

解説

VLSI のテスト容易化設計技術の研究動向†



樹 下 行 三†

1. は し が き

VLSI の集積度が増加し、その機能が複雑になるにしたがって、そのテストが非常に困難になり、VLSI の設計・製造の中で VLSI のテストの占める割合が急速に増大しつつある。ここで VLSI のテストとは、正しく設計された回路が VLSI の製造時または使用時に起こる故障の存在とその原因を見つけることをいい、設計そのものが正しいかどうかを調べる設計検証とはその立場を異にしている。すなわち、VLSI のテストにおいては、正しい回路設計が与えられているとして、製造された VLSI がその機能をもつかどうかを調べることを目的としている。しかし、実際上は、設計が正しいかどうか製造に依存する場合もあり、検証とテストを完全に分離して考えることは困難なようであるが、ここでは設計そのものは正しいとして、製造上・使用上に起こる故障に対するテストについて考える。

与えられた回路のテストを行うには、回路に適当な入力系列を印加し、それに対応して得られる出力系列が正しいかどうかを調べなければならない。テストのために用いられる入力系列をテストパターンという。テストパターンの作り方が不十分であれば、検出されない故障の存在を許すことになり、不完全なテストとなる。完全なテストを行うために、すなわちすべての故障を検出するためにどのようなテストパターンを用いればよいか、ということがテストにおける一つの基本課題でありテストパターン生成といわれている。

テストパターン生成の問題に関連して、故障モデルとテストパターンの制御点・観測点が重要な概念となる。物理的には多くの故障状態が考えられるので、完全なテストというときには、理論的にはある仮定され

た故障集合に対する完全なテストという意味になる。したがって、どのような故障集合、すなわち故障モデルを考えるかによって、テストパターンの生成が容易にも困難にもなる。簡単な故障モデルであればテストパターン生成は容易になるが、現実にかかる故障が含まれないことになる可能性がある。どのような故障モデルを考えるかということがテストパターン生成の鍵となっている。

故障モデルは、対象とする回路によって大きく変わる。ここでは論理 VLSI を対象としているので、論理動作ということが中心になる。したがって理論的には大前提として、故障によっても論理回路として動作すると考えている。これを論理故障という。しかし、これでは仮定の範囲が広いので、多くの場合には、論理素子の入出力信号線の値が 0 かまたは 1 に固定すると考える縮退故障などより具体的な故障モデルを対象として考えている。

MOS 回路においては、単に信号線での縮退故障だけでは扱えない故障もあり、トランジスタのスタックオープン故障、スタックオン故障などをも対象とする必要がある。PLA では、アレイ交点の開放故障や接続故障が主要な故障となるし、メモリにおいては、セルの縮退故障以外にパターン依存故障などをも考慮する必要がある。回路構成に応じて種々故障モデルを対象としなければならない。

テストパターンを印加する制御点は VLSI の外部入力端子であり、観測点は外部出力端子に限られている。一般に VLSI の外部端子の数は、内部素子数の平方根程度でしか増加しないので、集積度が増大するほど、制御・観測点数に対する素子数の比が増加し、テストがより困難となる方向に向かう。

いずれにしても、VLSI 化の傾向はテストを困難にする方向に作用し、テストのためになんらかの手段を講じなければならない。本来の回路の機能とは別に、テストパターン生成、故障検出、故障診断などテストに関することを考慮して設計することをテスト容易化

† Recent Development of Design for Testability Technologies for VLSI by Kozo KINOSHITA (Faculty of Integrated Arts and Sciences, Hiroshima University).

†† 広島大学総合科学部 現在大阪大学工学部応用物理学科

設計といっている。したがって、広い意味に解釈すれば、最近の VLSI はなんらかの意味でテスト容易化が行われているといえるが、ここでは明確な形で議論されている典型的なテスト容易化設計の手法¹¹⁻¹⁷とその最近の動向について解説する。

2. テスト容易化設計

本章ではテスト容易化設計に関する全般的な概念について述べる。組合せ回路のテスト容易化設計については 3. で、順序回路については 4. でより詳しく述べることにする。

テスト容易化設計の目標としては、

- A) テストパターン生成を容易にすること
- B) テストに要する時間を短くすること

をあげることができる。

この目標を達成するために、回路に付加的なハードウェアが用いられ、これが逆に VLSI チップのコストを高くする方向に作用する。コストに関連して評価すべき項目には、

- a) 素子数の増加
- b) 外部入出力ピンの増加
- c) 動作速度の遅れ

などが含まれる。

したがって、テスト容易化設計とは、a)～c) などの評価で増加するコストをできるだけ少なくするように A), B) の目標を達する設計手法であるといえる。しかし実際には、評価コストと目標のトレードオフをどのように関連づけるかという解析手段はなく、経験的な評価で行われている。

テスト容易化設計を実現する手法としては大きく、

- i) テストパターン生成を必要としない方法
- ii) 内部に制御・観測点を付加する方法
- iii) 回路構造の特殊性をいかして容易化する方法

などに分けることができる。

i) には、全数テスト¹⁸、万能テスト¹⁹といわれている方法が対応する。全数テストとは、入力端子群にすべての組合せ入力を印加する方法でテストができるように回路分割をしてテスト容易化を行う方法である。万能テストとは、回路構造が一定の場合、内部で実現される機能に関係なく、常に一定のテストパターンでテストを行う方法であり、一度テストパターンを求めておけば、以後テストパターンを生成する必要はない。いずれにしても、テストパターン生成に要するコストを最小化しようとする考え方である。

ii) の典型的な方法として、スキャン方式¹⁰⁻¹⁶がよく知られている。これは、順序回路のテストパターン生成を容易にするように、内部状態の値を直接観測・制御できるように付加端子を設ける方法である。一般にテスト容易化設計では、何かの方法で回路の内部情報を外部に引き出すことが考えられている。このとき、できるだけ少ない付加端子数で多くの内部情報を得る方法を考えなければならない。

