

システム オンシリコン時代のテスト容易化設計

Test Design Methodologies for System-on-Silicon LSIs

安藏 順一
K. Anzou

上戸鎖 裕史
H. Kamitokusari

大規模なシステム LSI の設計では、既存の設計資産を再利用したり、第三者から購入した機能ブロック (IP : Intellectual Property) を組み込んでシステムを構築するようになる。このようなシステム LSI のテストでは、それぞれのブロックのテスト設計を効果的に用いて全体のテストを行わなければならない。そのためには、各ブロックのテスト設計が十分になされていることはもちろん、それらのテスト設計手法が全体のテスト設計戦略に組み入れやすくなっていることが必要である。

当社は、システム LSI の各機能ブロックに適用されるテスト容易化設計 (DFT : Design For Testability) 手法をサポートするツールを開発してきた。また、各手法を統合して全体を効率的にテストするためのテスト設計環境を構築している。

Large system LSI designs contain several types of functional blocks. Some blocks are internally developed and reused, while others are offered as an intellectual property (IP) core by other companies. For testing such system LSIs, the test designs of each block must be effectively integrated for the overall chip test.

Toshiba has developed tools to support the test design methodologies applied to each functional block. We are also developing a test design environment for testing the overall system LSI.

1 まえがき

システム LSI は複数の異なる種類の機能ブロックから構成される。そのうちのある部分は、既存の設計資産を再利用したり、第三者から購入したものであったりする。システム LSI をテストするためには、各ブロックに対して異なるテスト手法を適用するとともに、それらを効率的に統合できなければならない。例えば、スキャン設計された論理ブロックがシステム LSI の一部としてあった場合、このテスト回路を使ってどのように外部からブロックをテストするかを考える必要がある（図 1）。

このようなことから、システム LSI のテスト容易化設計は従来の LSI に比べて多様かつ複雑になり、設計の後工程ですべてを行うことは困難になる。システム設計の段階で効率的なテスト戦略を立て、面積増加やテスト設計時間、テストでのテスト時間をより早い段階で見積もることが今後は重要になる。これにより、設計全体においてテスト容易化設計の占める割合を大幅に削減することができる。

ここでは、システム LSI の各機能ブロックに適用されるテスト容易化設計手法と、それらを統合する方法について説明するとともに、当社におけるテスト設計環境構築への取組み状況について述べる。また、実例の一つとして、RISC (縮小命令セットコンピュータ) プロセッサコアのキャッシュテスト用に組み込んだ自己テスト回路についても解説する。

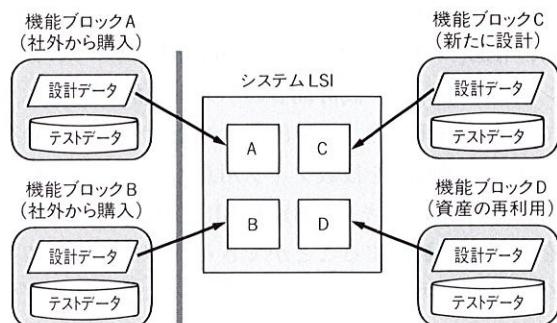


図1. システム LSI のテスト 各ブロックのテスト情報を用いて、システム LSI のテストを行なう手法を確立することは重要である。
Testing of system LSI

2 機能ブロックのテスト設計手法

2.1 スキャン設計

スキャン設計は、論理ブロックに対するもっとも一般的なテスト設計手法である。回路内の記憶素子 (フリップフロップ) をスキャンセルに置き換えてシフトレジスタとして接続し、外部から直接制御・観測できるようにする。

このうちすべての記憶素子をスキャンセルに置き換えるフルスキャン設計は自動化しやすく、またテストパターンの自動生成も簡単になる。このため、チップの面積増加は比較的大きいものの、特に time-to-market (市場への即応性) が重要となる ASIC (用途特定 IC) の分野では、多くの設計

で取り入れられてきた。

一方、記憶素子の一部をスキャンセルに置き換えるパーシャルスキャン設計では、置き換える割合が小さいほど面積や性能への影響は少なくてすむ。しかし、一方でテストパターンの生成は困難になり、高品質のテストパターンを作成するためには、置換の割合や置き換えるセルの選択を最適化する必要がある。これには設計のくり返しにより最適なものを選ぶ試行錯誤的な作業が必要で、フルスキャンに比べ設計時間がかかる。

ツールの性能向上により、パーシャルスキャンの採用の割合が増えると一時は予想された。しかし、実際はLSIの高集積化によりサイズ増加が大きな問題ではなくなりつつある。特に、システムLSIでは論理部がチップ全体に占める割合は相対的に小さくなり、この傾向はより大きくなる。このため、フルスキャン設計は引き続き主流になっていく。

