

オンチップPLLを用いたLSSD高速スキャンテストと ソースシンクロナスDDRインターフェイスのテストへの応用

横田 俊彦

日本アイビーエム 〒604-8175 京都市中京区円福町338 IBMビル

E-mail: tyoko@jp.ibm.com

あらまし 出荷するモジュール品質の改善のために、オンチップPLLを用いた高速スキャンテストをLSSD設計で行った。また、オンチップPLLを用いたテストの応用としてソースシンクロナスDDR (Double Data Rate) インターフェイスをJTAGを用いてテストする方法を提案する。

キーワード ハイスピードスキャンテスト、オンチップPLL、ACテスト、IEEE1149、Double data rate、Source synchronous

LSSD at speed scan test and Source synchronous DDR interface test by 1149 using on chip PLL

Toshihiko Yokota

TCS, IBM Japan, Ltd Enpukuji-cho 338, Nakagyo-ku, Kyoto-shi, Kyoto, 604-8175 Japan

E-mail: tyoko@jp.ibm.com

Abstract High speed scan-based test using on chip PLL has been developed in IBMASIC based on level sensitive scan design (LSSD) in order to improve shipped module quality. Source-synchronous double data rate interface was found to be testable at full function speed via IEEE1149 interface in a similar way and its implementation has been investigated.

Keyword High speed scan test, On chip PLL, AC test, IEEE1149, Double data rate, Source synchronous

1. はじめに

0.13 μ テクノロジーを境に、IBMASICでは出荷テストをパスしたにもかかわらず、製品の実機スピードで動作したときに問題の起こるLSIが無視できない数になっていた。そのころオンチップPLLを使った高速スキャンテストを試行する機会があったため、Last scan launch off(あるいはLaunch-off-shift、以下、LSLO)とCapture launch off(以下CLO)方式の有効性、問題点、LSSDでの適用の可能性など、本試行を通じて調べた。

またこの検討の際に、オンチップPLLを使ったテストの応用として、JTAGを介して、ソースシンクロナスDDRインターフェイスのテストを高速で実施する方法も検討した。ソースシンクロナスDDRはDRAMを始めモジュール間の高速インターフェイスを実現するもっとも有効な方法のひとつであり、多くのLSIで使われているのであるが、タイミング仕様をミートするのは困難であり、また、このインターフェイスにタイミングなどの問題があった場合、システムレベルのデバッグは困難なものとなりがちである。この問題解決のためには、機能回路の一部としてデバッグ回路を入

ればよいのであるが、このように広く使われるインターフェイスであることを考えると、ASICベンダーに標準的なデバッグ回路をテスト回路のように自動的に挿入するようにしたほうがよいと考える。JTAGテスト回路を利用して組み込むことにより、機能にかかわるバスへのタイミング的なインパクトを小さくした方法を提案する。

2. LSSD ACディレイテストとその限界

LSSD設計では、テスト専用LaunchクロックとCaptureクロックを用いて、Fig1のように、ラッチ-ラッチパスのディレイテストを行うことができる[1]。しかしこのテストのクロックはテスターから供給されるために、モジュールピンですでに少なくとも数百ピコ秒の到着時間誤差を伴うことになり、また、モジュール内でもデータのLaunch、Capture動作はシステムクロックではなくテストクロックツリーを用いて行われるために、本来テストをしたいシステム動作のセットアップ、ホールドタイムのテストを行っていない。IBMASICでは、0.13 μ まで、このテストクロックのスケールをできる限り小さくすることによって、“なるべく”正確なラッチ-ラッチパスのディレイテストを行

ってきており、[1]に述べられるように、ある程度の成功を納めてきた。しかしながら、このテストでは本来のSetup, Holdのテストができないために0.09 μ 以降のテスト方法としては不十分であると考える。

