

## 機能テスト発生システムの開発

河野 和義, 武井 勉, 関根 優年  
(株)東芝 ULSI 研究所

LSIの回路規模は年を経る毎に増大している。これに伴って、回路の設計も論理レベルの設計から機能レベルの設計へと移行しつつある。そこで問題になるのが、機能レベルでの設計検証ツールである。当社では1983年10月以来FALと呼ばれる機能レベルのシミュレーションシステムを開発してきた。今回、このシステムの機能強化をはかる意味で、機能レベル記述から回路の機能を検証するためのテストパターンを作成するツールである”機能テスト発生システム”の開発を行った。本システムはたんにテストパターンを発生するだけではなく、回路の情報を入力する段階での誤りを検出するという目的で開発を行った。まだ開発はプロトタイプ段階であるが、今後機能強化をはかり、評価を行っていく予定である。

Functional Test Pattern Generation Assistant ( FUNTASSI )

Kazuyoshi Kohno, Tsutomu Takei, Masatoshi Sekine  
ULSI Research Center, TOSHIBA Corporation

1, Komukai Toshiba-cho, Saiwaiku, Kawasaki 211, Japan

This paper describes a functional test pattern generation assistant (FUNTASSI). The authors' register transfer level language has a clear syntax to divide a description into a control flow, a clock timing flow, a data flow and a state transition flow. To reduce the execution time, FUNTASSI makes best use of this division. The simple path activation algorithm is adopted for the prototype. The user interface supports to define the data paths for the specified circuit function, and also to search signal flows for a simulation result analysis.

## 1. はじめに

近年の微細加工技術の進歩は目ざましく、それに伴い回路の集積度は年を追う毎に増加する傾向にある。このため大規模で複雑なシステムがVLSI化されるようになってきており、チップ作成前の完全な検証はますます困難になりつつある。

これに伴い、回路の設計も論理レベルの設計から、機能レベルの設計へと移行しつつある。我々は1983年10月以来、mixed level simulator FALを開発してきた。このシミュレータは、回路の機能レベルからゲートレベルに至る記述レベルに対してシミュレーションを行うことが可能なシステムである。このシステムにより、従来行われてきたブレッドボードによる検証に替わる検証手段が確立され、かなりの設計期間の短縮が実現された。

またFALシステムをとりまくサブシステムとして、機能図入力システムFSET、論理合成システムがある。

FSETは、機能レベルの回路を図形で入力するためのシステムである。設計者はFSETを用いて回路図を入力することにより、その回路に対応する機能レベルの言語記述を得ることができる。設計言語による回路の設計はプログラム作成とほぼ同じであり、設計者にとってなじみの薄いものである。従って、それゆえに誤りが混在しやすい。FSETによる入力は基本的には機能ブロックの配置とブロック間の結線を行うだけでよいので操作が簡単である。

また論理合成システムは、機能レベルの記述をゲートレベルの記述に変換するシステムである。

上述した2つのサブシステムは回路記述の作成をサポートするシステムと位置づけることができる。シミュレーションを実行するためには、回路情報とテストボタンが必要である。回路情報の作成は上述した2つのサブシステムによりかなりの効率を上げることができる。しかしテストボタンの作成は現状では人手で行っており、これが設計検証のネックになってきている。そこでこれを解消する一つの手段として機能レベルの記述からテストボタンを作成するシステムの開発を行った。

## 2. 機能テスト発生システム概要

### 2. 1. 機能テスト発生システムの位置づけ

第1図は現在のFALシステムのシステム構成図である。設計者はschematic editor (FDE)により回路図、またはtext editorによりRTL記述を入力する。FDEにより作成された回路図は機能図言語変換(FDLC)によりRTL記述に変換される。作成されたRTL記述は論理合成システムにより論理レベルの記述に落とすことも可能である。これらの記述はコンパイラ、リンカに通すことによりシミュレータ(FAL)用のロードファイルに変換される。

