



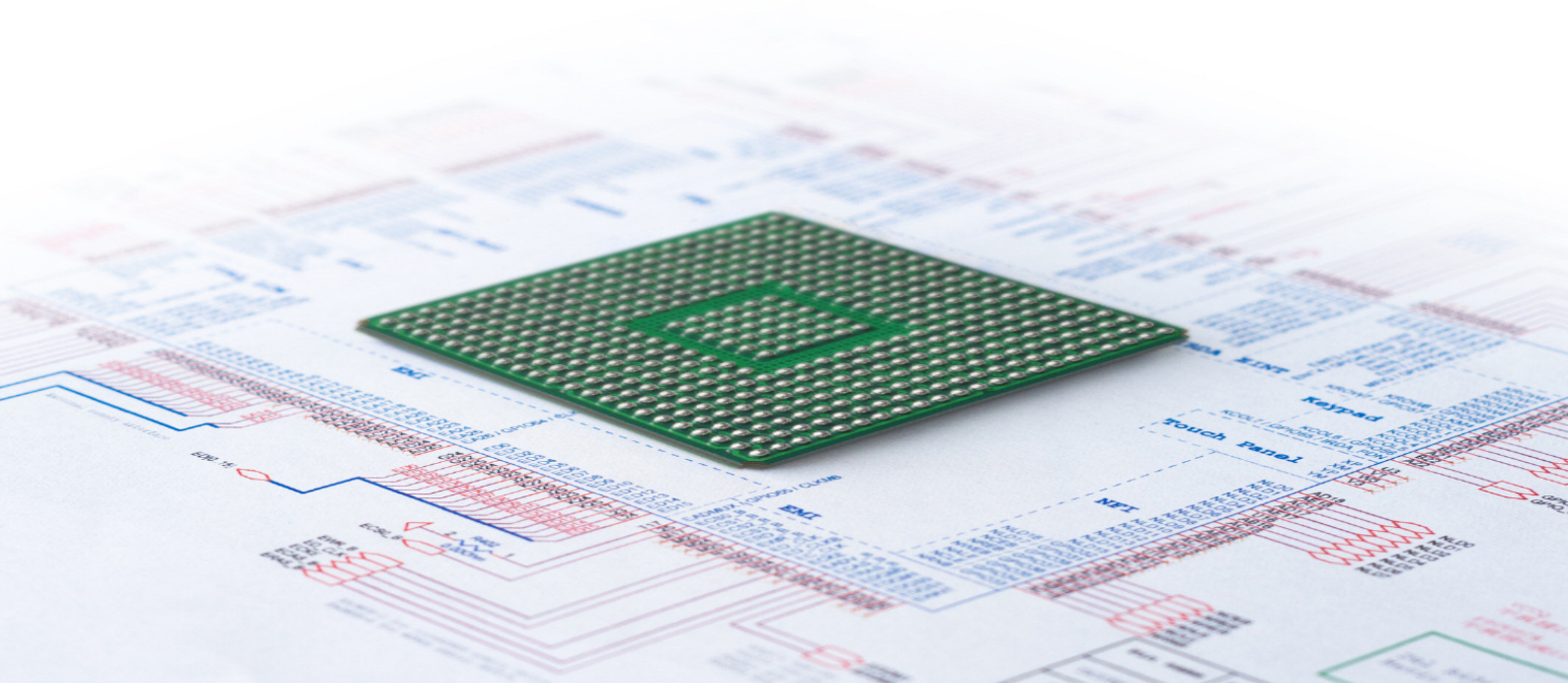
# Design for Test Richtlinien

---

Version 4.4

[enquiries@xjtag.com](mailto:enquiries@xjtag.com)

[www.xjtag.com](http://www.xjtag.com)