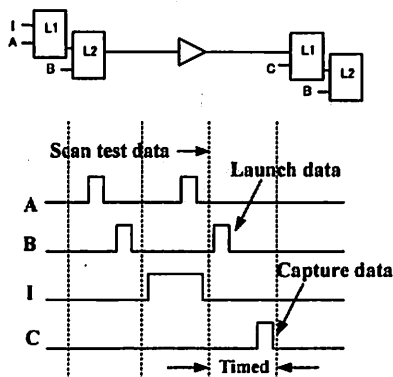


Figure 1: Illustration of AC delay test in LSSD

3. LSSD による CLO と LSLO の実現方法

高速スキャンテストを実現する CLO と LSLO を LSSD 設計で実現する方法を述べる。

3.1. Splitter

後の説明のためにはじめにIBMASICのクロック分配方式について説明をする。

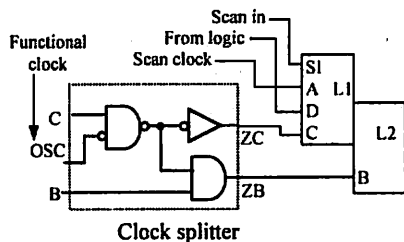


Figure 2: Clock splitter

IBMASICではクロックツリーのリーフレベルにFig2に示されるClock Splitter (以下SPC) と呼ばれるセルを置き、1相クロック信号を2相クロックに変換しLSSDラッチに分配する。ひとつのSPCが10-20のLSSDラッチのクロックをドライブする。ファンクション時はSPCのB, Cの入力信号はともに'1'の固定信号が与えられ、OSCから入力される1相クロック信号の反転信号と非反転信号が、ZC, ZBから出て行く。テスト時は、C='0'となり、SPCのB→ZBを介してスキャンクロックが供給され、さらに、B→ZB, C→ZCを介してLaunch、Captureクロックが供給される。

3.2. LSSD 設計で実現する CLO

[2]にCLOをLSSDで実現方法とその際のLaunch-Captureのシーケンスの波形が示されているが、

このシーケンスのSPCの各ピンでの波形をFig3に示した。それは以下になる。

(1) SPC/C: 0→1の遷移でDピンの値つまりシステムデータ入力をL1 (マスターラッチ) に取り込む。この動作はBroadside Loadと呼ばれ、続いてB='1'とする。

(2) PLLから1つ目のパルスがSPC/OSCに到達しそのライジングエッジでデータをL2からLaunchする

(3) PLLから2つ目のパルスがSPC/OSCに到達しそのライジングエッジでDピンに到達したテスト結果をL2にCaptureする。その後、このL2の値をスキャンによって観測する。

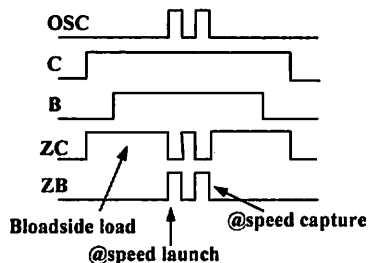


Figure 3: LOC wave form

3.3. LSSD 設計で実現する LSLO

LSSD設計でLSLOを実現するために、SPCにラッチ回路を追加するという変更を加える。この手法は[3]のようにUS特許が出願されているが、基本的にはSRラッチをFig4に示すようにSPCに追加し、Broadside Load信号をマスクするものである。その様子をFig5にSPCの各ピンの波形で示したが、点線で示されたZC='1'

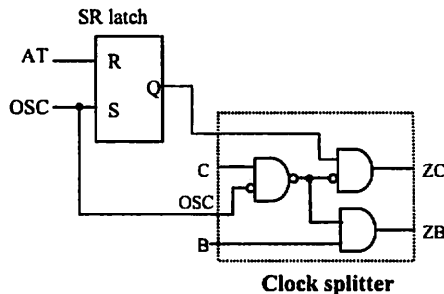


Figure 4: Modified clock splitter for LSLO

が消えるわけである。Launch-CaptureはCLOと同様にZBの最初のパルスのライジングエッジからZBの2回目のパルスのライジングエッジで行われる。重要なのはLaunchされるテストデータはスキャン経路でテスターから直接セットされた値であるということである。