シミュレーションを実行するためには回路の情報(ロードファイル)とテストボタンが必要となる。テストボタンを作成するツールとしては、マクロやニーモ

ニックを用いてテストパターンを記述するものや、波形によりテストパターンを入力するものがある。しかしこれらのツールを用いたとしても結局は人間がテストパターンを作成しなければならない。テストパターンを作成する作業は煩雑であり、特に回路の規模が大きくなると人手で行うのは困難になってくる。また設計の初期段階では回路の変更がたびたび起こるため、その都度テストパターンを作らなければならない。テストパターン作成の作業はかなり機械的な作業もあり、その部分は計算機で行わせることが可能である。また、テストパターンは回路記述をもとに発生させるため、記述が正しくなければテストパターンは発生できない。このことは、裏を返せば、テストパターンが発生できないときには、回路の入力時に誤りがあったということである。従って、テストパターンが発生できないときには、その情報を出力することにより回路の設計時に混入したバグを発見することができる。

現在開発を行っている機能テスト発生システムは、回路の機能レベルの設計段階に於て、設計者が回路の検証を行う際に必要となるテストパターンを作成し、また、テストパターンが作成できないときには、その原因となる情報を出力し設計の誤りの修正をサポートするツールである。

## 2. 2. 機能テスト発生システムの役割

回路は、”入力信号を受け取りその信号をもとになんらかの処理を行い、出力信号を出すもの”ととらえることができる。入力信号の種類には、データ信号、制御信号、クロック信号がある（第2図参照）。制御信号はデータの流れる方向を制御し、クロック信号はデータの流れるタイミングを制御し、その結果、データ信号が様々な機能モジュールを通過することにより出力データが得られる。従って、回路は、信号の流れに着目すると大まかに以下の3つで構成されていると考えられる。

- ・データ信号が流れる経路（データ系バス）
- ・制御信号が流れる経路（制御系バス）
- ・クロック信号が流れる経路（クロック系バス）

回路の機能を検証するためには、

- (1) 検証したいデータバスを求める。
- (2) (1)で求めたデータバスに対して入力側から出力側へデータを伝搬させるためのテストパターンを求める。
- (3) (2)で得られたテストパターンを用いてシミュレーションを行ない得られたシミュレーション結果をもとに検証を行う。

という手順を踏めばよい。上記(1)、(2)の役割をになうのが機能テスト発生システムである。(1)の処理を行う部分をインタフェイス部、(2)の処理を行う部分をバス活性化部と呼ぶ。

検証を行うための処理フローを第3図に示す。まずインタフェイス部によりデ

ータバスを指定する。しかし、この時テストしたいデータバスを指定できない場合も考えられる。これは、素子間の接続関係が設計者の意図した通りに作られていないということの意味する。従って、まずデータバスを指定する段階で、設計の誤りを発見することができる。

設計者の望むデータバスが指定できた場合には、そのバスを活性化するためのテストボタンを求める処理に移る。"バスを活性化する"とは、"バスの入力側から出力側へ信号を伝搬させる"ことである。バス活性化部ではデータバス情報と接続情報をもとにデータバスの活性化を行う。このとき、与えられたデータバスを活性化するためのテストボタンが求められない場合が考えられる。これは、各端子に可能な限りの値設定を行っても、同時刻に異なる値を設定しなければならない端子が存在することを意味する。この様な状況は、

- ・ 回路の制御部の設計がおかしい。
- ・ データを転送するタイミングがおかしい。

という場合に起こると考えられる。従って、バス活性化の段階でも設計の誤りを発見することができる。

与えられたデータバスに対してテストボタンが発生できたときは、そのテストボタンを用いてシミュレーションを実行する。機能テスト発生システムにより得られるテストボタンは、"特定のデータバスを活性化するための入力ベクトル列"であり、"機能レベルの故障を仮定し、その故障を検出するためのテストボタン"ではない。従って、設計が正しいか否かの判断は、シミュレーションの出力結果を設計者が観察することにより行う。出力結果が設計者の期待したものと一致しなかった場合には、回路の設計に誤りがあるということになる。期待したものと一致した場合には、シミュレーション結果は期待値として用いることができる。