## Inhaltsverzeichnis

Einführung .....	3
Geben Sie JTAG-Komponenten an und verwenden Sie sie .....	3
Überprüfen Sie die BSDL-Dateien für JTAG-Komponenten .....	3
Stellen Sie sicher, dass die 1149.x-Ketten richtig designt und ausgelegt sind .....	3
Verwendung der korrekten Beendigung für alle TAP-Signale .....	4
Mehrere JTAG-Ketten .....	5
Konnektor-design .....	5
JTAG-Modus wählen .....	6
Beachten Sie den Takt zu synchronen Komponenten .....	6
Ziehen Sie einen Watchdog-Einsatz in Betracht .....	7
Testen mit nicht-konfigurierten FPGAs und CPLDs .....	7
Sicherstellen eines geringen Schiefbaus zwischen TAP-Signalen .....	7
Gewährung von Zugriff zur Unterstützung der Suche nach Scan-Kettenfehlern .....	7
Pufferung der TAP-Signale und Spannungspegel-Trennung .....	8
Nebenschluss optionaler 1149.x-konformer Komponenten .....	9
Fügen Sie der 1149.x-Kette Add-ons und optionale Boards hinzu .....	9
Einbezug von 1149.x-Kompatibilität bei Entwurf oder Spezifizierung von ASICs .....	9
Machen Sie programmierbare Abschnitte von der 1149.x-Kette aus zugänglich .....	10
Stellen Sie direkten Zugriff auf Programmiersignale zur Verfügung .....	10
Umschließen Sie Logikgruppen mit JTAG-Komponenten .....	11
Erweitern Sie 1149.x-Testen durch Steckverbinder .....	11
Ziehen Sie das Testen von Mehrfachleiterplatten als Einheit in Betracht .....	12
Testen Sie analoge Schaltungen .....	12
Verlassen Sie sich nicht auf programmierbare Pull-Widerstände .....	12
Ziehen Sie die Nutzung der Funktionalität von JTAG-Komponenten in Betracht .....	12
Verwenden Sie die Testfunktionalität in Nicht-JTAG-Komponenten .....	13
Interaktive tests .....	13
Programmieren Sie nicht-flüchtige Speicherkomponenten .....	13
Schnelle Flash-Programmierung .....	13
Glossar .....	14

## Einführung

Die folgenden Richtlinien geben Anregungen zur Verbesserung der Prüfbarkeit von Schaltungen mit XJTAG. Diese Richtlinien sollen nicht als strenge Regeln gelten. Die potenziellen Vorteile hinsichtlich der Prüfbarkeit sollten zusammen mit allen anderen eventuellen Auswirkungen berücksichtigt werden (z.B. Funktionalität, Komponentenkosten und Einteilung des Boardbereichs).

Es wird davon ausgegangen, dass die Leser dieses Dokuments ein Grundverständnis der IEEE-Standards 1149.1 und 1149.6 besitzen.

In diesem Dokument bezieht sich der Begriff „1149.x“ auf die IEEE 1149.x (JTAG) Boundary-Scan-Standards, mit „JTAG-Komponenten“ sind jene gemeint, die JTAG-Boundary-Scan implementieren und vollständig mit diesen Standards übereinstimmen, während „Nicht-JTAG-Komponenten“ jene meint, die den JTAG-Boundary-Scan nicht implementieren.

### ➤ Geben Sie JTAG-Komponenten an und verwenden Sie sie

Obwohl XJTAG in der Lage ist, die Netze zwischen JTAG-Komponenten und Nicht-JTAG-Komponenten zu testen, wird eine größere Abdeckung durch die Verwendung von möglichen JTAG-Komponenten erreicht. Je mehr JTAG-Komponenten in eine Schaltung integriert sind, desto größer ist die Anzahl der Knoten, die vollständig ausgeübt und getestet werden können.

Der 1149.6-Standard ermöglicht die Prüfung von kapazitiv gekoppelten High-Speed-unsymmetrischen oder symmetrischen Signalen, die nicht mit 1149.1 getestet werden können. Um diese Funktionalität nutzen zu können, ist darauf zu achten, dass auf diese Weise zu prüfende Netze auf jeder Seite des Koppelkondensators 1149.6-fähig sind.