iii) は、PLA やメモリのように、規則的な回路構造をもつ場合には、テストパターンも規則的に作ることが可能であるので、わずかの付加回路で多くの情報を制御・観測することができ、古くからカットポイントセルラアレイ¹⁷、半加算器アレイ^{18,19}、NORNAND セルラアレイ²⁰など多くの方法が提案されていた。最近では、PLA の実用化にともなって PLA のテスト容易化設計について多くの方法が提案されている²¹⁻³³。PLA のテスト容易化設計については、本特集の 2.1 で詳しく述べられるので本稿では省略する。メモリについても同様な考え方でテスト容易化設計を考えるとができ、テストパターン発生部をも付加回路内で行う、組み込みテスト型のテスト容易化設計手法も提案されている^{34,35}。最近では、これらの手法に基づいて作られた 16 M ビットチップの実用例も報告されている⁴⁰。メモリのテスト容易化設計については本特集の 2.2 で詳しく解説されるので本稿ではこれ以上の記述は省略する。

通常、テスト容易化設計は回路レベルで実現されたものに対して考えられているが、状態遷移図など、より高位な論理設計におけるテスト容易化設計をも考えることができる。たとえば、チェックンシーケンスとして知られている Hennie の方法⁴¹によるテストパターン生成法は状態遷移図に基づいた方法であり、具体的な回路には依存しない。このテストパターンの構成法から状態図レベルでのテスト容易化設計について

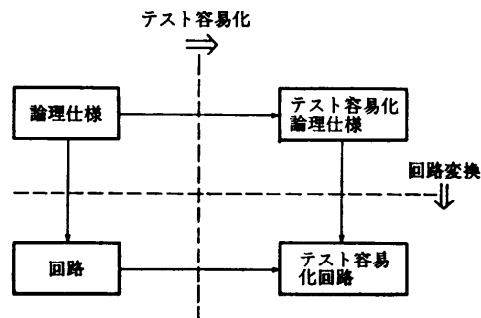


図-1 テスト容易化へのアプローチ

での提案がなされている^{42)~44)}。したがって、図-1 に示すようにテスト容易化を回路レベルで達成するか、論理レベルで達成するかの二つのアプローチが存在する。

最近では、マイクロプロセッサなど高機能の VLSI に対して、機能ブロックを用いたり^{45), 46)}、命令語を用いたり⁴⁷⁾、記号的なテストパターン生成手法^{48)~50)}など、ハイレベルなテストパターン生成手法が用いられるようになってきている⁵¹⁾。またマイクロプロセッサのコアとなる制御回路のテスト容易化設計の提案もなされている⁵²⁾。マイクロプロセッサのテスト容易化設計については、本特集 2.3 で述べられるので本稿では省略するが、ハイレベルなテストパターン生成ということからみて、ハイレベルのテスト容易化設計がこれからの中心課題の一つとなると考えられる。

テスト容易化の新しい動向として、電子ビームテストを用いるような非常に可観測性のよい環境でのテストパターン生成の問題がある^{53), 54)}。電子ビームテスト自身は単なるテストというよりは設計検証および診断用のツールと考えるべきであるが、回路の主要部が電子ビームテストで観測されやすいようにするという意味でのテスト容易化設計の問題が存在する。

さらに、可観測性のよいテスト容易化設計として、クロスチェック法が提案されている⁵⁵⁾。これは VLSI の多層配線技術を活用して、制御・観測用の信号線層を作ることによって、内部の可観測性、可制御性を向上させようという方法である。この手法が実用化されることになれば、テスト容易化としては大きな進展をみることになるが、制御点・観測点をどのようにして選ぶかという新たな問題が提起されることになる。

3. 組合せ回路のテスト容易化設計

本章では、組合せ回路のテスト容易化設計として知られている手法の特徴について述べる。

全数テストと回路分割^{56), 57)}：組合せ回路の出力は論理関数で記述されるので、すべての入力組合せに対して出力値が正しいかどうかを調べれば、すべての論理故障に対するテストを行うことができる。組合せ回路の場合には入力の順序は問題とならないので、カウンタなど適当な系列発生器を用いればテストパターンを容易に作ることができる。

一般に、 n 入力 m 出力の組合せ回路の場合、すべての出力が n 個の入力に依存していれば 2^n 個のテストパターンを用いなければならないが、各出力が入力

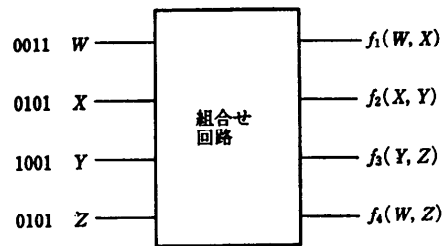


図-2 出力関数に応じた全数テスト

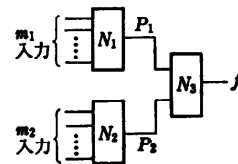


図-3 回路分割

の一部のみに依存する場合には 2^n より少ないパターンでテストのできる可能性がある。

たとえば、図-2 の回路のように各出力が 2 入力だけに依存する場合には、図に示すような長さ 4 のテストパターンですべての出力関数に対する入力組合せを尽くすることができる。単一出力としても入力数の多い場合には、図-3 のように回路分割を行う。このとき N_1 と N_2 の入力集合は必ずしも独立である必要はなく、重複した共通部分をもってよい。 N_1 , N_2 に対して全数テストパターンを求め、点 p_1 , p_2 に 0, 1 のすべての組合せが現れるように編成すればよい。一般に、 $2^{n_1} + 2^{n_2} \ll 2^{n_1+n_2}$ であるから、回路分割によってテストパターンを短くすることが可能である。