### 3. 各プログラムの役割

第4図は機能テスト発生システムのシステム構成図である。以下でシステムを構成している主だったプログラムの説明を行う。

#### 3. 1. ネット作成部の役割

ネット作成部は機能レベルのHDLを入力としてネット情報を作成する。ネット情報には以下に示す情報が盛り込まれている。

- ・ 回路を構成している素子の接続関係（接続情報）
- ・ 素子間を信号が伝搬するための条件（信号伝搬条件）

接続関係には次の3種類がある。

- ・ データ系の接続
- ・ 制御系の接続
- ・ クロック系の接続

上記した3つの接続関係は機能レベルのHDLを用いて表現することが可能である。ネット作成部ではHDL記述に含まれているこれらの接続関係を抽出する働きをする。また、信号の伝搬が条件により制御される場合（即ち、制御系の接続がある場合）、またはクロックにより制御される場合（即ち、クロック系の接続がある場合）、ネット抽出部では、その条件、クロック信号の抽出を行う。

### 3. 2. インタフェイス部の役割

機能の検証をするために、まず第一に行うことは回路記述から検証したいデータバスを求めることである。データバスを求める方法としては、

- (1) 計算機が回路の接続関係を頼りに、自動的にデータバスを求める方法
- (2) 設計者が計算機と対話形式でデータバスを指定してしていく方法

等が考えられる。

インタフェイス部は(2)の方法でデータバスを求める際、設計者と計算機の間には、データバスの指定を手助けする役割をになう。

第1図に示す通り、現在のFALシステムでは回路の入力方法には、

- ・ schematic editor による回路図の入力
- ・ text editor によるHDL記述の入力

の2通りがある。これに対応して、バスの指定方法にも、

- ・ 回路図によるバスの指定
- ・ HDL記述によるバスの指定

が用意されている。

### 3. 3. バス活性化部の役割

バス活性化部は、インタフェイス部で指定されたデータバスの入力側から出力側へデータを伝搬させるためのテストパターンを求める役割をになう。

データバスはHDLのレベルでみれば、文の列としてとらえることができる。活性化の処理では、データバスの入力側から順に、対応する文を活性化するためのテストパターン（即ち、文の右辺の端子から左辺の端子へデータを伝搬させるた

めのテストパターン)を求めて行く。信号の伝搬が条件により制御されている場合、その条件を成立させる必要がある。制御条件を成立させるためのテストパターンを求める処理を、制御系バスの活性化という。また、信号の伝搬がクロックにより制御されている場合、データ信号を伝搬させる時にクロック信号を立ち上げる必要がある。クロック信号を立ち上げるためのテストパターンを求める処理を、クロック系バスの活性化という。

第5図に活性化の処理フローを示す。上でも記した通り活性化の処理はデータバスに対応する文列の入力側から順に活性化を行っていく。活性化途中で値設定に矛盾が生じたときには、値設定の方法、制御バスの変更、クロックバスの変更、クロック信号の立ち上げ時刻の変更等を行い矛盾の解消を行う。それでも矛盾の解消が出来ない場合には、現在活性化を行っている文より以前に活性化を行った文で変更可能なものがあれば、そこから改めて活性化を行う。変更可能な文がないときには活性化不能であるとして処理を終了する。

### 3. 3. 1. 制御系バスの活性化

制御系バスの活性化は、以下の2つの処理からなる。

#### (1) 制御系バスの探索

回路内の端子の接続関係はネット情報にその情報が含まれている。制御系バスの探索は、条件式に現れる各端子毎に、その端子を始点として入力方向にたどって行けばよい。バス探索時に複数の選択する余地がある場合には、適当な選択基準を用いて選択を行う。

