

テストプログラム自動生成ツール「MTST」によるシステムレベル検証

6B-4

関 誠司, 白井 健治, 平岡 精一, 三部 健, 佐久間 孝夫, 君島 達也

三菱電機(株)

1 はじめに

近年、計算機システムの高速化/高機能化にともない、計算機システムの動作は複雑化している。このため、システムレベル検証の検証項目は増大し、検証内容は複雑になってきている。従来の人手による検証では次のような問題点がある。

- テストプログラムの生成にかなりの時間を費やさなければならない。
- 検証洩れの発生する危険性がある。

上記のような問題点を解決するためにテストプログラム自動生成ツール「MTST」を用いた。このツールを用いることにより、テストプログラムの生産性は従来の方法に比べて5~10倍に向上し、テストカバレッジも向上した。本稿では、我々の行なったCPU開発において、このテストプログラム自動生成ツールである「MTST」を用いて検証効率の向上を達成したシステムレベル検証について報告する。

2 検証方法

2.1 従来の検証方法の問題点と解決方法

従来のシステムレベル検証では検証項目のリストアップ、テストプログラムの生成及びシミュレーション結果の解析を人手によって行なってきた。システムの構成がかなり複雑になってくると、システムレベル検証の検証項目は増大してきて全ての項目を人手でリストアップすることは困難になり、検証洩れの発生する危険性が出てきた。また、それぞれの検証項目の内容は、特定のタイミングでリクエストを起こさなければいけなかったり、複数のオペレーションを連続させなければいけないものが多くなっている。そのため、テストプログラム作成時に、リクエストのタイミング調整及びメモリやキャッシュ等の初期値やテスト終了時の期待値等の生成に多くの時間を費やさなければならなくなった。

そこで、我々は上記のような問題点を解決するために、以下の特徴を持つ「MTST」を使用して検証を行なう

System-level Verification Using Automatic Test-program Generator 'MTST'
Seiji SEKI, Kenji SHIRAI, Seichi HIRAOKA, Ken SAMBU, Takao SAKUMA, Tatsuya KIMISHIMA
Mitsubishi Electric Corp.

ことにした。

- メモリ、キャッシュ及びLSI内部のレジスタ等の初期値の設定を自動的に行なう。
- テスト終了時の期待値を自動的に生成する。
- 各バスドライバに指定するリクエストのタイミングをランダムまたは固定的にずらすことができる。
- オペレーションの種類やリクエストのタイミング等といった簡単な条件指定だけで複雑な試験を生成できる。
- オペレーションの種類やリクエストのアサートタイミング等の指定条件の一部、または、全部をランダムまたは固定的に設定し、試験を生成できる。
- システムバス上のオペレーションのログを出力する。

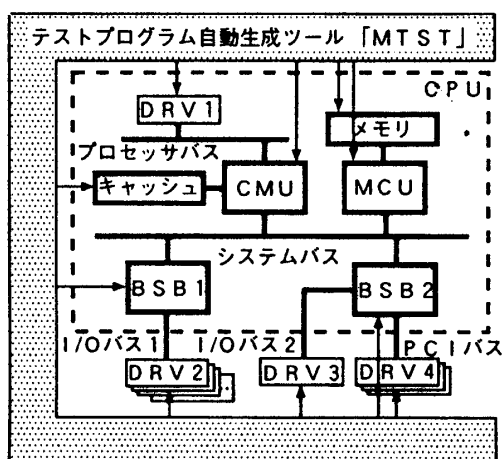
2.2 検証対象

今回、検証の対象となったCPUを図1に示す。CPUの内部バスとしてプロセッサバス及びシステムバスがあり、I/OバスとしてPCIバスとその他に2本のバスがある。システムバスには複数のLSIが接続されている。それらは、MCU（メモリ制御ユニット）、CMU（キャッシュ管理ユニット）及び2つのバスブリッジBSB1、2である。BSB2はPCIバスとI/Oバス2のバスブリッジになっている。これらのLSIのほとんどが能動的にシステムバス上にリクエストを出力することができる。また、このシステムバスではリクエストを複数スタックできるようになっている。PCIバスには3つのエージェントが接続可能で、その3つのエージェントは能動的にリクエストを出力することができる。