### ➤ Überprüfen Sie die BSDL-Dateien für JTAG-Komponenten

Die Einhaltung der 1149.x-Standards erfordert, dass die Komponenten über eine BSDL-Datei (Boundary Scan Description Language) verfügt. Diese Datei beschreibt die spezifische 1149.x-Implementierung der Komponente. Stellen Sie sicher, dass diese Datei verfügbar ist (in der Regel direkt von der Website des Herstellers) und dass sie vollständig validiert wurde (BSDL-Syntaxprüfung ist integraler Bestandteil von XJTAG).

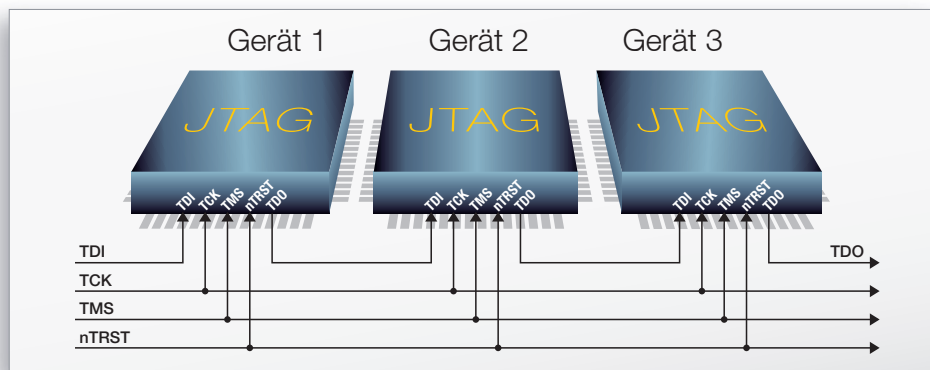
### ➤ Stellen Sie sicher, dass die 1149.x-Ketten richtig designt und ausgelegt sind

JTAG-Komponenten können über individuelle Anschlüsse verfügen, um Zugriff auf die TAP-Signale zu erhalten. Sie können aber auch in „Kaskaden“-Format mit dem TDO-Signal von einer Komponente verbunden werden, das mit dem TDI des nächsten verbunden ist.

Die TCK-, TMS- und (optional) nTRST-Signale müssen parallel mit allen im „Kaskaden“-Format angeschlossenen Komponenten verbunden sein. Es ist wichtig, dass nTRST-Signale nicht direkt mit Masse verbunden werden, da dies JTAG vollständig deaktivieren würde und zwar nicht nur für ein einzelnes Gerät, sondern für die gesamte Scankette.

Wenn möglich, führen Sie die TAP-Signale stets von anderen aktiven Signalen weg, um das Rauschen zu reduzieren und die Signalintegrität zu verbessern. Die serielle JTAG-Schnittstelle läuft typischerweise mit einer Taktrate von 10 MHz bis 30 MHz und ein schlechtes Layout kann Fehler verursachen, die sehr schwer zu erkennen sind und ein Re-Spin erfordern können, um wieder ausgebessert zu werden.

