

# JTAG/ BOUNDARY SCAN – WAS KANN ES, UND WAS MUSS MAN DAFÜR TUN?

Dipl.-Ing. Mario Berger, GÖPEL electronic GmbH, Jena

## 1 Das Testen einer integrierten Schaltung

Seit es integrierte Schaltungen gibt besteht die Notwendigkeit deren Funktion zu überprüfen. Handelt es sich dabei um digitale Schaltungen, gestaltet sich deren Test relativ einfach: Man legt an den Eingängen nacheinander sämtliche möglichen Testvektoren nacheinander an und vergleicht die Reaktion der Schaltung an den Ausgängen (IST-Ausgangsvektor) mit den erwarteten Mustern (SOLL-Ausgangsvektor). Ergibt der Vergleich keine Abweichung, ist die Schaltung in Ordnung.

Für ein einfaches UND-Gatter mit zwei Eingängen ist die Anzahl der Testvektoren noch überschaubar. Nach Moore und McCluskey errechnet sich diese nach der Formel:

$$Q=2^{(x+y)}$$

Q = Mindestanzahl der Testvektoren

x = Zahl der Eingänge

y = Zahl der Speicherelemente (bei sequentiellen Schaltungsteilen)

Da ein UND-Gatter gewöhnlich keinerlei Speicherelemente beinhaltet, kommt man auf eine Anzahl von gerade einmal vier notwendigen Testvektoren, was eine leicht zu handhabende Menge darstellt. Führt man diese Berechnung allerdings für eine Schaltung mit einer angenommenen Anzahl von 25 Eingängen und 50 Speicherelementen durch, wird man sofort erkennen, mit welchem Problem sich die Ingenieure in den Chip Entwicklungen in den 70er Jahren konfrontiert sahen.

Bei IBM wurde Anfang der 70er Jahre mit der Erfindung des ersten „Level Sensitive Scan Design (LSSD)“ Verfahrens eine bahnbrechende Idee geboren. Hierzu werden die in einem Chip vorhandenen Speicherelemente in Ihrer Funktion erweitert. Sie bekommen vier zusätzliche Anschlüsse: einen Eingang (IN), einen Ausgang (OUT) und zwei Clocks (A und B); siehe Abbildung 1. Mit diesen zusätzlichen Ressourcen ist es möglich ebenfalls auf die Ein- und Ausgänge der Speicherelemente zuzugreifen.

Anfang der 80er Jahre begann man auch auf Baugruppen-Ebene, die Problematik der „zunehmenden Komplexität der Baugruppen mit immer höheren Packungsdichten“ anzugehen. Als eine der ersten beschäftigte sich die 1985 gegründete „Joint European Test Action Group“ mit dem Thema. Damals bestand diese Gruppe aus Test-Ingenieuren der

großen europäischen Chip Hersteller. 1986 traten ihr weitere Firmen aus Nordamerika bei, und die Gruppe benannte sich in „Joint Test Action Group (JTAG)“ um.

Die JTAG konstruierte dann ein Verfahren, welches sich stark an das von Ed Eichelberger entwickelte LSSD Verfahren anlehnte. So definiert es ebenfalls Speicherelemente innerhalb eines Chips, welche miteinander in einer Schiebekette verbunden sind. Nur mit dem Unterschied, das diese Speicherelemente nun zusätzlich an der Peripherie, sprich an der Bauteilgrenze, der „Boundary“, eingebracht wurden.

Das entwickelte Verfahren nannte man deshalb auch Boundary Scan. Standardisiert wurde es im Jahre 1990 vom „Institute of Electrical and Electronics Engineers (IEEE)“ als 1149.1 „Standard Test Access Port and Boundary Scan Architecture“.

## **2 Der Boundary Scan Standard IEEE1149.1**

Der Boundary Scan Standard IEEE1149.1 beschreibt den statischen, digitalen Verbindungstest. Spricht man heute von Boundary Scan oder JTAG, so bezieht sich dies immer auf den Standard IEEE1149.1.

Im Standard selbst ist der Aufbau eines Boundary Scan-fähigen Bausteins dargelegt, wie auch die Beschreibungssprache, die „Boundary Scan Description Language (BSDL)“, welche die für jeden Baustein einzigartige Boundary Scan Ressourcen offen legt.