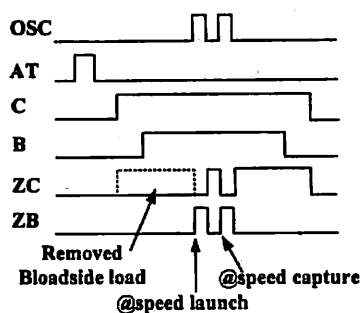


Figure 5: Loc wave form

3.4. CLOとLSLOの試行結果、MUXSCANとの比較

この2つの方法でCadence Encounterを使ってATPGを行ってみると、CLOが約10日かかって60%ほどのテストカバレッジに到達したのに対して、LSLOは2日ほどで65%ほどに達した。その様子をFig6に示した。両者のLaunchされるテストデータが決定される仕組みを考えれば、この結果は妥当であろう。つまり、LSLOはテスターから直接テストデータをスキャンを介してラッチにセットできるが、CLOの場合スキャンを介してラッチに一旦データをセットした後、これを、組み合わせ回路を経由してBroadside Loadで1ラッチに取り込む、そしてこれがテストデータとなりLaunchされる。Fig7はCLOの場合の本テスト時のデータの流れを概念的に説明したものであるが、ScanChain iにスキャンからセットされたデータは下流の組み合わせ回路を伝播してScanChain i+1のフリップフロップに、Broadside Loadされ、その後PLLから供給されるクロックによってScanChain i+1 → i+2間でLaunch、Captureが行われる。つまり、ScanChain i+2でCaptureされる値はScanChain iにテスターからセットした値が、2段の組み合わせ回路を伝播してきたものである。一方Fig8に示されるLSLOでは、ScanChain i+2でCaptureされる値はScanChain i+1にスキャンを介して直接テスターでセットされた値であり、1段の組み合わせ回路を伝播した値である。同様にScanChain i+1でキャプチャーするのはScanChain iにスキャンを介して直接セットされた値によってドライブされた値である。例えば、ScanChain iの3番目のラッチの出力はFig7ではScanChain i+2の3つのラッチのCaptureデータに影響するが、Fig8ではScanChain i+1の3つのラッチのCaptureデータに影響するだけである。つまりテストデータとテスト結果の関係は、LSLOでは組み合わせ回路の入出力問題であるのに対して、CLOは1段の順序回路の入出力の問題である。問題の複雑度はおよそ3-4倍程度あるであろう。そして、これがCPUのランタイム

の差となって現れており、ATPGのランタイム的にLSLOのほうがCLOよりメリットがずいぶん大きい。

次にMuxscanと比較すると、LSLOはLSSDならばSICにラッチを追加すれば簡単に実現できるが、MUXSCANでLSLOを実現しようとする、Fig9, 10に示すようなマスターラッチのクロックをゲートする方法か、Fig11のようなSEをPLLのクロックに同期して高速に'0'-'1'に遷移させる方法になると思われる。1番目の方法は各フリップフロップにセットリセットラッチを追加するので、面積へのオーバーヘッドが大きくなるが、そのかわりLSSDと同様にタイミングの作りこみの問題は少ない。2番目の方法はクロック周波数が速くなると、SEのセットアップホールドタイムをミートするのが困難になるだろう。

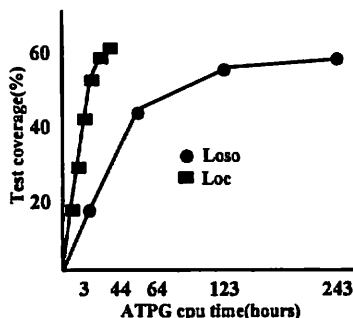


Figure 6: Loc vs Loso cpu time

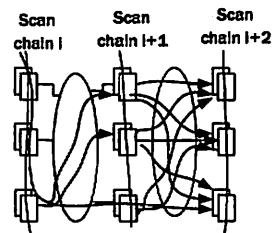


Figure 7: Data flow in LCO test

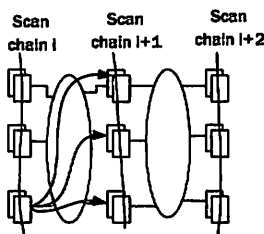


Figure 8: Data flow in LSLO test

3.5. アットスピードテストの効果

0.13μのASICにCLOを適用した際の結果は以下のとおりである。従来のテスト(ACディレイテストなど)をパスして出荷し、システムテストでフェイルしてリターンされてきたModuleにアットスピードテスト:

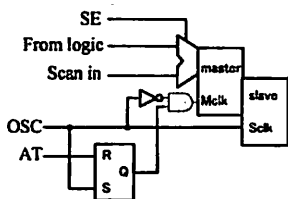


Figure 9: Muxscan flipflop for LOSO

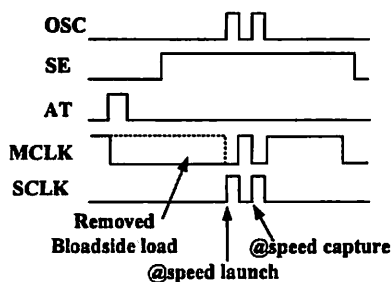


Figure 10: LOSO sequence in MUXSCAN

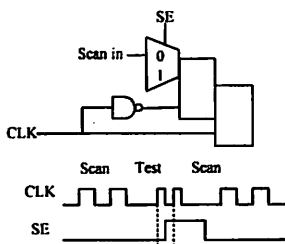


Figure 11: Test sequence in Muxscan

実施した結果、40%がフェイルした。つまり、アットスピードテストを行えば40%のリターン率改善が期待できる。また、アットスピードテストのテストカバレッジは50%だったので、仮に、残り50%の回路をテストできれば、単純に計算すれば80%を出荷時にテストでき、リターン率は80%改善する。

4. ソースシンクロナス DDR 回路をオンチップ PLL で高速テストする方法

4.1. はじめに

アットスピードテストをLSSDで実現する方法を検討していた際、ソースシンクロナスのDDRインターフェイス（以下SSDDR）のテストにオンチップPLLを使う方法も考えた。このインターフェイスはDRAMや、CPUのローカルバスなど、高速性が要求されるあらゆるModule間に使われているわけであるが、年々そのスピードが速くなっており、その結果IBMASICではこのタイミングを作りこむのに、常に大変な苦勞をしている。他のASICベンダーも同様の状況であろうと推察する。システムレベルでデバッグする際も、このインターフ

ェイスに問題があると苦勞すると容易に想像できる。たとえばFig12のようにCHIP#1からCHIP#2に'1'を2ビット送る際に、途中に問題があっても、受け側は50%の確率で'1'か'0'が観測されるわけで、いくつかのビットは“正しく”転送されてしまいこれがデバッグを時に困難にしてしまう。

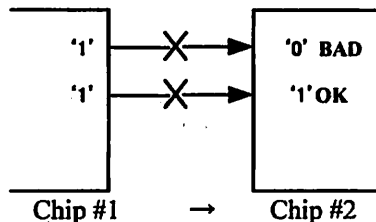


Figure 12: Problem in SSDDR interface

本インターフェイスの典型的な回路構成はFig13のようになると思われる。CHIP#1からCHIP#2にデータとクロックが転送されるが、両者の出力のスキューを描えるために、フリップフロップで一度取り直して出力している。CHIP#2は受け取ったクロックを1/4波長ずらして入力フリップフロップに供給している。

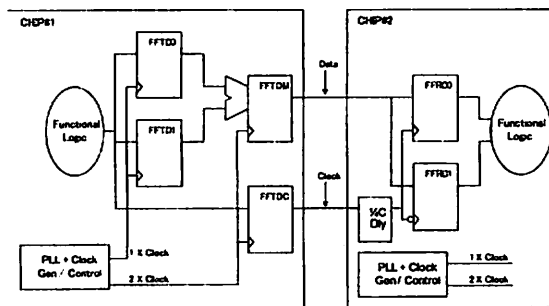


Figure 13: Typical SSDDR interface implementation

この回路に、本テストのための回路を入れた概念図をFig14に示す。挿入した回路はグレイで示してあるが、CHIP#1にテストパターンをアプライする回路を、CHIP#2にテスト結果をオブザーブする回路が追加される。そして、CHIP#1のFFTD0, 1は初期値をJTADTDRによってセットし、CHIP#2のFFRD0, 1がCaptureした値はJTADTDRを介してTDOから観測されるようにする。

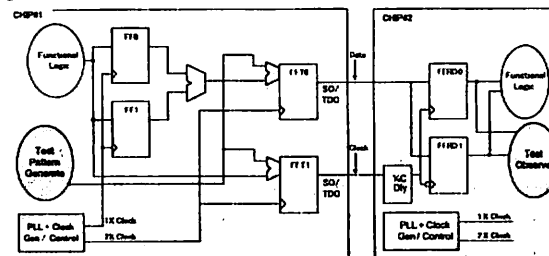


Figure 14: SSDDR interface with test circuit inserted

Receive側のオブザーブ用のTDRの実装は容易であ

るので、以降では詳細な説明はしない。そして主に Transmitter側の回路実装について、MUXSCAN設計、LSSD設計でどう実現するか考える。

4.2. MUXSCAN 設計での実現

MUXSCANで、Fig14を実現した図が、Fig15である。FF0,1はアウトプットラッチとTDRを兼用したものであり、本テスト用にトグルするためのセレクターとインバータ1つが追加された回路構成になり、また、システムクロックとTckを切り替えるセレクターがクロックラインに挿入されている。ClockDR[4]でTDIからデータをシフト(ShiftDR[4])して、FF0,1に初期データをセットした後、FF0,1をトグルモードに切り替える。そして、PLLから必要な数のパルスを与えて、データとクロックをCHIP#2に転送する。テストの時の波形はFig16に示した。はじめにCHIP#1はShiftDRで初期値をFF0,FF1にセットし、次にPLLから2発のクロックを出す。するとデータとクロックパルス1つがCHIP#2に送られる。

