

非同期式プロセッサ TITAC-2 の検証とテスト

1 G-7

深作 泉

高村明裕

小沢基一

上野洋一郎

南谷 崇†

東京工業大学 情報理工学研究所

†東京大学 先端科学技術研究センター

1 はじめに

クロック分配に起因する同期式システム設計の限界を突破することを目指し、事象駆動原理に基づいた非同期式プロセッサが実用レベルの機能と規模で実現できることを実証するために、32ビット非同期式マイクロプロセッサ TITAC-2 を設計した [1]。一般に、論理システムが仕様どおりに正しく設計されていることを保証するためには、実現される論理(機能)そのものの検証と、動作タイミングの検証が必要である。また、その設計に従って製造された LSI チップが仕様どおりに正しく動作することを保証するためにテストが必要である。しかしながら、32ビットマイクロプロセッサのような大規模な非同期式システムに有効な設計検証およびテスト方式に関する研究はまだ十分ではなく、現段階では有効な方法が確立されているとは言えない状況にある。本稿では、TITAC-2 の論理設計にあたって採用した設計検証方法とテスト方法について報告する。

2 論理設計の検証

Delay-Insensitive (DI) モデルを前提として正しく設計された回路であれば、論理自体の検証のみを行えばよく、動作タイミングの検証は不要である。Quasi-Delay-Insensitive (QDI) モデルについても、同様に動作タイミングの検証はほとんど不要である。しかし、TITAC-2 では、性能の向上を図るため、局所的な組合せ論理回路では、回路要素の遅延の相対的変動幅に上限があると仮定する比例遅延変動モデル [1] を採用している。この遅延モデルのもとでは動作タイミングの検証を十分に行う必要がある。

TITAC-2 は、仕様から Verilog-HDL で機能レベル記述をし、それを基にゲートレベル記述を行うという手順で設計が行われた。記述はいくつかの機能単位でモジュール化されており、分割することが可能である。そこで、TITAC-2 では、設計単位となったモジュール毎に検証し、その後全体の検証を行うという方法をとった。

2.1 順序回路と組合せ論理回路の分割検証

TITAC-2 は順序回路を含むパイプライン制御回路と、組合せ論理回路である各機能単位に分割して設計されている。

パイプライン制御回路は、QDI モデルに基いているため、トレース理論を用いた検証器 [2] により形式的検証を行った。

組合せ論理回路の論理検証に関しては、同期式、非同期式の差はないので同期式回路の検証に用いられる市販の論理シミュレータを用いた。機能レベル記述及びゲートレベル記述のシミュレーションにおいて、特定の機能の検証を意図して設計者が選択した入力、及び乱数入力を与え、出力結果の一致を確認した。

動作タイミングの検証には、レイアウト後のゲート遅延、

配線遅延データを使用した。それらのデータには標準値、最大値、最小値がある。そこで、この3種類の遅延を設定した論理シミュレーションに対し、組合せ論理回路による遅延が最小、及び最大になると考えられる入力と乱数入力を与え、出力が確定するまでの遅延変動が設計段階で予測した範囲内であることを確認した。

この論理シミュレータを用いる検証方法には問題がある。実際の回路ではゲート及び配線毎に標準値を中心とした別々の遅延値を取りうる。そのため、上記の方法による検証では、潜在しているタイミングフォールトを見逃してしまう可能性がある。そこで今回は、設計中にタイミングフォールトの可能性のある部分を特定しておき、遅延が決定した後に人間の手でチェックを行ない、タイミングフォールトが発生しないことを確認した。

2.2 全体結合後の検証

機能単位に分割して行う検証での入力タイミングは、必ずしも全体を結合したシステムにおける実際のタイミングを表すものではない。そこで全体を結合してシミュレーションを行った。