Zum inneren Aufbau eines Boundary Scan Bausteins definiert der Standard IEEE1149.1 hierzu vier wesentliche Bestandteile, über die ein Boundary Scan-fähiges Bauteil verfügen muss:

- ein Test Access Port (TAP)
- ein TAP Controller
- ein Befehlsregister („Instrucion register“)
- ein oder mehrere Datenregister

### ***Test Access Port (TAP)***

Das „Test Access Port“ stellt die Schnittstelle zwischen der im Baustein befindlichen Boundary Scan Logik und der Außenwelt dar. Es sind drei Eingänge (zzgl. eines optionalen vierten) und ein Ausgang beschrieben. Die Eingänge sind:

- Test Clock (TCK)
- Test Mode Select (TMS)
- Test Data Input (TDI)
- *Test Reset (/TRST) - optional*

Der Ausgang ist:

- Test Data Output (TDO)

Bei den beiden Signalen TCK und TMS sowie beim optionalen /TRST Signal handelt es sich um Broadcast Signale, wohingegen das TDI hin zum TDO eine serielle Kette bilden, die sogenannte Scankette oder auch Scanpfad (siehe Abbildung 2). Auf Baugruppen-Ebene spricht man vom Testbus.

Das Geniale bei diesem Konstrukt ist, dass nie mehr als vier (optional fünf) Signalleitungen benötigt werden, unabhängig davon, wie viele Bausteine in die Scankette geschaltet werden.

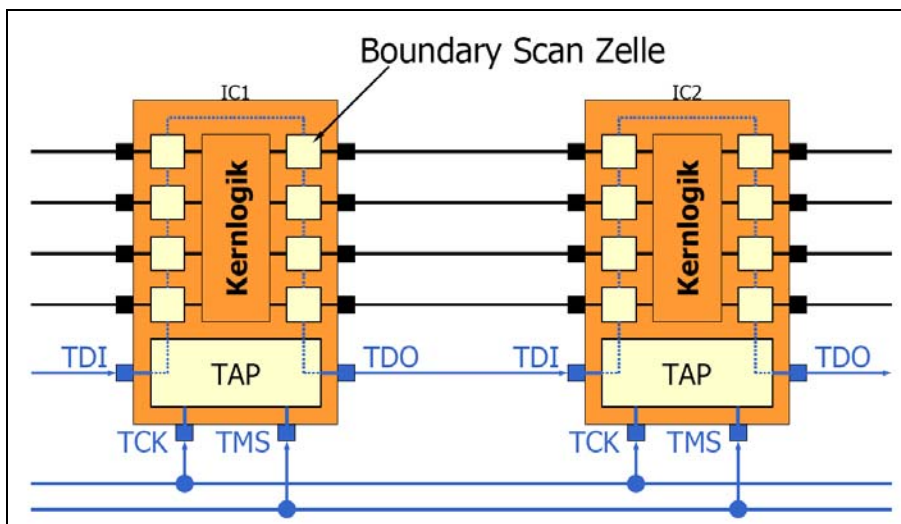


Abb1: Testbus Verdrahtung von zwei Boundary Scan Bausteinen

Im Boundary Scan Baustein sind der „Test Clock“, das „Test Mode Select“ wie auch der „Test Reset“ direkt mit dem „TAP Controller“, sprich statisch, verbunden. Die Signale sind einzig und allein verantwortlich für den dessen Zustand. Das bedeutet gleichzeitig auch, dass sich alle Boundary Scan Bausteine einer Scankette immer im gleichen TAP Zustand befinden. Das bedeutet nicht automatisch, dass sich auch alle Bausteine immer im gleichen Betriebsmode/ Befehl befinden müssen.

### **TAP Controller**

Der „TAP Controller“ ist verantwortlich für die komplette Steuerung der Boundary Scan Logik im Baustein. D.h., er ist unter anderem dafür verantwortlich, ob eine Boundary Scan Zelle (siehe Kapitel *Boundary Scan Zelle*) aktiviert bzw. deaktiviert wird und ob diese gerade messen oder treiben soll.

Herzstück des „TAP Controllers“ ist die „TAP state machine“. Die darin enthaltenen Zustände haben einen unterschiedlichen Einfluss auf die Steuerung der internen Boundary Scan Logik.

