *Proc. Int. Symp. on Fault-Tolerant Computing*, pp. 4-9, 1984.
- [4] 片山徳康, 南谷崇. PLAの故障モデルとSFS性に関する一考察. 電子情報通信学会技術研究報告, pp. 1-8, February 1990. FTS89-42.