シンδροームテスト容易化設計⁵⁶⁾： n 変数論理関数 f の中の最小項の数を M とするとき、 $s(f) = M/2^n$ を関数 f のシンδροームという。 f を実現する組合せ回路に全数入力を与えたとき、出力に現れる 1 の数の割合がシンδροーム $s(f)$ である。 $s(f)$ の値を用いてテストを行うことをシンδροームテストといっている。

一般に、故障によって $s(f)$ の値が変わることが期待されるが、常にそうなるとは限らない。たとえば、図-4 に示す回路について考える。この回路の $s(f)$ は 10/16 である。信号線 A または B の値が 1(0) に縮退すると、 $s(f)$ は 12/16 (6/16) となりシンδροームテスト可能である。同様に、 E または H の値が 1(0) に縮退すると、 $s(f)$ は 13/16 (7/16) となりシンδροームテスト可能である。 F また G についても同様に 1(0) 縮退故障に対して $s(f)$ は 13/16 (7/16)

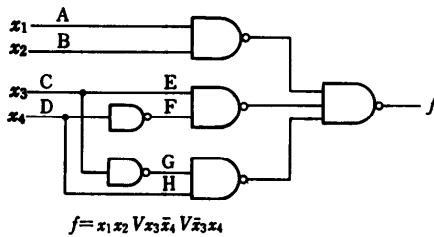


図-4 組合せ回路例

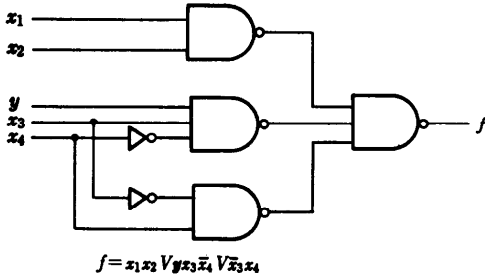


図-5 シンドロームテスト容易設計

となる。しかし、C または D の 1(0) 縮退故障に対しては、 $s(f)$ はともに 10/16 となり、正常値に等しいのでシンドロームテスト可能ではない。

シンドロームテスト可能でなく、付加入力を用いてシンドロームテスト可能にすることをシンドロームテスト容易化設計といっている。上に示した例では、図-5 に示すように付加入力 y を用いることにより、シンドロームテスト可能にすることができる。

シンドロームテスト容易化設計の詳しい方法⁵⁷⁾を述べるゆとりはないが、単一縮退故障に対して常にシンドロームテスト可能になるように組合せ回路を構成する方法が知られている⁵⁸⁾。また、すべての単一縮退故障と二つの線間のショート故障に対してシンドロームテスト可能になるような組合せ回路の構成法も知られている⁵⁹⁾。詳しくはそれぞれの文献を参照されたい。

全可観測な環境でのテスト容易化設計：テスト容易化設計の立場で、テストパターン生成をどこまで簡単にできるかを知る方法として、全可観測な環境でのテストということが考えられている。ここで全可観測な環境とは、すべての論理素子の出力の値が観測可能であるとする立場である。

簡単な例として図-6 の回路について説明する。2 入力 AND 素子の縮退故障は $S_A = \{011, 101, 110\}$ に属す二つの系列でテスト可能であり、2 入力 OR 素子の縮退故障は、 $S_O = \{100, 010, 001\}$ に属す二つの系列でテスト可能である。したがって、図-6 のように

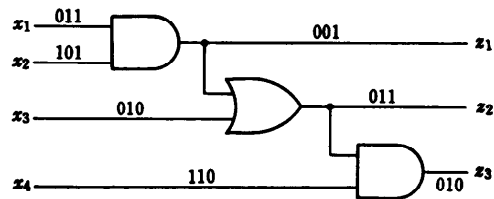


図-6 3 入力でテスト可能な回路例

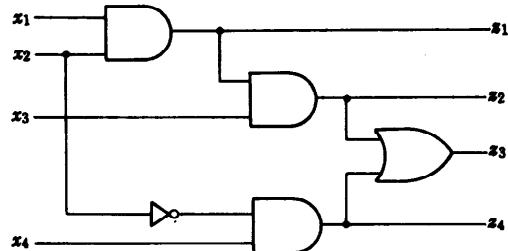


図-7 3 入力でテスト可能でない回路例

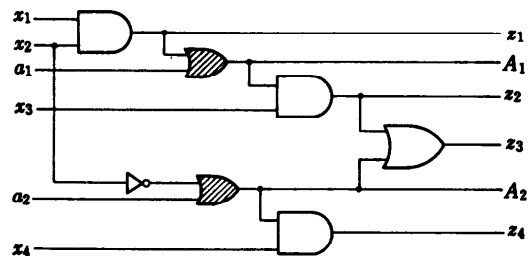


図-8 3 入力テスト可能な回路への変換

入力割当を行えば、長さ 3 の系列ですべての縮退故障がテスト可能となる。しかし、図-7 の回路の場合には 3 入力テスト可能ではない。このとき次のような方法で 3 入力テスト可能に変換することができる⁶⁰⁾。

1) 与えられた組合せ回路を 2 入力の AND 素子と OR 素子と NOT を用いて樹枝状回路として実現する。

2) 外部入力に $S_A \cup S_O$ の系列を割り当てる。各素子について割り当てられた入力系列がテスト可能ならばその系列を出力に伝搬させる。テスト可能でなければ、AND または OR 素子を用いて展開し、テスト可能にする。