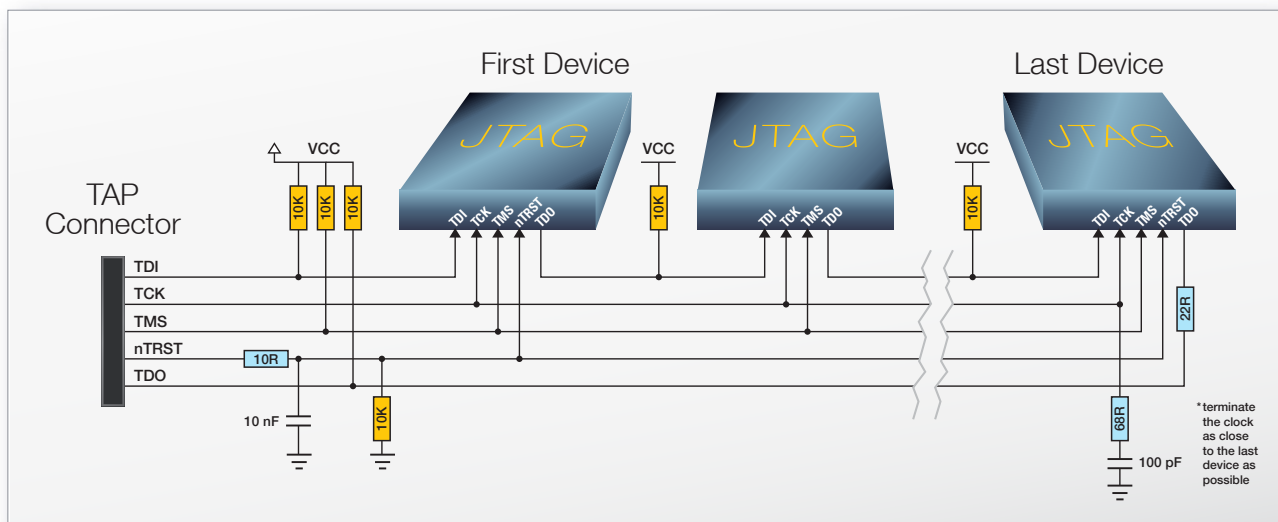
Verschachtelte TAP-Signale mit Strom oder Masse können helfen, Probleme mit nicht funktionierenden JTAG-Ketten zu diagnostizieren. Ein TAP-Signal, das zu einem konstanten Signal kurzgeschlossen ist, wird viel leichter diagnostiziert als zwei TAP-Pins, die zusammen kurzgeschlossen sind.



## ➤ Verwendung der korrekten Beendigung für alle TAP-Signale

Der TCK sollte mit einem 68  $\Omega$ -Widerstand und einem 100 pF-Kondensator in Serie auf Masse gelegt werden, so nah wie möglich an der Stelle, an der das Signal in das Endgerät der Kette dringt. TDI und TMS sollten mit 10 k $\Omega$ -Widerständen auf die Stromschiene gezogen werden. Das TDO einer jeden Komponente sollte mit einem 10 k $\Omega$ -Widerstand auf die Stromschiene gezogen werden und einen 22  $\Omega$ -Serienwiderstand in der Nähe des Endgeräts der Kette montiert haben, nah am Pin. Es wird auch empfohlen, der nTRST-Leitung einen 10 k $\Omega$ -Pull-Down-Widerstand hinzuzufügen, um offene Eingänge zu vermeiden.

Um zu verhindern, dass vom Kabel aufgenommenes Rauschen falsche Resets verursacht, sollte ein kleiner RC-Filter auf nTRST in der Nähe des Anschlusses platziert werden. Empfohlen werden Werte von 10  $\Omega$  und 10 nF.

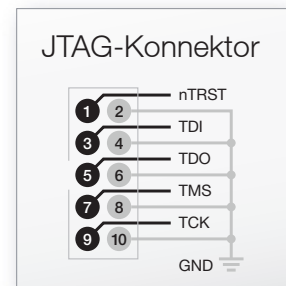


## >> Mehrere JTAG-Ketten

Es kann notwendig sein, einige Komponenten direkt mit ihren eigenen Anschlüssen während der Entwicklung oder Programmierung zu Debug-Zwecken zu verbinden. Die zusätzliche Option, diese Konnektoren zu einer einzelnen Kette für Produktionstests zu verknüpfen, kann durchaus vorteilhaft sein und sowohl die Stückliste (BOM) als auch die Handhabungskosten während der Produktion reduzieren. Der XJLink2-Controller von XJTAG kann bis zu vier JTAG-Anschlüsse auf einem Board anschließen.