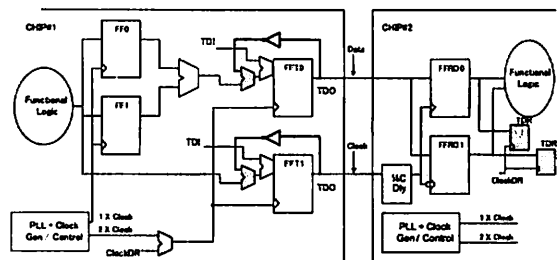


Figure 15 SSSDR with test circuit inserted in muxscan

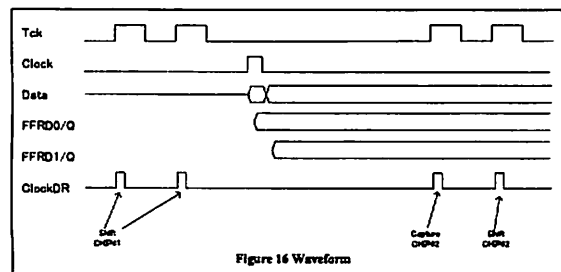


Figure 16 Waveform

CHIP#2ではFFRD0,1が転送データをCaptureし、この結果をJTAGインターフェイスにてTDRに取り込み、ShiftDRでTDOから観測する。

4.3. LSSD 設計での実現

次に、LSSD設計で本テストを実現した回路のうちCHIP#1のTransmitter側をFig17,18に示した。Fig18はLSSD+SPCを結合して1つのセルにしたものである。C1, C2信号は[4]のFig A-2にあるとおりTckのライジングエッジ、フォールディングエッジから作られるパルスである。このテストのシーケンスの波形をFig19に示した。MUXSCANの例との相違は、LSSDラッチは、TDIのポートとファンクションデータのポートがそれぞれ別に存在

するために、Clockラインにセレクターを入れてTckとシステムクロックを切り替える必要がないことである。

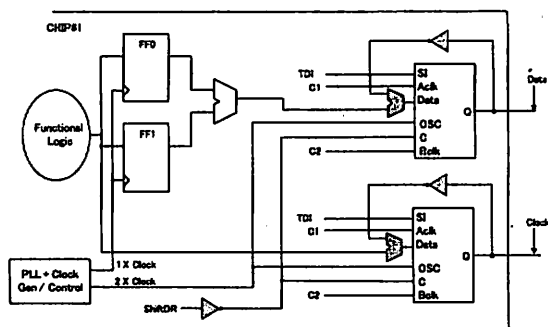


Figure 17 Transmitter with test circuit inserted in LSSD

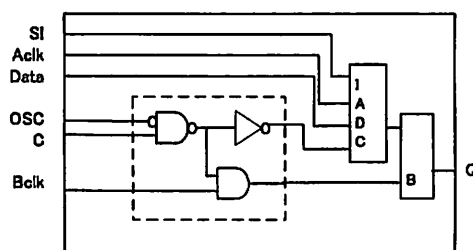


Figure 18 Clock splitter + LSSD

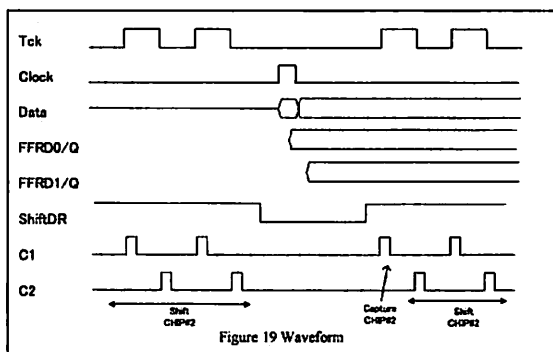


Figure 19 Waveform

5. まとめ

以上、オンチップPLLを用いた2つのテスト方法を説明した。高速スキャンテストはCLO方式とLSLO方式があり、LSSD設計ではLSLOが容易に実現できること、一方MUXSCAN設計ではLSSDと同様にマスターラッチのクロックをゲートする方法とスキャンイネーブルを高速で切り替える方法を示したが、前者はひとつのフリップフロップに1つのセットリセットラッチを追加する必要があり、後者は、クロックスピードが高速になると、タイミング制約をミートするのが大変困難にな