3) すべての素子に系列が割り当てられるまで 2) を繰り返す。

たとえば、図-7 の回路は図-8 のように 3 入力テスト可能とすることができる。このとき、二つの素子と 2 出力が付加されている。しかし、テストパ

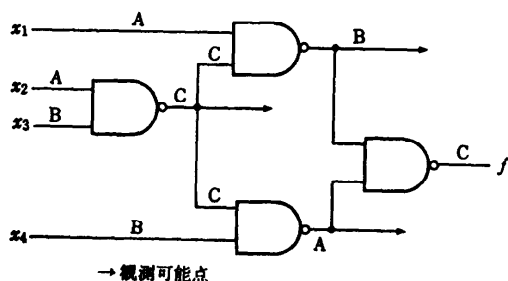


図-9 長さ7のテストパターンでテスト可能な回路

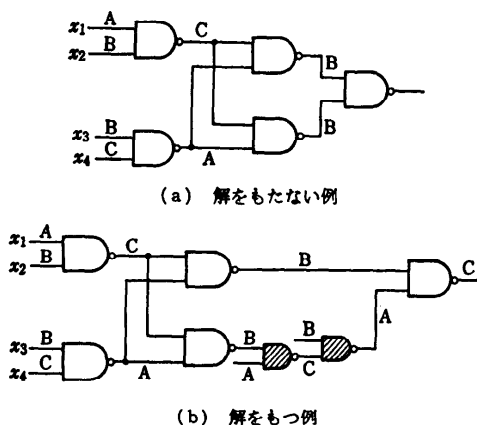


図-10 固定長テスト可能な回路への変換

ターンは、その長さを3より短くすることができないので、これは最小長のテストパターンである。

このように固定長のテストパターンを用いる方法は同一入力数をもつ NAND 素子のみで構成される組合せ回路へ拡張することができる。

2入力 NAND 素子の縮退故障およびスタックオープン故障は、 $S_N = \{A, B, C\}$ に属する二つの系列によってテスト可能である。ここで、 $A=1010111$ 、 $B=1101101$ 、 $C=0111010$ である。 S_N に属する任意の二つの系列の NAND 演算は残りの一つの系列となるので、 S_N は NAND 演算に関して閉じている。

図-9 に示す2入力 NAND 回路に対して、図中に示すように系列A, B, Cを割り当てることができ、各出力が可観測であるとすれば長さ7のテストパターンでテスト可能になる。しかし、一般には、図-10(a)に示すように三つの系列A, B, Cを矛盾なく割り当てる解をもたない例が存在する。このときには、図-10(b)のように2個の NAND 素子を付加することにより、回路の機能を変えずにテスト可能にするこ

とができる。

一般に、 k 入力 NAND 素子のみから構成される組合せ回路に対して、全可観測な環境では、固定的な長さ $k(k+1)+1$ のテストパターンでテスト可能のようにテスト容易化設計を行うことができる⁶⁴⁾。

通常の VLSI の設計においては、全可観測という仮定は現実的ではないが、電子ビームテスト^{61), 62)}を用いたり、クロスチェック法⁶⁶⁾を利用できる環境を考えれば、それ程非現実な仮定ではないと思える。

PLA のテスト容易化設計: PLA に関しては多く興味ある結果が知られている。これについては、本特集の 2.1 を参照されたい。

4. 順序回路のテスト容易化設計

順序回路のテスト容易化設計の手法は、状態図レベルで考えられているものと回路レベルで考えられているものに分けることができる。しかし、いずれも順序回路のテストパターンを求める困難さを軽減させる方法と考えられている。本章ではこれらの手法の概要について説明する。

状態図レベルのテスト容易化設計: 既約で強連結な状態遷移図が与えられたとき、その状態図を一意的に決める入出力系列(固有系列)を求める方法が Hen-nie により示されている⁴⁴⁾。故障によって状態図が変わると仮定すれば、固有系列をテストパターンの一部として用いることができる。

固有系列を求めるとき、状態図に区別系列¹⁾が含まれるかどうかにより、固有系列の長さが極端に異なる。ここで区別系列(distinguishing sequence)とは、その出力系列によって、初期状態を一意的に決めることのできる入力系列である。任意の状態図に常に区別系列が存在するとは限らないので、区別系列をもつように状態図を変更することが、順序回路に対する一つ

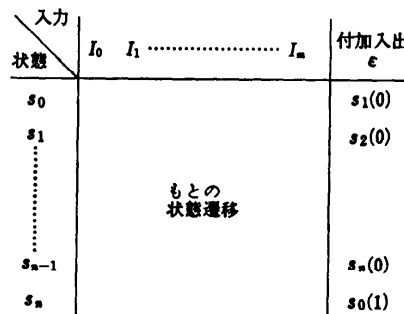


図-11 状態遷移の拡大

のテスト容易化設計となる。

このとき、何を付加して区別系列をもたせるかによって種々の方法が提案されている^{42)~44)}。たとえば、図-11に示すように、入力 ε を付加しこれにカウンタの機能をもたせると、 n 個の ε の系列が区別系列となる⁴²⁾。二つの入力 $\varepsilon_0, \varepsilon_1$ を付加すれば、区別系列と状態間遷移の長さが高々 $\log_2 n$ (n は状態数) になるように構成でき、固有系列の長さは、 $O(mn \log_2 n)$ となる⁴⁴⁾。ここで、 m は入力の数である。

完全な状態遷移表を求める困難さと、固有系列が長くなることから、この方法は実用化されていないが、最近のハイレベルな設計記述とテスト生成への傾向からみて、この考え方の利用価値があるように思える。

スキャン設計：外部入出力のみから順序回路のテストパターンを求めようとすると、未知の内部状態が存在することによって、組合せ回路のように簡単に求めることができない。したがって、図-12(a)に示すよ