2.2 バウンダリスキャン

バウンダリスキャン手法は、もともとボード上のチップ間配線のテストを容易に行うための技術としてIEEE(米国の電気電子学会)で標準化されたものである(IEEE 1149.1)。これはLSIの入出力信号部にバウンダリスキャンセルという特殊なセルでシフトレジスタを構成し、これを制御回路から4~5本の外部端子で直接制御・観測できるようにしたものである(図2)。

一方、IEEEの標準仕様ではこの制御回路とバウンダリスキャン回路を用いて、設計固有のテスト機能の拡張を可能にしている。当社では、標準的なテスト命令および固有命令を備えた制御回路を複数タイプ用意している。これらを用いてチップ内部のスキャン回路やBIST(Built-In Self Test)回路を外部から制御することができる。バウンダリスキャン回路を用いてテストを行えば、多ピンLSIのチップテストを高価な多ピン用テスタを用いることなく行うことができる。また、プロセッサコアなどの機能ブロック自身にバウンダリスキャン回路を構成し、外部からのテストやデバッグに利用することもできる。

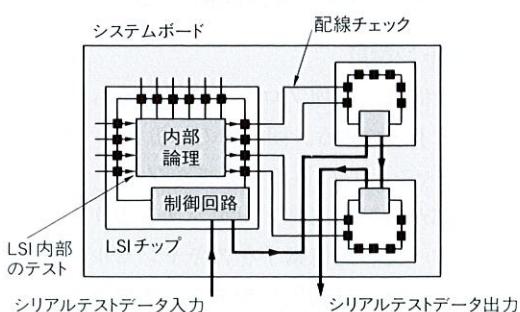


図2. バウンダリスキャン手法 ボードの配線チェックだけでなく、LSI内部のテストを制御回路を用いて行うことができる。

Boundary-scan architecture

2.3 メモリBIST

BISTはチップ内のデバイスへのテストパターンの生成・入力と出力結果の解析を、デバイス周辺に構成された論理回路により自動的に行うテスト手法である(図3)。手法そのものは従来からのものであるが、機能ブロックベースの設計が一般的になるにつれて重要度が増している。特に埋込みメモリの数やサイズが増大すると、従来のようなマルチプレクサ切出しによる直接テストの手法では回路変更が非常に煩雑な作業となる。また、個々のメモリのテストパターンを順次与えることによるテスタメモリの消費やテスト時間の増加も問題になる。

BISTによるメモリのテストは、これらの問題点を解決する。まず、テストに必要な外部信号が少なく、回路変更がしやすい。テストパターンは外部的に用意する必要がなく、テスタメモリを消費しない。また、テストパターンの生成と結果解析を回路内部で行うため、高速なテストが可能になる。これらのことから高価なテスタを用いる必要がなくなり、テストコストを大幅に削減できる。特に、比較的簡単なアルゴリズム的なテストパターンでテストが可能なスタティックRAMでは、BIST論理もメモリの構成に合わせて構成しやすく、BISTによるテストが一般的になりつつある。

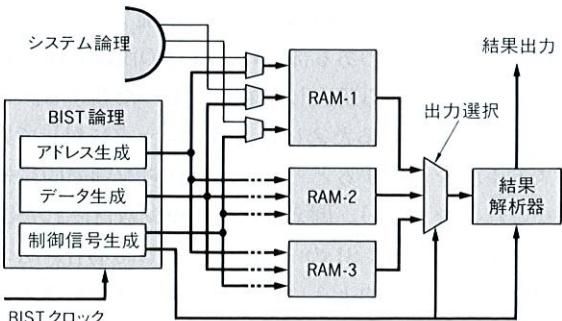


図3. メモリBIST手法 当社のBIST生成ツールでは、複数のRAMに対して共通のBIST制御回路を生成する。

Memory BIST (built-in self test) architecture

2.4 ロジックBIST

ロジックBISTは、メモリBISTと同様に設計内の論理回路部分のテスト生成と結果解析を内部のテスト回路により行う。これによりテスタメモリの消費を削減できる。

ロジックBISTで問題になるのは、内部で生成されるテストパターンが乱数的であり、高い故障検出率を得ることが難しいことである。このため、図4のように内部のレジスタをスキャン化して直接制御・観測する手法が必要となる。なお、ここではコアの入出力信号もバウンダリスキャンレジスタで制御・観測している。通常は、フルスキャンによ