### ***Befehlsregister***

Das Befehlsregister („Instruction register“) entscheidet über den Betriebsmode des Boundary Scan Bausteins, der wiederum Einfluss auf die Steuerung der Boundary Scan Zellen wie auch auf die Auswahl des aktuell in die Scankette (Register zwischen TDI und TDO) geschalteten Datenregisters hat. Der Standard IEEE1149.1 definiert drei zwingend erforderliche Befehle:

- BYPASS
- SAMPLE/PRELOAD
- EXTEST

Jedem Befehl ist ein entsprechender Befehlscode (Bitcode) hinterlegt. Dieser kann von jedem Chip Hersteller frei definiert werden (ausgenommen ist der BYPASS Befehl; dieser muss sich vollständig aus Einsen zusammensetzen). Auch die Länge des Befehlsregisters ist beliebig wählbar. Eine beispielhafte Zuordnung stellt Tabelle 2 dar. Dabei wurde die Länge des Befehlsregisters auf zwei Bit definiert.

Betriebsmode/Befehl	Befehlscode (binär)
BYPASS	11
SAMPLE/PRELOAD	01
EXTEST	00

*Tabelle 1: Beispiel für eine Befehlsregister Definition*

### ***Datenregister***

In einem Boundary Scan-fähigen Baustein können mehrere Datenregister enthalten sein. Diese dienen dazu, Informationen im Baustein abzulegen oder auch daraus auszulesen.

Der Standard IEEE1149.1 beschreibt im Minimum zwei zwingend notwendige Datenregister:

- bypass
- boundary-scan

Auch hier sind wieder zusätzliche Register möglich, wie das „device identification“ oder umgangssprachlich auch „idcode“ Register.

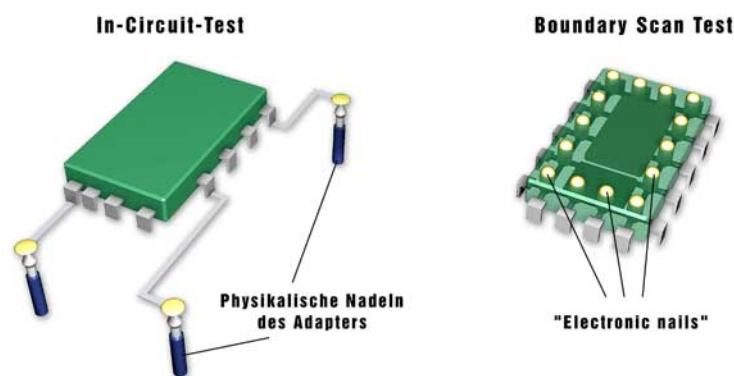
Das „bypass“ Register stellt dabei die Möglichkeit dar, den Baustein aus einem Verbund von Boundary Scan Bausteinen zu lösen, sprich diesen zu umfahren, ihn zu „bypassen“. Es hat dazu eine minimale Länge von nur einem Bit. Der Wert des Bits ist unveränderlich und definiert mit 0.

Das für ein späteres Testen wesentlich interessantere Datenregister ist jedoch das „boundary-scan“ Register, welches die Aneinanderreihung der einzelnen Boundary Scan Zellen darstellt. Da jeder Chip eine andere Anzahl an Boundary Scan Zellen besitzt, ist die Länge dieses Registers variabel.

### **Boundary Scan Zelle**

Die Boundary Scan Zelle ist der Hauptbestandteil des Boundary Scan Testverfahrens. Alle bisher beschriebenen Konstrukte dienen einzig und allein der korrekten Ansteuerung der einzelnen Boundary Scan Zellen.

Die Boundary Scan Zelle ist die geniale Möglichkeit, den Bauteilpin eines Bausteins gelöst von dessen normaler Funktion zu kontrollieren, d.h. einen bestimmten Pegel zu treiben oder auch zu messen. Zu diesem Zweck befindet sich die Boundary Scan Zelle zwischen der Kernlogik des Bausteins und dessen Peripherie (Ausgangstreiber, Eingangstreiber). Aufgrund der funktionellen Ähnlichkeit zu den physischen Antastnadeln des In-Circuit Testverfahren, welche dort den Zugriff auf die einzelnen Testpunkte realisieren, spricht man bei den Boundary Scan Zellen auch von „electronic nails“.