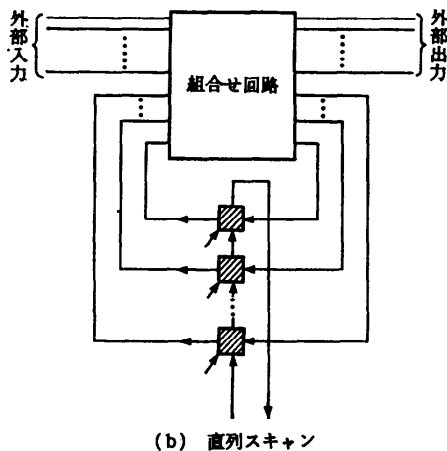
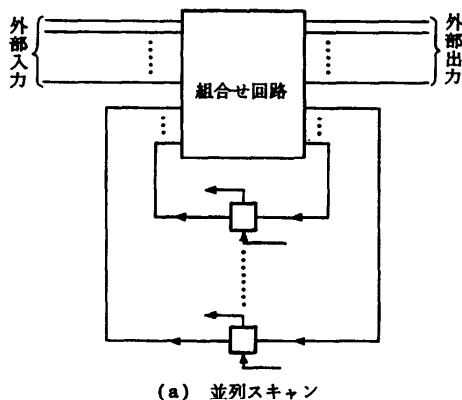


図-12 順序回路のスキャン設計

うに、内部状態を直接制御・観測できるようにすれば、順序回路に対するテストパターン生成の問題は、組合せ回路のそれと同等になる。

しかし、図-12(a)のような方法では付加入出力端子数が多くなりすぎるので、図-12(b)のようにシフトレジスタと通常のフリップフロップの動作が兼用できる回路を用いて付加入出力端子の数を少なくしている。このフリップフロップとシフトレジスタの機能をどのような回路で実現するかによって多くの回路が提案されている^{10)~16)}。各種のスキャン設計については、適用例を含めて本特集 3.1 で解説されるので、詳しくはそれにゆずることにしたい。

部分スキャン設計：スキャン設計を行った順序回路に対するテストパターンを求めるために、組合せ回路に対するテストパターン生成手法を用いることができる。しかし、回路規模が大きくなれば、組合せ回路に対しても常に 100% の故障検出率を達成するテストパターンが得られるとは限らない。したがって、実用的な観点からすれば、必ずしもすべてのフリップフロップをスキャンするのではなく、テストパターン生成において必要となったフリップフロップだけをスキャンできるようにしても検出率はそれほど悪くならないと考えられる⁶³⁾。

スキャン設計では、フリップフロップをスキャン可能にするために付加的なコストが必要となるので、この数を少なくすることにより、テスト容易性の性能をそれほど悪くしないで、コストを大幅に削減することができる。実験結果によれば⁶³⁾、351 個のフリップフロップを含む順序回路のスキャン設計における検出率が 98.75% であるのに対して、176 個のフリップフロップを含む部分スキャンでも 96.37% の検出率が得られると報告されている。すなわち、スキャン化率を 50% ぐらいにしても、全体の検出率をそれほど悪くしないでテストパターン生成が可能であることを示唆している。

以上、順序回路について、そのテスト容易化設計手法について述べた。これまでのところ、順序回路については、ほとんどの場合スキャン設計による方法を採用し、組合せ回路のテストパターン生成法が用いられ、順序回路を順序回路と見なすようなテストパターン生成はほとんど行われていなかった。

しかし、最近になって順序回路に対するテストパターン生成が活発になり、多くの手法が提案されているので^{64)~70)}、その実行結果により、順序回路に対す

るテストパターン生成のボトルネックがどこにあるかを知ることができるものと期待される。ISCAS '89においてベンチマーク用の順序回路が提供されたので⁷¹⁾、この分野への関心はさらに高まるものと期待している。順序回路に対する新しいテストパターン生成手法が提案されれば、それに応じた新しい順序回路のテスト容易化設計への道が拓けるであろうと考えている。

5. 組み込み自己テスト (BIST)

最近のメモリチップのように集積度が高くなると、テストパターンそのものが長くなり、一つのチップに要するテスト時間が重要な問題となる。テスト時間が重要視されるのは、テストに付随して用いられるテストのコストが高いため、量産を行うために多数のテストを導入しなければならない点にある。

VLSI 化にともなってテスト時間の増加が本質的なものであるとすれば、いかにテストコストを安くするかということを考えざるを得ない。テストコストを安くするという一つの考え方として、テスト機能をVLSI チップの中に組み込む方式、すなわち組み込み自己テスト (Built-In Self-Test: 以後 BIST と書く) が用いられるようになってきている。

一般に BIST は、図-13 に示すように、BIST の対象となる回路に、テストパターン発生部とテスト結果判定回路を付加して構成される。これらが BIST としてチップ内で実現されるためには、テストパターン発生部およびテスト結果判定回路が対象回路に比べて簡単に実現されなければならない。

テストパターン発生部としては、

- 1) 疑似ランダムパターン
- 2) 全数パターン
- 3) 規則的に発生されるパターン
- 4) マイクロコード生成によるパターン

などが用いられている。

疑似ランダムパターンは、線型フィードバックシフトレジスタ (Linear Feedback Shift Register: 以後 LFSR と書く) により容易に実現できるので⁴⁾、

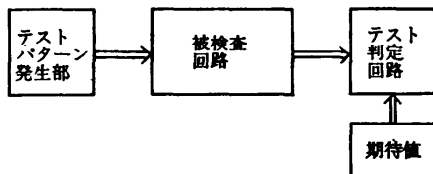


図-13 BIST の構成

BILBO⁷²⁾ を始めとし、メモリ³⁶⁾⁻³⁸⁾ やマイクロプロセッサ⁷³⁾ の BIST 化において多く用いられている。全数パターンは、疑似ランダムパターンと同様に用いることができるが、3. で述べたシンドロームテストの PLA への応用として、PLA の BIST において積極的に用いられている^{32), 33)}。