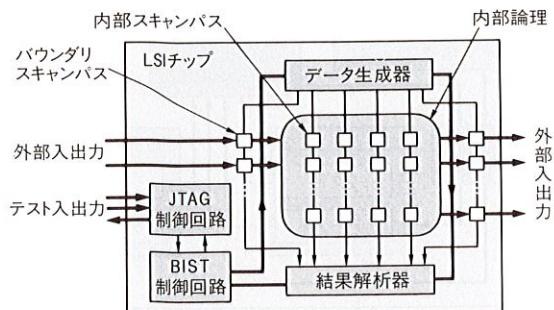


図4. ロジック BIST 手法 フルスキャンとバウンダリスキャン回路を用いて組合せ論理部のテストを行う方式がもっとも容易である。

Logic BIST architecture with boundary scan

っても高い故障検出率を達成するのは難しく、故障を検出しにくい部分にテスト用の回路を追加していくなどの作業が必要となる。この作業は、テスト設計時間を引き延ばすことになるが、ロジック BIST においては避けられない。

このように、ロジック BIST はテスト設計者にとって煩雑なものといえる。しかし、一方ではテスト用の信号が少なくて済み、コアとともに初期設定やクロック数、テスト結果の期待値といったテスト仕様だけが提供されればテストでき、テストパターンの管理や編集の作業が簡単である。つまり、コアを購入するユーザにとっては、システムのテスト設計に組み込みやすいテスト手法であり、コアとしての付加価値が高いといえる。

3 システム LSI のテスト設計手法

3.1 テストの統合化

以上のように、システム LSI 内部の各機能ブロックに対しては、それらの種類に応じてテスト容易化設計の手法が存在する。システム LSI のテストではこれらを統合し、効率的にテストできる仕組みが必要となる。従来手法の延長としては、ブロック間の接続中にマルチプレクサを挿入して外部から直接テストする方法がある。また、システムバスをテスト専用に使えるテストモードを設け、直接テストできるようにもできる（テスト専用バスは面積コストが大きい）。いずれもシステム LSI チップ単体をテストするには十分であるが、チップがボード上にある状態で検査を行うことも考慮すると、IEEE 1149.1 準拠の制御回路とバウンダリスキャン回路をテスト制御のために使うのが、一つの効果的な方法である。

図5は、IEEE 1149.1 準拠の制御回路を使ったシステム LSI のテスト設計の例である。制御回路はバウンダリスキャン回路を制御するほか、論理部のスキャンバスや BIST 回路を制御する。各ブロックのテストパターンは編集され、制御回路を通じて各ブロックをテストするのに用いられる。

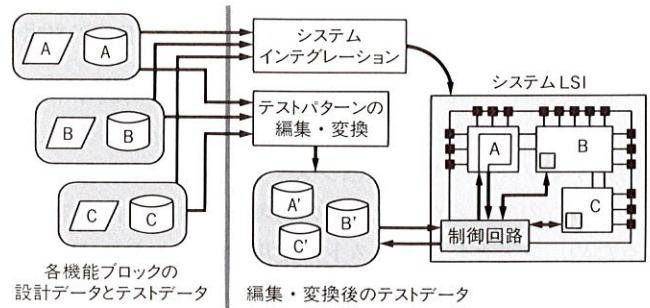


図5. システム LSI の統合化されたテスト 編集・変換された各ブロックのテストパターンを、IEEE 1149.1 制御回路を通じて入力する。
Integrated testing of system LSI

当然このときに、各機能ブロックに導入されているテスト手法について熟知している必要がある。コアが BIST 設計されていればテストの適用は非常に簡単になる。テストパターンデータがコアとともに提供されている場合は、チップの外部ピンから適用できるように編集したり、バウンダリスキャンを用いる場合はシリアルなパターンに変換する作業が必要となる。機能ブロックに十分なテスト設計がなされていない場合は、ユーザが自分でテスト回路を附加する必要があるが、回路変更のできないハード的な IP コアではこれが不可能となる。

このように、統合しやすいテスト回路をもち、そのインターフェース情報を提供することは、IP コアの付加価値を高めるために非常に重要な要因となる。

3.2 テスト設計環境

システム LSI のテスト容易化設計では、各機能ブロックのテスト設計を自動化するツールや、それらのテストパターンを編集・変換するツールなどが必要になる。当社ではスキャン設計やメモリ BIST 回路・バウンダリスキャン回路の生成のためのツールを用意しており、引き続きテスト設計環境の整備を行っている。特に、BIST については有効な手法として整備を推し進めていく。

将来の発展の方向としては、トップダウン設計スタイルの普及に伴い、RTL (Register Transfer Level) におけるテスト設計環境の構築が必要になってくる。RTLによる設計と、自動論理合成・最適化の繰返しによる設計サイクルに従来のゲートレベルのテスト設計ツールを導入しようとすると、RTL 記述の修正のたびにツールを適用する必要がある。RTL 記述の段階でテスト設計を行っておけば、ツールの適用は最初の一回だけに済む。BIST 制御回路やバウンダリスキャン制御回路はそれ自体を RTL 記述で用意あるいは生成し、デザインに組み込んでおくことが一般的である。また、RTL の段階でテスト容易性を解析しておき問題箇所を修正しておくことも、テスト設計期間の短縮には効果的である。これら RTL での設計環境についても整備していく。