りそうである。こういった点を考えると、“アットスピードテスト”という観点からは、LSSD設計も悪くないテスト設計方法である。IBMASICは 0.65μ テクノロジーから標準テストとしてCLOを行う予定であるが、LSLOも実験的に実施を続け、どちらのほうが良いのか、テスト品質、コストの両面で引き続き検討を続けていくことになるだろう。いずれにしろ、試行結果からも明らかのようにアットスピードテストは出荷品質を改善する有効なテストであり、今後、さらに改良などされていくであろう。

“3.4”までで説明した方法は、同一クロックドメイン内のラッチ-ラッチのテストである。しかし、現実のASICは複数のクロックドメインが存在し、クロックドメイン間の信号も、当然タイミング制約が存在しアットスピードでテストしたいところであるが、それをどう実現するかは今後の課題であろう。Register Fileは上記で試行してみたところ、テストカバレッジが20-30%と極端に低かった。アドレスデコード回路のために一度に一つのアドレスにしか書き込めないためであろうと思われる。これとSRAMはBistで処理したほうがいだろう。IBMASICも 0.65μ 以降はアットスピードBISTを行う予定である。また、IOのオープンバスのテストも今後の課題であろう。IBMASICはACディレイテストをこれらに対して引き続き使う予定である。

そして、1149JTAGのインターフェイスを利用してソースシンクロナスDDRをアットスピードでテストする方法を、オンチップPLLを使ったテストの応用として提案した。このインターフェイスは、回路構成自体は非常に単純なのであるが、タイミング仕様をミートするのは大変難しい。それにもかかわらず、1149.1や1149.6では単に接続だけのテストだけしか行うことができない。1149標準テストが決められた当時から比べてModule IOのスピードは格段に上がっているのに、その変化に対応するために1149.1や6をさらに一步進めた本提案のような、タイミング的な接続性のテストを含めたものが必要なのではないだろうか。そしてこれをASICベンダーが自動挿入すれば、機能回路がどのように設計されたかにかかわらず、この方法を使ってビットごとの転送を観測できるようになるので、システムレベルのデバッグの効率もあがるであろう。また、ボードのテストにも変化をもたらすかもしれない。従来、ボードテストとしてシステムソフトウェアを実行していると思うが、本テストによってModule間のタイミングを含んだ接続性がテストされれば、

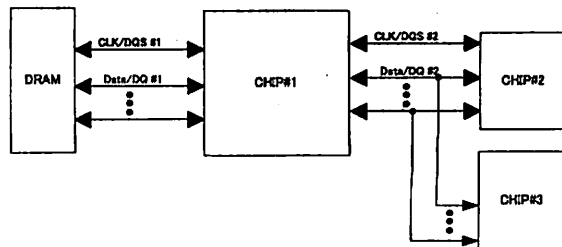


Figure 20 SourceSync interface among all modules

例えば極端な例であるが、Fig20に示すように、Module IOが全てDDRであろうがSDRであろうがソースシンクロナスであれば、本テストを実行するだけでボードテストとできるのではないだろうか。いくつかのDRAMベンダを調べてみると、DDR SDRAMは1149JTAG回路が実装されていないようだが、RLDRAM, QDRAMは1149JTAGを実装しているようである。これらのDRAMが将来、この方法をサポートしたならば、その可能性も出てくるのではないだろうか。

文 献

- [1] P. Gillis et al. Low overhead delay testing of ASICs. Proc. Int. Test Conf., pp. 534-542, 2004
- [2] G. Vandling, Modeling and testing the Gekko Microprocessor, and IBM powerpc derivative for NINTENDO, Proc. Int. Test Conf., pp. 593-599, 2002.
- [3] B. Koenemann, W.H. McAnney and M.L. Shulman. USpatent 1991-0008-11205, Dec 1991.
- [4] IEEE Standard Test Access Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-1990.