メモリは、その構造が規則的であることから、そのテストパターンも規則的に作ることができ、これをカウンタや簡単な系列発生器を用いて実現することができる。規則的なパターンは、マイクロコードを用いて発生させることも容易である^{37), 40)}。

テスト結果判定回路において、各出力値を基準値と比較する方法では、基準値の記憶容量が膨大になるので、出力系列を適当に圧縮してから基準値と比較することが必要になる。

簡単な圧縮関数としては、

$$C_1(R) = \sum_{i=1}^n r_i$$

$$C_2(R) = \sum_{i=1}^{n-1} \bar{r}_i \cdot r_{i+1}$$

$$C_3(R) = \sum_{i=1}^{n-1} r_i \cdot \bar{r}_{i+1}$$

などがよく用いられる。ここで、 $R = r_1 r_2 \dots r_n$ で、 r_i は 0 または 1 の 2 値をとる。

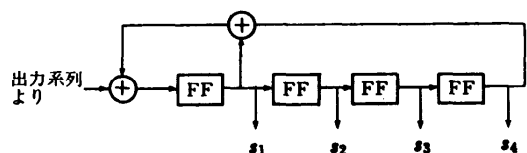
C_1 は 1 の数をかぞえる関数、 C_2 は 0 か 1 への変化の数をかぞえる関数、 C_3 は 1 か 0 への変化の数をかぞえる関数であり、これらは $\log_2 n$ ビット以下のカウンタで容易に実現できる。シンドロームテストでは C_1 が圧縮関数として用いられる。

いま、正常な場合の出力系列が

$$R = 00 \dots 011 \dots 1 = 0^h 1^k$$

であるとし、これを関数 C_1 、 C_2 、 C_3 を用いて圧縮すると、 $C_1(R) = k$ 、 $C_2(R) = 1$ 、 $C_3(R) = 0$ となる。逆に、長さ $n = h + k$ で、 $C_1 = k$ 、 $C_2 = 1$ 、 $C_3 = 0$ を満たす系列は、 $R = 0^h 1^k$ に限られるので、長さ $h + k$ の情報を $\log_2 k + 2$ ビットに圧縮することができ、基準値の記憶容量を大幅に削減することが可能となる。

圧縮関数としては、これ以外に LFSR が用いられ



る。この例を図-14 に示す。すなわち、出力系列の値を順次 LFSR の入力として加え、最後に残ったレジスタの値 (S_1, S_2, S_3, S_4) を圧縮値とする。この圧縮値のことをシグネチャとっている。LFSR によるシグネチャを用いた場合には、その値によってそれに加えられる系列を一意的に定めることができないので、誤まった出力系列に対しても正しいシグネチャを与えてしまう可能性がある。これの起こる確率を見逃し率とっている。

以上、BIST についての基本的な要求と構成方法について述べた。具体的な BIST の適用例については、本特集の以下の各項目の中で説明されるのでここではこれ以上の説明を省略するが、どのような故障モデルを対象としているか、テストパターン生成部がどのように構成されているか、どのような圧縮関数がいられ、どの程度の見逃し率があるかなどに注目すれば、その特徴を理解することは容易であろう。

6. あとがき

以上、VLSI のテスト容易化設計技術の動向として、これまでに確立された技術、いま問題になっている技術を中心に述べた。今後問題になるであろうと思われる方向についても若干の私見を加えたが、これからどのように動くかを予測することは容易ではない。

しかし、テスト容易化設計技術は、単にテストのための手法ではなく、高信頼化設計の中での一つの設計技術であると認識しなければならないことは事実であろう。

参 考 文 献

- 1) 樹下行三, 藤原秀雄: デジタル回路の故障診断(上), 4章, 工学図書 (1983).
- 2) 樹下行三, 浅田邦博, 唐津 修: VLSI の設計 II, 6章, 岩波講座マイクロエレクトロニクス, 岩波書店 (1985).
- 3) Fujiwara, H.: Logic Testing and Design for Testability, Chapters 5~9, The MIT Press (1985).
- 4) Bardell, P.H., McAnney, W.H. and Savir, J.: Built-In Test for VLSI, Wiley Interscience (1987).
- 5) McCluskey, E.J.: Logic Design Principles, with Emphasis on Testable Semicustom Circuits, Chapter 10, Prentice Hall (1986).
- 6) Agrawal, V.D. and Seth, C.S.: Test Generation for VLSI Chips, Chapters 5, 6, IEEE Computer Society Press (1988).
- 7) Williams, T.W. and Parker, K.P.: Design for Testability-A Survey, Proc. of IEEE, Vol. 71, No. 1, pp. 311-325 (1983).
- 8) McCluskey, E.J.: Verification Testing-A Pseudoexhaustive Test Technique, IEEE Trans. Comput., Vol. C-33, No. 6, pp. 541-546 (1984).
- 9) Fujiwara, H. and Kinoshita, K.: A Design of Programmable Logic Arrays with Universal Tests, IEEE Trans. Comput., Vol. C-30, No. 11, pp. 823-838 (1981).
- 10) Funatsu, S., Wakatsuki, N. and Arima, T.: Test Generation Systems in Japan, Proc. 12th DAC, pp. 114-122 (1975).
- 11) William, N.J.Y. and Angell, J.B.: Enhancing Testability of Large Scale Integrated Circuits Via Test Point and Additional Logic, IEEE Trans. Comput., Vol. C-22, No. 1, pp. 46-66 (1973).
- 12) Toth, A. and Holt, C.: Automated Data Base Driven Digital Testing, IEEE Computer, pp. 13-19 (1974).
- 13) Eichelberger, E.B. and Williams, T.W.: A Logic Design Structure for LSI Testability, Proc. 14th DAC, pp. 462-468 (1977).
- 14) Yamada, A. et al.: Automatic System Level Test Generation and Fault Location for Large Digital Systems, Proc. 15th DAC, pp. 347-352 (1978).
- 15) Stewart, J.H.: Application of Scan/Set for Error Detection and Diagnostics, Proc. Test Conference, pp. 152-158 (1978).
- 16) Ando, H.: Testing VLSI with Random Access Scan, COMPCON 80, pp. 50-52 (1980).
- 17) Minnick, R.C.: Cutpoint Cellular Logic, IEEE Trans. Comput., Vol. EC-13, No. 6 (1964).
- 18) 岡本 務, 樹下行三: 半加算器アレイの故障検査について, 信学論C, Vol. 54-C, No. 5, pp. 362-369 (1971).
- 19) 松田 潤, 樹下行三: 半加算器アレイの故障検査の一般化について, 信学論D, Vol. 56 D, No. 3, pp. 140-145 (1973).
- 20) 高松雄三, 樹下行三: NORNAND セルラアレイの最小故障検査入力に関する一考察, 信学論D, Vol. 60 D, No. 9, pp. 734-741 (1977).
- 21) Fujiwara, H., Kinoshita, K. and Ozaki, H.: Universal Testsets for Programmable Logic Arrays, Proc. FTCS-10, pp. 137-142 (1980).
- 22) Hong, S.J. and Ostapko, D.L.: FITPLA: A Programmable Logic Arrays for Function Independent Testing, Proc. FTCS-10, pp. 131-136 (1980).
- 23) Son, K. and Pradhan, D.K.: Design of Programmable Logic Array for Testability, Proc. ITC, pp. 163-166 (1980).
- 24) Pradhan, D.K. and Son, K.: The Effect of