*Bild 2: Vergleich der Testverfahren ICT und Boundary Scan*

Der interne Aufbau einer einzelnen Boundary Scan Zelle kann sich sehr unterschiedlich gestalten. Der Standard IEEE1149.1 beschreibt in der Fassung 2001 allein zehn unterschiedliche Zell Typen (BC\_1 bis BC\_10). Eigene Strukturen sind zusätzlich möglich. Der Aufbau ist oft sehr ähnlich.

### ***Boundary Scan Description Language (BSDL)***

Jeder Boundary Scan fähige Baustein hat seine ganz spezielle Boundary Scan Struktur, ohne deren Kenntnis ein Testingenieur bzw. eine Testsoftware nicht in der Lage wäre, sinnvoll mit dem Baustein zu arbeiten. Der Standard IEEE1149.1 schreibt zwar einiges zwingend vor, lässt aber auch genügend Freiräume für Individualität. Die ist auch notwendig, was gerade am Beispiel der Struktur/ Anzahl der Boundary Scan Zellen deutlich wird: Ein Baustein mit 20 Anschlusspins verfügt sinnvollerweise über eine geringere Anzahl an Zellen verglichen mit einem Baustein mit 1.500 Pins.

Um diese Individualität zu beschreiben, wurde die „Boundary Scan Description Language (BSDL)“ entwickelt. Sie ist das Verständigungsmedium zwischen Chip Hersteller (der allein das „Innenleben“ seines Chips kennen kann) und dem Testingenieur (der dieses „Innenleben“ in seinem speziellen Einsatzfall verwenden möchte). Es handelt sich dabei um eine Datei.

So liefert das BSDL-File u.a. Angaben über:

- verfügbare Testbussignale (insbesondere Angaben über das Vorhandensein des optionalen /TRST Signals und auch zur maximalen TCK Frequenz, bis zu der der Baustein betrieben werden kann)
- mögliche „Compliance“ Pins
- das Befehlsregister (verfügbare Befehle inkl. deren Bitcode; Länge des Befehlsregisters)
- die Datenregister (verfügbare Datenregister inkl. möglicher voreingestellter Werte, z.B. IDCODE des Bausteins)
- die Struktur der Boundary Scan Zellen (Anzahl, Typ, Funktion, Zuordnung zum Bauteilpin)

### **3 Möglichkeiten und Grenzen des Standards IEEE1149.1**

Der statische, digitale Verbindungstest nach Standard IEEE1149.1 ermöglicht alles, was im digitalen Bereich liegt und nicht zeitkritisch ist. So können Widerstände auf ihre Anwesenheit, Quarze, Treiber-Bausteine, Logik-Gatter, Reset-Bausteine und sogar RAM-Bausteine, Flash-Bausteine (parallele wie auch serielle) getestet werden, eben alles, was digital möglich ist. Bei letzteren werden zum Beispiel einfach die notwendigen Schreib- und Leseprotokolle über die Pins des Boundary Scan Bausteins nachgeahmt. Das funktioniert dann genau wie ein Funktionstest, nur dass es infolge der seriellen Boundary Scan Kette langsamer erfolgt.

Und da liegt dann auch die Limitation des Testverfahrens: die maximal mögliche Schalt-/ Messfrequenz an den Bauteilpins. Diese ergibt sich aus der Anzahl der Boundary Scan Zellen (also der Länge des „boundary-scan“ Registers) sowie der „Test Clock“ Frequenz. Dabei ist es egal, ob der Signalpegel nur eines oder mehrerer Bauteilpins geändert werden soll; in jedem Fall muss immer durch ALLE Zellen geschoben werden.

Bei einem mittelgroßen, Boundary Scan-fähigen Baustein mit 500 Boundary Scan-Zellen und einer typischen TCK Frequenz von 10 MHz ergibt sich so eine Dauer von 50  $\mu\text{s}$  für einen Schiebevorgang. Mit einem Schiebevorgang kann allerdings nur ein einziger Signalwechsel am Bausteinpin ausgelöst werden. Für die entgegengesetzte Flanke wird ein weiterer Schiebevorgang benötigt, wodurch sich eine maximal erreichbare Frequenz am Bausteinpin von  $100 \mu\text{s}^{-1}$ , also 10kHz ergibt.