## >> Konnektor-design

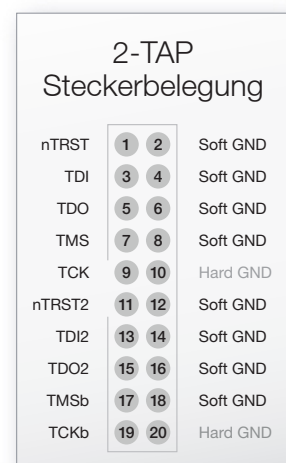
Bei der Angabe der Signalpositionen am JTAG-Stecker ist es wichtig, mögliche Übersprech-/Interferenzprobleme zu berücksichtigen. Eine Verschachtelung aktiver Signale mit Masseverbindungen minimiert diese Effekte. Wenn die Anzahl der auf dem Steckverbinder vorhandenen Pins dies unpraktikabel macht, dann priorisieren Sie die Platzierung von TCK neben Masse.



Es ist besonders wichtig, eine Angrenzung von TDO- und TCK-Signalen zu vermeiden. TDO-Übergänge treten an den fallenden Kanten von TCKs auf, daher kann das Übersprechen von TDOs Fehler auf TCKs verursachen und Daten verfälschen. Wenn es nicht möglich ist, Erdungen mit jedem Signal zu verschachteln, können Sie TDI und TMS zu beiden Seiten von TDO routen, da sie an der steigenden Flanke von TCK abgetastet werden, wodurch sich Störungen von TDO auflösen können, bevor TDI und TMS einen halben Clock-Zyklus später gelesen werden können.

Es ist sehr wichtig, eine starke Masseverbindung zwischen dem JTAG-Controller und dem zu prüfenden Board zu haben. Wenn es Ersatzpins vom JTAG-Stecker zum Board gibt, kann das Hinzufügen von zusätzlichen Massesignalen von Vorteil sein. Der XJLink2 hat feste Masseverbindungen an den Stiften 10 und 20, so dass diese die Signalintegrität verbessern.

Bei Platinen mit mehreren JTAG-Ketten ist es wichtig, dass jeder TAP mindestens einen Masseleiter hat, der mit seinen Signalen durch das Kabel läuft. Da ein XJLink2 nur zwei feste / harte Erdungen hat, sollte für Schaltungen mit mehr als zwei TAPs ein Splitter / Signal Integrity Board oder ein Äquivalent verwendet werden, damit die Erdungsverbindungen entsprechend verteilt werden können.



Im Idealfall wird jedes neue Design den gleichen Steckverbindertyp und das gleiche Signallayout verwenden, so dass die Kabel wiederverwendet werden können. Diese Anschlüsse sollten „kodiert“ werden, um eine Fehlausrichtung zu vermeiden.

Wenn der Stecker versehentlich in umgekehrter Reihenfolge eingesetzt werden könnte, wählen Sie ein Pin-Layout, das Kurzschlüsse zwischen den Anschlusspins für die Stromversorgung und GND- (oder „Soft GND“) Pins des TAP-Controllers verhindert.

Wenn Sie gezwungen sind, eine Steckerbelegung für Ihr Board zu verwenden, die TDO nicht ausreichend von TCK isoliert, sollten Sie die Verwendung eines Adapterkabels in Betracht ziehen, damit die Signale getrennt werden können, bevor Sie ein Kabel unter führen welches länger als ein paar Zentimeter ist.