- Untestable Faults in PLAs and a Design for Testability, Proc. ITC, pp. 359-367 (1980).
- 25) Saluja, K. K., Kinoshita, K. and Fujiwara, H.: A Multiple Fault Testable Design of Programmable Logic Arrays, Proc. FTCS-11, pp. 137-142 (1981).
 - 26) Saluja, K. K., Kinoshita, K. and Fujiwara, H.: An Easily Testable Design of Programmable Logic Arrays for Multiple Faults, IEEE Trans. Comput., Vol. C-32, No. 11, pp. 137-142 (1983).
 - 27) Yajima, S. and Aramaki, T.: Autonomously Testable Programmable Logic Arrays, Proc. FTCS-11, pp. 41-43 (1981).
 - 28) Ramanatha, K. S. and Biswas, N. N.: A Design for Complete Testability of Programmable Logic Array, Proc. ITC, pp. 67-74 (1982).
 - 29) Ramanatha, K. S. and Biswas, N. N.: A Design for Testability of Undetectable Crosspoint Faults in Programmable Logic Arrays, IEEE Trans. Comput., Vol. C-32, No. 6, pp. 551-557 (1983).
 - 30) Khakbaz, J.: A Testable PLA Design with Low Overhead and High Fault Coverage, Proc. FTCS-13, pp. 426-429 (1983).
 - 31) Fujiwara, H.: A New PLA Design for Universal Testability, IEEE Trans. Comput., Vol. C-33, No. 8, pp. 745-750 (1984).
 - 32) 山田輝彦: シンドロームテスト容易な PLA の一構成法, 信学論 D, Vol. 66-D, No. 8, pp. 932-938 (1983).
 - 33) 山田輝彦: 大規模 PLA のシンドロームテスト容易な設計とシンドロームの計算法, 信学論 D, Vol. 68-D, No. 5 (1985).
 - 34) Kinoshita, K. and Saluja, K. K.: Built-in Testing of Memory Using On-Chip Compact Testing Scheme, Proc. ITC, pp. 271-281 (1984).
 - 35) Saluja, K. K., Boswell, C. and Kinoshita, K.: A Design of Parallel Testable Programmable Logic Arrays, Proc. ISCAS, pp. 1325-1328 (1985).
 - 36) Kinoshita, K. and Saluja, K.: Built-in Testing of Memory Using an On-Chip Compact Testing Scheme, IEEE Trans. Comput., Vol. C-35, No. 10, pp. 862-870 (1986).
 - 37) Saluja, K. K., Sng, S. H. and Kinoshita, K.: Built-in Self-Testing RAM: A Practical Alternative, IEEE Design and Test, Vol. 4, No. 1, pp. 42-51 (1987).
 - 38) Franklin, M., Saluja, K. K. and Kinoshita, K.: Row/Column Pattern Sensitive Fault Detection in RAMs Via Built-in Self-Test, Proc. FTCS-19, pp. 36-43 (1989).
 - 39) Jarwala, N. T. and Pradhan, D. K.: TRAM: A Design Methodology for High-Performance, Easily Testable, Multimegabit RAM's, IEEE Trans. Comput., Vol. 37, No. 10, pp. 1235-1250.
 - 40) Takeshima, T., Takada, M., Koike, H. and Watanabe, H. et al.: A 55 ns 16 Mb DRAM, Digest of Technical Papers, ISSCC, pp. 246-247+p. 353 (1989).
 - 41) Hennie, F. C.: Fault Detecting Experiments for Sequential Circuits, Proc. Switching Theory and Logical Design, Princeton (1964).
 - 42) Murakami, S., Kinoshita, K. and Ozaki, H.: Sequential Machines Capable of Fault Diagnosis, IEEE Trans. Comput., Vol. C-19, No. 11, pp. 1079-1083 (1970).
 - 43) Kohavi, I. and Kohavi, Z.: Variable Length Distinguishing Sequences and Their Application to the Design of Fault Detecting Experiments, IEEE Trans. Comput., Vol. C-17, No. 8 (1968).
 - 44) Fujiwara, H., Nago, Y., Sasao, T. and Kinoshita, K.: Easily Testable Sequential Machines Using Extra Inputs, IEEE Trans. Comput., Vol. C-24, No. 8, pp. 821-826 (1975).
 - 45) Somenzi, F. et al.: Testing Strategy and Technique for Macro-Based Circuits, IEEE Trans. Comput., Vol. C-34, No. 1, pp. 85-90 (1985).
 - 46) Chandra, S. J. and Patel, J. H.: A Hierarchical Approach to Test Vector Generation, Proc. DAC, pp. 495-501 (1987).
 - 47) Thatte, S. M. and Abraham, J. A.: Test Generation for Microprocessors, IEEE Trans. Comput., Vol. C-29, No. 6, pp. 429-441 (1980).
 - 48) Lin, T. and Su, S. Y. H.: Functional Test Generation of Digital LSI/VLSI Systems Using Machine Symbolic Execution Technique, Proc. ITC, pp. 660-668 (1984).
 - 49) Murray, B. T. and Hayes, J. P.: Hierarchical Test Generation Using Precomputed Test for Modules, Proc. ITC, pp. 221-229 (1988).
 - 50) Lai, K.-W. and Siewiorek, D. P.: Functional Testing of Digital Systems, Proc. DAC, pp. 207-213 (1983).
 - 51) Bhattacharya, D., Murray, B. T. and Hayes, J. P.: High-Level Test Generation on for VLSI, IEEE Computer, Vol. 22, No. 4, pp. 16-24 (1989).
 - 52) 岡本卓爾, 大西伸一, 樹下行三: ランダムロジックで構成したプロセッサ制御回路の検査容易設計, 信学論 D, Vol. J70-D, No. 7, pp. 1294-1303 (1987).
 - 53) 梶原誠司, 温曉晴, 板崎徳禎, 樹下行三: 可観測な環境でのテストパターン生成について, 情報処理学会設計自動化研究会, 44-5, pp. 35-42 (1988).
 - 54) 樹下行三, 温曉晴, レディ, S. M.: 可観測な環境での NAND 論理回路の故障診断, 電子情報通信学会フォールトトレラントシステム研究会 FTS-