#### 4 Was können moderne Tools?

Nach heutigen Erkenntnissen ergeben sich einige Grundansprüche an ein Boundary Scan Testsystem. So möchte sich ein Anwender sicher keine Gedanken über das richtige Schalten des „Test Mode Select“ Signals machen, um in den richtigen Graphen in der „TAP state machine“ zu kommen. Und er möchte sich bestimmt ganz und gar nicht mit einer „TAP state machine“ herumplagen. Allenfalls den Betriebsmode für die Boundary Scan Bausteine vorgeben, mehr aber nicht.

Glücklicherweise werden einem diese lästigen Arbeiten von modernen Tools tatsächlich abgenommen. Aber was ist überhaupt gemeint, wenn man über ein Boundary Scan Tool spricht? Ein Boundary Scan Testsystem besteht immer aus Hardware und Software. Die Hardware muss dabei „einfach nur“ in der Lage sein, die TAP Signale anzusteuern. Diese Grundfunktionalität hat jede Boundary Scan Hardware weltweit (es gibt allerdings zum Teil gravierende Unterschiede in der Performance, also echten Durchsatz, und der Flexibilität).



*Bild 3: Hardware – Boundary Scan Controller von GÖPEL electronic (SFX-TSL1149.x)*

Die Unterschiede der jeweiligen Hersteller liegen in der Software, und deshalb ist in aller Regel die Software gemeint, wenn man über ein Boundary Scan Tool spricht.

Man darf von einer modernen Boundary Scan Software erwarten, dass sie die notwendigen Testvektoren automatisch erzeugt und dass sie den Operator im Prüffeld im Fehlerfall möglichst einfach zur fehlerhaften Stelle des Prüflings führt, also eine bestmögliche Diagnose. Denkt man jedoch an die Integration des Boundary Scan Tests in die

Fertigungslinie oder in ein anderes Testsystem, so ergibt sich der Anspruch, dass ein modernes Boundary Scan Tool über entsprechende Schnittstellen verfügt.

## 5 Design For Testability (DFT)

Die besten Boundary Scan Testsysteme mit den mächtigsten ATPGs können nichts ausrichten, wenn gewisse Design Regeln schon beim Schaltplanentwurf oder noch einen Schritt eher, bei der Bauteilauswahl, nicht eingehalten wurden. Die folgenden Kriterien zeigen eine begrenzte Auswahl der wohl wichtigsten „Design For Testability“ Regeln:

### *Compliance Pattern*

Bei Boundary Scan-fähigen Bausteinen ist es üblich, sich die TAP Pins mit anderen Funktionen z.B. zum Debuggen zu teilen. Aus diesem Grund verfügt ein solcher Baustein in aller Regel über einen Pin, der über den Zweck entscheidet. Solch ein Pin könnte z.B. JTAG#/DEBUG heißen und würde bei einem High den Debug Mode aktivieren. In diesem Beispiel muss also zwingend ein Low am Pin angelegt werden, damit dieser mit Boundary Scan getestet werden kann.

### *Testbusabschluss*

Für eine schnelle Testabarbeitung ist ein guter Testbusabschluss unerlässlich. Als groben Anhaltspunkt kann man davon ausgehen, dass die Testzeit sich direkt proportional zur Frequenz des „Test Clock“ verhält.

Heutige Testsysteme sind in der Lage, das TCK Signal mit 80 oder gar 100 MHz zu betreiben. Damit ist offensichtlich, dass große Sorgfalt bei der Verdrahtung der TAP Signale angebracht ist.