## » JTAG-Modus wählen

Bei einigen Komponenten können TAP-Pins so konfiguriert werden, dass sie noch andere Funktionen als 1149.x-Boundary-Scan haben. Daher ist es wichtig sicher zu stellen, dass das Design nicht verhindert, dass Komponenten für das JTAG-Testen verwendet werden können. Die Funktion dieser Pins wird normalerweise durch das Abtasten anderer Pins auf der Komponente, wenn es zurückgesetzt wird, konfiguriert. Je nach Anforderung des Designs kann diese Konfiguration auf mehrere Weisen erreicht werden:

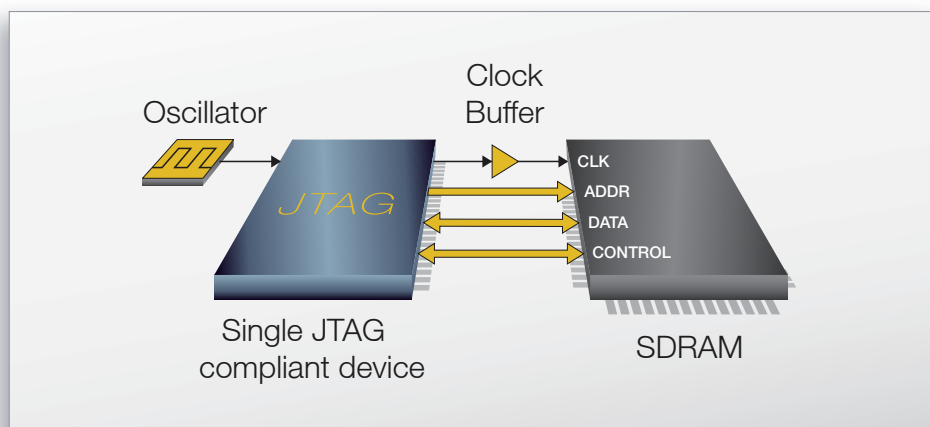
- Wenn Boundary-Scan die einzig benötigte Funktion auf den TAP-Pins ist, können die Konfigurations-Pins an die erforderlichen Level gebunden werden.
- Wenn Boundary-Scan die dominierende Funktion für die Pins ist, aber es möglich ist, dass andere Funktionen erforderlich sein können, dann können optionale Widerstände die Einstellung der Konfigurations-Pins auf high oder low mit nur einer kleinen Änderung des Boards ermöglichen.
- Wenn ein einfaches Umschalten zwischen den Funktionen erforderlich ist, können die Konfigurationssignale an einen Stecker geführt werden. Pull-Widerstände können verwendet werden, um den Standardzustand zu definieren, der dann von einem Controller wie dem XJLink2 übersteuert werden kann.

Die EMU0- und EMU1-Pins auf einer Vielzahl von Texas Instruments-Prozessoren sind ein übliches Beispiel für solche Konfigurationsstifte, genau wie die Mode Pins auf Freescale-Prozessoren und das JTAGSEL-Signal auf der AT91-Prozessorfamilie.

Die erforderlichen Werte für diese Pins werden im Datenblatt / Referenzhandbuch für die Komponente definiert. Sie können jedoch auch im Abschnitt COMPLIANCE\_PATTERNS der BSDL-Datei für die Komponente angezeigt werden.

## » Beachten Sie den Takt zu synchronen Komponenten

Viele synchrone Komponenten können durch Boundary-Scan getestet werden, dies ist jedoch abhängig von der Quelle der Taktung. SDRAM kann beispielsweise nur getestet werden, wenn die Taktung von einer 1149.x-kompatiblen Komponente gesteuert wird. Wenn eine selbsterregte Taktung direkt mit dem SDRAM verbunden wäre, dann wäre es nicht möglich, die Boundary-Scan-Testvektoren in Echtzeit auf diese zu synchronisieren.



Wenn einer konfigurierbare JTAG-Komponente, wie z.B. eine CPLD oder ein FPGA, verfügbar ist, führen Sie die Taktquelle über dieses Gerät zum SDRAM oder zum Taktgeber, um die Testabdeckung zu maximieren, wie unten gezeigt.

Es ist auch wichtig sicherzustellen, dass die Taktpuffer keine minimale Betriebsfrequenz haben. Einige Taktgeber haben einen speziellen Steuereingang, um niederfrequente Signale zu ermöglichen. Solche Signale müssen von einem JTAG-fähigen Pin zugänglich sein.