- 88-33, pp. 41-48 (1988).
- 55) Gheewala, T.: Cross-Check: A Cell Based VLSI Testability Solution, Proc. DAC, pp. 706-709 (1989).
- 56) Savir, J.: Syndrome-Testable Design of Combinational Circuits, IEEE Trans. Comput., Vol. C-29, No. 6, pp. 442-451 (1980).
- 57) Savir, J.: Syndrome Testing of Syndrome-Untestable Combinational Circuits, IEEE Trans. Comput., Vol. C-30, No. 8, pp. 606-608 (1981).
- 58) Barzilai, Z., Savir, J., Markowsky, G. and Smith, M.G.: The Weighted Syndrome Sums Approach to USI Testing, IEEE Trans. Comput., Vol. C-30, No. 12, pp. 996-1000 (1981).
- 59) Bhattacharya, B.B. and Gupta B.: Syndrome Testable Design of Combinational Networks for Detecting Stuck-at and Bridging Faults, Proc. ITC, pp. 446-452 (1983).
- 60) Saluja, K.K. and Reddy, S.M.: On Minimally Testable Logic Networks, IEEE Trans. Comput., Vol. C-23, No. 1, pp. 552-554 (1974).
- 61) Tamama, A. and Kuji, N.: Integrating an Electron-Beam System into VLSI Fault Diagnosis, IEEE Design and Test, Vol. 3, No. 2, pp. 23-29 (1986).
- 62) Wolfgang, E.: Electron Beam Testing, in Handbook of Advanced Semiconductor Technology and Computer Systems (Ed. G. Rabaat), Norstand Reinhold Co. (1987).
- 63) Agrawal, V.D., Cheng, K.-T., Johnson, D.D. and Lin, T.: Designing Circuits with Partial Scan, IEEE Design and Test, Vol. 21, No. 4, pp. 8-15 (1989).
- 64) Bryant, R.B. and Cho, K.: Test Pattern Generation for Sequential MCS Circuits by Symbolic Fault Simulation, Proc. DAC, pp. 418-423 (1989).
- 65) Cheng, W.-T. and Yu, M.-L.: Differential Fault Simulation-A Fast Method Using Minimal Memory, Proc. DAC, pp. 424-428 (1989).
- 66) Cheng, K.-T. and Agrawal, V.D.: A Concurrent Test Generation and Design for Testability, Proc. ISCAS, pp. 1935-1938 (1989).
- 67) Cheng, W.-T. and Davidson, S.: Sequential Circuit Test Generator (STG) Benchmark Results, Proc. ISCAS, pp. 1939-1941 (1989).
- 68) Hayashi, T., Hatayama, K., Ishiyama, S. and Takakura, M.: Two Test Generation Methods for Sequential Circuits, Proc. ISCAS, pp. 1942-1945 (1989).
- 69) Liou, A., Montessoro and Gai, S.: The Complexity of Sequential ATPG, Proc. ISCAS, pp. 1946-1949 (1989).
- 70) Marlett, R.: Evolution of an Effective DFT/ATG Solution for Sequential ASICs, Proc. ISCAS, pp. 1950-1953 (1989).
- 71) Brglez, F., Bryan, D. and Kozminski, K.: Combinational Profiles of Sequential Benchmark Circuits, Proc. ISCAS, pp. 1929-1934 (1989).
- 72) Könemann, B., Mucha, J. and Zwihehoff, G.: Built-in Logic Block Observation Techniques, Proc. ITC, pp. 37-41 (1979).
- 73) Daniels, R.G. and Bruce, W.C.: Built-in Self-Test Trends in Motorola Microprocessors, IEEE Design and Test, pp. 64-71 (1985).

(平成元年8月8日受付)