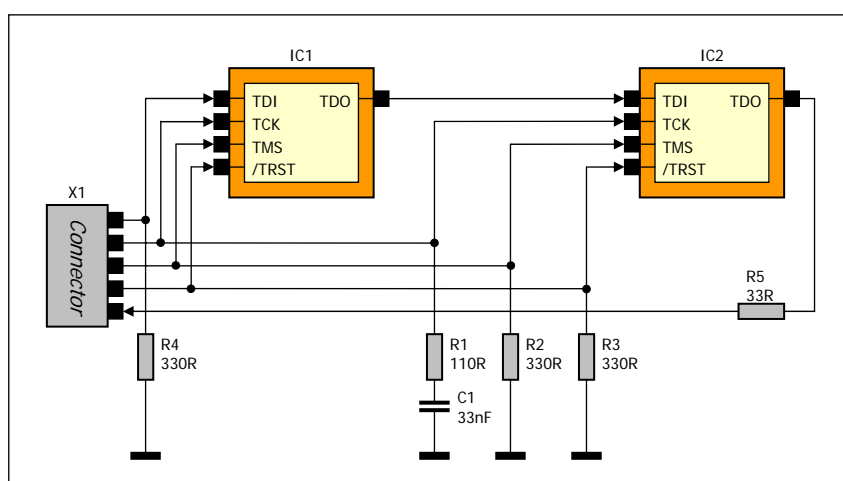


Abb 4: Testbusabschluss



### **Flexible Scankette**

Es ist übliche, Baugruppen in verschiedenen Bestückungsvarianten zu fertigen. Vorsicht ist dann geboten, wenn solch eine Bestückungsvariante auch die Boundary Scan-fähigen Bausteine betrifft. Dann kann es passieren, dass ein Baustein in der Scankette fehlt und somit der serielle Pfad (TDI → TDO) unterbrochen ist. Das führt natürlich unweigerlich zum Totalausfall.

### **5.1 Access = Success**

„Access = Success“; diese „Design For Testability“ Grundregel gilt beim Boundary Scan Testverfahren ebenso, wie bei dem klassischen In-Circuit Testverfahren. Nur ist die Umsetzung in beiden Fällen eine völlig andere.

Beim In-Circuit Test heißt es, Testpunkte zu setzen wo es nur geht. Bei Boundary Scan hingegen „schlummern“ diese in den Boundary Scan Bausteinen zum Teil ungenutzt in Form von nicht verdrahteten Bauteilspins. Dabei handelt es sich üblicherweise um in der „normalen“ Funktion der Baugruppe nicht benötigte Pins eines Bausteins (gerade bei programmierbaren Logikbausteinen).

Diese ungenutzten „Testpunkte“ einzusetzen um die Testabdeckung wesentlich zu erhöhen sollen zwei typische Beispiele aufgezeigt werden.

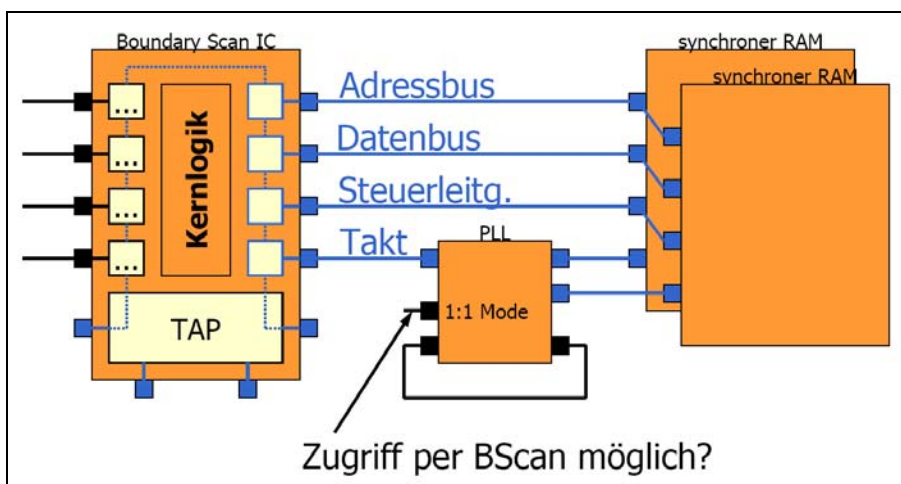


Abb 5: RAM Bank mit PLL

Abbildung 5 zeigt als erstes Beispiel eine RAM Bank, deren Takt von einer PLL verteilt wird. Um die RAM Bausteine mit Boundary Scan testen zu können ist jedoch ein statischer Zugriff auf sämtliche Bauteilpins erforderlich. Dies ist durch die PLL beim Takt Signal nicht gegeben, was zu einem großen Verlust bei der Testabdeckung führt.

Abbildung 6 zeigt ebenfalls, wie der Zugriff auf einen einzigen Bauteilpin über die Testbarkeit eines kompletten Bausteins entscheidet. Abbildung 15 zeigt einen Baustein mit integriertem NAND-Tree Test. Er ließe sich somit optimal per Boundary Scan testen. Einzige Bedingung ist dabei, dass der NAND-Tree Test mit einem bestimmten Signalpegel an einem vorgegebenen Bauteilpin aktiviert werden muss (in Abbildung 6 ist dieser mit „TEST“ benannt).