2.3 検証項目と検証方法

今回は多くの検証項目の中から、検証項目数が膨大で検証項目の内容が複雑であるという事と新たに採用したバスで検証経験がないという事のそれぞれの理由で以下の2つの大項目について「MTST」を用いて検証を行なった。

- システムバス上でのオペレーションの競合
- I/Oバス上でのリクエストの競合



注) MCU:メモリ制御ユニット
 CMU:キャッシュ管理ユニット
 BSB:バスブリッジ
 DRV:バスドライバ

図1 検証モデル

2.3.1 システムバス上でのオペレーションの競合

能動的に動作する複数のLSIからのオペレーションはシステムバス上で競合する。システムバス上ではオペレーションを複数スタックすることができるために、3つのオペレーションがシステムバス上で競合する検証項目だけでもその数は約1万項目になった。それら全てを手で作成するのは時間的に無理なので、ランダム試験を導入することにした。全ての条件をランダムにしたのではテストプログラムが冗長になるので、条件を指定することにより代表的なテストプログラムを作成し、その後で指定した条件をランダムにすることにより効率的にテストプログラムを生成した。

システム動作としてオペレーションが競合する状況には次の3種類がある。

- メモリアクセスオペレーション
- メモリアクセス以外のオペレーション（LSI内部のレジスタなどへのアクセス）
- 上記2種類のオペレーション間の組合せ

まず、それぞれの分類毎に代表的なテストプログラムを生成した。メモリアクセスオペレーションのテストプログラムでは、メモリアクセスのオペレーションの種類、リクエストのタイミング及びキャッシュの初期状態を指定した。LSI内部のレジスタなどへのアクセスオペレーションのテストプログラムでは、LSIの種類及びリクエストのタイミングなどを指定した。その後、2種類のオペレーションを含んだテストプログラムを作成した。そして、作成した代表的なテストプログラムのリクエストのタイミングをランダムに変更するように設定し、リクエストの順序や間隔を変更して、競合試験を行なった。

2.3.2 I/Oバス上でのリクエストの競合

今回、I/Oバスの1つにPCIバスを新規に採用した。そのために、従来、他の2つのI/Oバスのように発生しなかった競合条件が起こるようになった。システムバスからPCIバス上のエージェントに対してアクセスするリクエストとPCIバス上のエージェントからメモリに対するアクセスのリクエストが競合する場合である。まず、初めにリクエストのタイミングやオペレーションの種類などを指定して代表的な状況のテストプログラムを作成した。そして、作成した代表的なテストプログラムのリクエストのタイミングをランダムに変更するように設定し、リクエストの順序や間隔を変更して、競合試験を行なった。

3 効果

「MTST」を用いて検証を行なった結果、大きく以下の3点の効果が得られた。

- テストプログラム生産性の向上
 従来の人手によるテストプログラム生成に比べ、以下の点を自動化したのでテストプログラムの生産性は5~10倍程度向上した。
 - メモリ及びキャッシュの初期値の設定
 - LSI内部のレジスタ等の初期値の設定
 - 期待値の生成
 - 指定したリクエストのタイミングの設定
- テストカバレッジの向上
 人手では生成できないまたは生成しにくいテストプログラムを、以下の機能により網羅的に生成するのでテストカバレッジが向上した。
 - オペレーションの種類やリクエストのタイミング等の指定条件の一部、または、全部を自動的に設定し、試験を生成できる。
- テストカバレッジの低下の防止
 以下の機能を自動的に実行しない、マスタのログファイルと実行したログファイルを比較することにより、バス上に期待した動的な状態が発生しているかをチェックできるので、テストカバレッジの低下を防止できた。
 - システムバス上で発生したオペレーションのログの出力

4 まとめ

コンピュータのCPU開発において、複数の条件が同時に発生する検証項目にはテストプログラム自動生成ツールである「MTST」は非常に有効な検証手法であることが分かった。