全体結合後の検証には、2種の方法を用いた。

ソフトウェア面から見て等価な動作をするエミュレータを、TITAC-2 の仕様から C 言語で作製した。このエミュレータと、機能レベル記述、及びレイアウト後の遅延情報を用いたゲートレベル記述による論理シミュレータで、TITAC-2 各部の機能の検証を意図したプログラム、及び命令列を乱数で生成したランダムパターンプログラムを実行した。そして、実行後の内部レジスタの状態及びメモリ内容が互いに一致することを確認した。

しかし、この方法では、レジスタやメモリに書き込まれている誤った演算結果が、以後の実行に用いられず、正常な結果に上書きされた場合に、設計ミスを見逃す可能性がある。そこで、次に述べる方法で追加検証と、設計時に入力タイミングに制約を課している部分でのタイミング余裕を確認した。

シミュレーション時に、TITAC-2 の一つのパイプラインのステージを検証の対象とし、そのステージへの入力値と入力のタイミング、そのステージからの出力値、ステージ間ラッチを介して接続されている次のステージへの入力値を記録した。エミュレータからそのステージに相当する部分を切り出し、記録されている入力値を与え、それによる出力と、記録されている出力の一致の確認により、対象ステージの論理の検証を行った。また、対象ステージからの出力と、次のステージへの入力の一一致の確認から、対象ステージからの出力の遅延変動幅が、ステージ間ラッチへの入力遅延変動の許容範囲内であったことが確認できる。記録された対象ステージの入力タイミングから、タイミング余裕が十分にあることの確認を行った。

3 製造後のテスト

同期式プロセッサのテストに良く用いられる方式に、スキャンバス方式がある。TITAC-2 にそれを適用しようとした場合、回路量の大幅な増加と性能の低下が予想されたため、採用できなかった。そのため、製造後のテストはプログラムの実行によって行う必要がある。プログラムによるテストでは、故障検出率を上げる方法が問題となる。

Verification and testing of the asynchronous processor TITAC-2
Izumi Fukasaku, Akihiro Takamura, Motokazu Ozawa, Yoichiro Ueno
Tokyo Institute of Technology, Graduate School of Information Science and Engineering
Takashi Nanya
Research Center for Advanced Science and Technology, The University of Tokyo

3.1 テスト用付加回路

TITAC-2は大域的な配線にはDIモデル、局所的な回路に対しては比例遅延変動モデルを用いて設計されている。DIモデルで設計された部分での縮退故障は、即座にプロセッサを停止させるため、容易に故障検出が可能である。しかし、QDIモデルや、比例遅延変動モデルによって設計された部分での故障では、回路が停止しないまま誤動作になる可能性がある[3]。

そのような場合の故障検出率を高めるために、非符号語検出回路を設けた。TITAC-2では、1ビットのデータを肯定線、否定線の2本で表現する2線2相式[1]を採用している。肯定線、否定線が共に1の符号は非符号語であり、正常時には発生しないことから、故障の検出に利用できる。そこで、非符号語の発生を検出する回路をパイプラインの各ステージ間ラッチ[1]に付加した。このステージ間ラッチを図1に示す。この検出回路の出力probeを、テスト用出力端子として外部に引き出し、製造後のLSIテストに利用することにした。

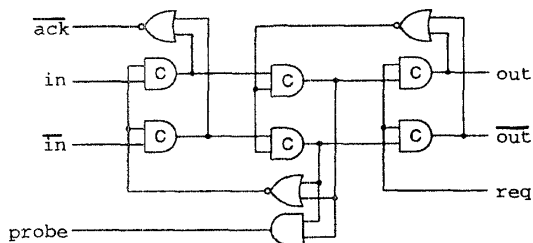


図1: 非符号語検出付きステージ間ラッチ

プログラムから直接参照できないレジスタの出力や制御回路の制御信号等も、テスト用出力としてLSI外部に引き出されている。

3.2 非同期式LSIと、同期式LSIテストの同期

通常の同期式LSIのテストでは、LSIとLSIテストがクロック信号によって同期し、テストパターンの送出力、出力期待値の照合が行われる。しかし、TITAC-2にはクロック信号が存在しないため、それを用いた同期方法は使用できない。そこで、外部メモリアクセスが発生した時点でLSIテストと同期を取る方法を用いた。