#### (2) 信号値の入力方向への伝搬

制御系バスが求められたら、そのバスの終点の端子(一番出力側の端子)に特定の値を設定するために、そのバスの始点の端子(入力端子)の値を決定しなければならない。入力端子の値の決定は、出力側から順に値を求めることにより行う。制御系バスは一般的にtree状の形をしており、出力側の1つの値に対して入力側の値は一般に複数個存在する。例えば、

$$T3 = T1 ! T2; \quad (! \text{ は OR 演算子を表す})$$

のような記述が制御系バス中に出てきた場合、出力側の端子(T3)の値と演算(OR)から入力側の端子(T1とT2)を求める必要がある。T3の値が1の時、T1, T2の値として、

$$(T1, T2) = (1, 0), (0, 1), (1, 1)$$

の3通りが考えられる。このなかから値設定をしても矛盾がないものを選択し処理を進めていく。

### 3. 3. 2. クロック系パスの活性化

記憶素子に値を転送するときには、その記憶素子に付随するクロック信号の値を0から1に変化させる（たち上げる）必要がある。クロック信号の流れる経路をクロック系パスという。クロック系パスの活性化も、制御系パスの活性化と同様に、以下に示す2つの処理からなる。

- (1) クロック系パスの探索
- (2) 信号値の入力方向への伝搬

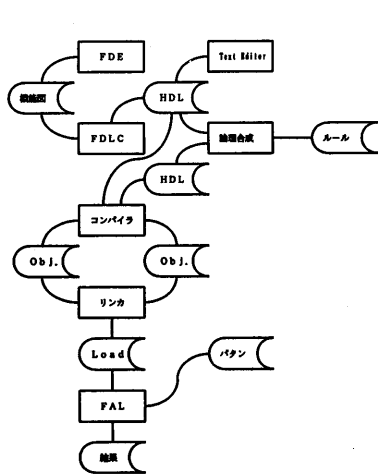
これらの処理は制御系の場合とほぼ同様である。

## 5. まとめと今後の課題

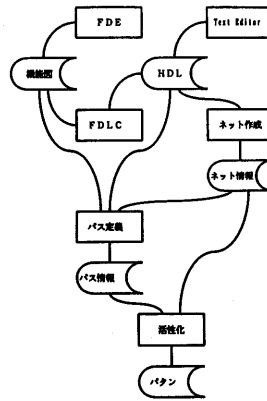
機能レベルの回路記述をもとにその機能を検証するためのテストバタンを発生するシステムの開発を行った。本システムは東芝製ラップトップパソコンJ3100上に実現されている。現在のところ、本システムはプロトタイプの段階であり、扱える回路記述に少なからず制限がある。従って、実際の回路を用いた評価等は行っていない。今後この制限を取り払い、実際の回路を用いた評価を行っていきたい。また、処理の効率向上をはかるうえで今後強化・開発を行っていかうと考えている主だった項目を記す。

- (1) データパスの指定方法の強化
- (2) 活性化方法の強化
- (3) 活性化不能時の情報出力機能の強化

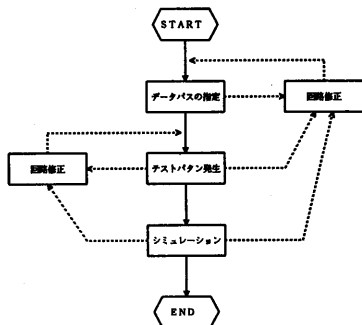
特に(3)は今後重要となってくると思われる。即ち、設計の初期段階では、設計の誤りが多数存在することが考えられる。従って、設計の初期段階に於て、テストバタンが得られることはまれであると考えられる。テストバタンが得られなかった場合には回路の修正を行わなければいけない。このときに、その参考となる情報が出せれば、設計の効率のかなりの向上が期待できると考えている。



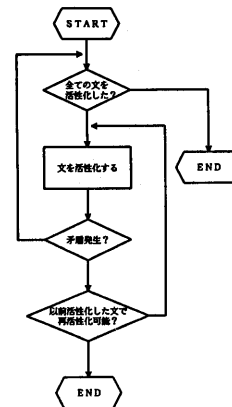
第1図 FALシステム図



第3図 テスト発生システム図



第4図 検証の処理フロー



第5図 活性化の処理フロー