## **6 Standards der Zukunft**

Der Erfolg des JTAG/ Boundary Scan Standards IEEE1149.1 hat die Beteiligten beflügelt und zu vielen Ideen angeregt, das Testverfahren noch besser, noch grenzenloser zu machen. Aus der Vielzahl der neu entwickelten und zum Teil schon verabschiedeten Standards werden in diesem Kapitel zwei kurz vorgestellt.

### ***IEEE1149.4***

Der Durchbruch des Standards IEEE1149.4 würde vermutlich das Aus für das klassische In-Circuit Testverfahren bedeuten, denn hierbei handelt es sich um den „Mixed-Signal“ oder auch analogen Verbindungstest.

Das Verfahren ist auch hier sehr einfach gehalten. Zusätzlich zu den vier (oder optional fünf) TAP Signalen werden zwei „Analog Test Access Port (ATAP)“ Signale „Analog Test 1 (AT1)“ und „Analog Test 2 (AT2)“ benötigt. Diese beiden zusätzlichen Pins können unabhängig voneinander per Testbus-Befehl intern auf einen beliebigen Bauteilpin des IEEE1149.4 fähigen Baustein geschaltet werden.

Man könnte auch sagen, dass ein IEEE1149.4 fähiger Baustein über eine interne Relaismatrix verfügt, die per Testbus auf beliebige Pins geschaltet werden kann.

Wird nun an das ATAP eine entsprechende externe Messtechnik angeschlossen, hat man vom Prinzip her einen klassischen In-Circuit Tester (mit eingeschränkten Funktionen).

### ***IEEE1149.6***

Der Standard IEEE1149.6 ermöglicht das Testen serieller, digitaler High-Speed Verbindungen. So beschreibt er den „Advanced Digital Network“ Verbindungstest.

Ein großer Vorteil des Standards liegt darin, dass er mit den vorhandenen TAP Signalen auskommt. Er begnügt sich mit wenigen neuen Befehlen, einer etwas erweiterten Boundary Scan Zelle und einem integrierten Testmuster Generator.

Das Prinzip ist auch hier wieder äußerst simpel. Einige Pins des IEEE1149.6 fähigen Bausteins sind mit einem neuen Typ Boundary Scan Zelle verbunden. Diese verfügt im Gegensatz zu den „alten“ Zelltypen über einen speziellen Eingang, der mit dem internen Testmuster-Generator verbunden ist. Per Befehl schaltet die Boundary Scan Zelle auf den

neuen Eingang um, und der Testmuster-Generator sendet unabhängig vom „Test Clock“ Signal das Testmuster an die Boundary Scan Zelle und somit an den Bauteilpin.

Dies war zunächst die Sendeseite. Auf der Empfängerseite wird im gleichen Augenblick begonnen, die Testmuster vom Bauteilpin einzulesen und in einen Buffer zu schreiben. Anschließend erfolgt ein Vergleich mit dem vom Sender geschickten Testmuster und eine Gut/ Schlecht Aussage. Diese wird in einer Boundary Scan Zelle als 0 oder 1 abgelegt und kann vom Testsystem ausgelesen und ausgewertet werden.

## **7 Zusammenfassung**

JTAG/ Boundary Scan ist das wohl genialste elektrische Testverfahren. Es ist der Sprung vom physikalischen Zugriff auf die Leiterbahnen einer Baugruppe (wie es beim In-Circuit Testverfahren notwendig ist) mit all seinen physischen Grenzen hin zum elektrischen und somit grenzenlosen Zugriff. Dabei kommt das Testverfahren mit gerade einmal vier Steuerleitungen und einer Handvoll wichtiger „Design For Testability“ Regeln aus!

Spricht man heutzutage von JTAG oder Boundary Scan, bezieht man sich immer auf den Standard IEEE1149.1, also den statischen, digitalen Verbindungstest. Dessen Grenzen liegen im analogen Bereich wie auch im digitalen High-Speed Bereich. Doch auch hier erfolgen Weiterentwicklungen mit den beiden Standards IEEE1149.4 und IEEE1149.6.

Ein Boundary Scan Testentwickler muss sich heutzutage nicht mehr zwingend mit jedem einzelnen Detail des Verfahrens beschäftigen, da moderne, auf Bauteilmodellen basierende Tools den Großteil der Arbeiten für ihn übernehmen.