TITAC-2の命令読み出し時におけるメモリアクセスの代表的なタイミングチャートを図2に示す。

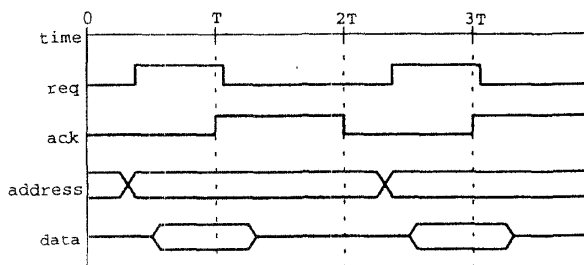


図2: 入出力タイミングチャート

図2のTはテストパターン周期を示している。reqは命令読み出し時のメモリアクセスの要求、ackはその要求に対する応答、addressはアドレスの出力、dataはデータの入力である。通常、req、address、dataは外部メモリに接

続され、ackは遅延素子を介してreqに接続されている。

メモリアクセスのサイクルは、まずアドレスの確定から始まる。その後、reqが立ち上がり、ackの立ち上がりを待ってデータが取り込まれ、reqが立ち下がる。そして、ackの立ち下がりをもってサイクルが終了する。このように、TITAC-2のメモリアクセスは要求-応答方式となっている。

テスト時は、data及びackはLSIテストから送出される。応答信号ackの変化する時間は、テストパターン周期Tに同期している。TITAC-2はackの変化を待って動作するため、この時点でLSIテストと同期する。この際のテストパターン周期Tは、命令の読み出しによって、プロセッサが動作し、次の命令読み出しのメモリアクセスの要求が発生する時間よりも十分長い必要がある。

この例の場合には、期待値との照合はアドレスの出力値によって行う。アドレス出力は、メモリアクセスサイクルの間は確定しているので、テストパターン周期の倍数であるT、2T及び3Tの直前で期待値との照合が可能である。

3.3 テストパターンの生成

テストパターンと出力の期待値は、論理シミュレータ上でテストプログラムを実行した結果から生成した。これと同時に、テスト用出力端子を用いて制御回路の制御信号の状態確認、及び非符号語発生を検出を行うパターンを付加した。

テストプログラムは、同期式プロセッサにおけるデータバス部の単一縮退故障の検出手法[4]を基にして作成した。非同期式プロセッサでは、故障があった場合停止する可能性が高いと考えられるが、停止せずかつ非符号語も発生しない場合もあり得る。そこで、プログラムは演算結果や外部へのストアとロードの結果が事前に求められている期待値と異なった場合等に、異なったアドレス、データを外部に出力するように作成した。

4 まとめ

32ビット非同期式マイクロプロセッサTITAC-2の設計・試作において採用した設計検証方法とテスト方法およびその問題点を明らかにした。これらの問題点の解決は今後の重要な課題である。

本研究の一部は、新エネルギー・産業技術総合開発機構(NEDO)提案公募型・最先端分野研究開発事業受託研究C-026、並びに科学研究費補助金(試験研究B)07558036によって行われたものである。

参考文献

- [1] 高村, 桑子, 南谷. 非同期式プロセッサTITAC-2の論理設計における高速化手法. 電子情報通信学会論文誌, Vol.J80-D-I, No.3 (March 1997) 掲載予定
- [2] Tomohiro Yoneda and Takashi Yoshikawa. Using partial orders for trace theoretic verification of asynchronous circuits. In *Proceedings International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp.152-163, March 1996
- [3] Arthit Thongtak, 南谷 崇. Quasi-Delay-Insensitive 論理回路の縮退故障テスト. 電子情報通信学会論文誌, Vol.J80-D-I No.2, pp.1-9, February 1997
- [4] S.M.Thatte and J.A.Abraham. Test generation for microprocessors. *IEEE Trans.Comput.*, vol.C-29, pp.429-441, June 1980