4 RISC プロセッサ コアのキャッシュ BIST 設計

ここでは、当社の RISC プロセッサ コアの内蔵キャッシュのテスト用に設計した BIST 回路について述べる。ここで紹介するプロセッサ コアはシステム ASIC コアとして利用されるものであるが、テストの物理的なメモリは有限であるため、コアとしてのテストに使用するテストのメモリは少ないほうが望ましい。これに加え、実スピードのテストを行うためや故障解析が簡単にできるようにするために、以下に紹介する BIST 回路を設計して適用した。

図 6 に BIST 回路の概要を示す。テスト対象となるのは RISC プロセッサ の命令キャッシュとデータキャッシュであり、それぞれがデータ部とタグ部からなる。BIST 回路はこれら 4 種類のメモリアレーをそれぞれ個別にテストでき、テストアルゴリズムも Marching や Checkerboard など複数種類を外部から選択できる。これらによりキャッシュメモリに想定される故障のほとんどを検出できる。コアに内蔵されるキャッシュは複数のサイズが用意されているが、BIST 回路はそのサイズを自動的に認識してサイズに応じたテストを実行するので、回路は共通に用いることができる。

この BIST 回路の動作を次に説明する。テスト用端子(a)でキャッシュテストモードが設定されると、テスト制御回路(b)から BIST テスト用のクロック信号、BIST 用イネーブル信号、およびテスト対象のメモリアレーやテストアルゴリズムを選択する信号などが供給されて、BIST によるキャッシュテストが可能な状態になる。BIST 制御回路(c)は BIST 用クロックに同期してキャッシュメモリ用の入力データ、入力アドレスその他の制御信号を生成し、選択されたアルゴリズムに従いキャッシュのテストを実行する。

キャッシュからのテスト結果の出力データは、BIST 回路でテスト対象となったデータラインを選択した後、バスを介して外部端子にパラレルに出力される。ここでキャッシュからの出力データを比較や圧縮するのではなくパラレルに出力しているのは、故障解析を効率的に行うためである。

また、BIST 回路自体はスキヤン設計されており、ほぼ 100 % の故障検出率でテストできる。同時に、BIST 回路はキャッシュをスルーするモードをもち、故障がキャッシュ側にあるのか、BIST 側にあるのかを切り分けやすい。

BIST 回路のサイズは、プロセッサ コア論理部の 2 % 程度と非常に小規模である。BIST 回路の導入により実スピード

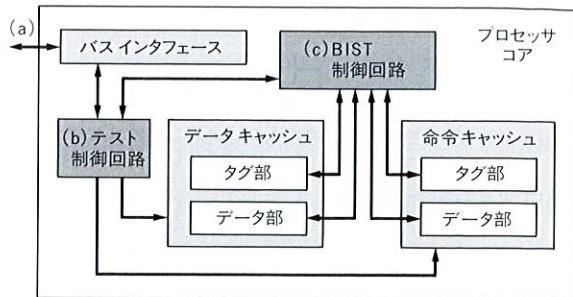


図 6. RISC キャッシュの BIST 回路 キャッシュの BIST 回路により外部からのテストやデバッグが容易になる。

BIST logic for RISC processor cache

テストが可能となり、また故障解析機能をもたせることができるようにになった。

5 あとがき

複数の異なる種類の機能ブロックからなるシステム LSI のテスト設計では、個々のブロックのテスト設計手法を効果的に組み合わせることがかぎとなる。一方、コアの設計においても容易に組合せのできるテストインターフェースをもつことが必要となる。テスト設計用のツールもこれに合わせ、各手法を単独ではなく組み合わせて効率的に実現する方向で発展していく。当社では、トップダウン設計手法に適した RTL でのテスト設計を含め、設計環境の整備を引き続き行っていく。

文 献

- (1) 安藏顕一、他：ASIC テスト設計サポートシステム、東芝レビュー、50, 6, pp.457-459 (1995)

安藏 顕一 Ken'ichi Anzou

半導体設計・評価技術センター 半導体評価技術部主務。
LSI のテスト容易化設計システムの開発に従事。
Semiconductor DA & Test Engineering Center



上戸鎖 裕史 Hirohumi Kamitokusari

半導体設計・評価技術センター 半導体評価技術部主務。
LSI のテスト容易化設計システムの開発に従事。
Semiconductor DA & Test Engineering Center