Andere Geräte, wie z.B. Ethernet-PHYs und Schalter, benötigen einen bestimmten Frequenz-Takteingang, um zu funktionieren. In diesen Fällen ist es wichtig sicherzustellen, dass der Takt nicht von einem 1149.x-fähigen Pin bereitgestellt wird. Ein Kristall oder Oszillator sollte mit der nicht-kompatiblen Komponente entweder direkt oder über einen Taktpuffer verbunden sein.

Während es nicht möglich ist Boundary-Scan zur Interaktion mit nicht-kompatiblen Komponenten mit diesen Geschwindigkeiten zu verwenden, erlaubt die Bereitstellung eines geeigneten Takts oft zumindest ein gewisses Maß an Tests, die sonst nicht möglich wären.

## ➤ Ziehen Sie einen Watchdog-Einsatz in Betracht

Geben Sie Links oder 1149.x-steuerbare Logik an, um sicherzustellen, dass der Betrieb der Watchdogs während des Tests deaktiviert werden kann. Das Zurücksetzen von Ereignissen, wenn ein Watchdog während des Testens keine regelmäßige Eingabe erhält, kann ein unerwartetes Testergebnis erzeugen oder sogar verhindern, dass Tests abgeschlossen werden.

## ➤ Testen mit nicht-konfigurierten FPGAs und CPLDs

Konfigurierbare Komponenten wie FPGAs und CPLDs haben einen hohen Anteil an I/O-Signalen, die ideal für eine hohe Testabdeckung sind. Sind diese Komponenten jedoch konfiguriert, bieten viele dieser Signale nicht ihre volle I/O-Fähigkeit, da die Eingabe- oder Ausgabefunktionalität, die in der Anwendung nicht benötigt wird, „ausgeschaltet“ ist. Durch die Verwendung von Boundary-Scan wird diese Funktionalität nicht reaktiviert, so dass die Testabdeckung reduziert wird.

Es gibt einen [Anwendungshinweis](#) auf der XJTAG-Website, der die beste Strategie für die Prüfung von Leiterplatten mit diesen Komponenten beschreibt.

## ➤ Sicherstellen eines geringen Schiefenfalls zwischen TAP-Signalen

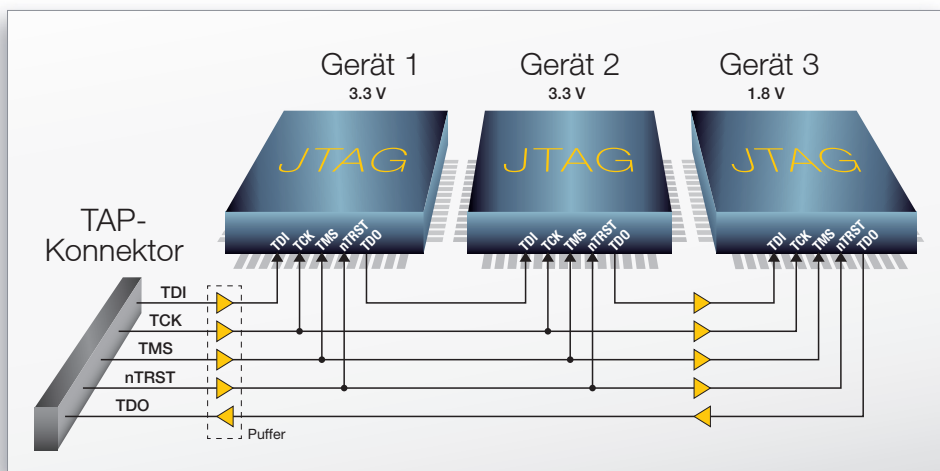
Für den korrekten Betrieb ist es wichtig, dass zwischen den TAP-Signalen eine geringer Versatz liegt. Die Spurlängen und Pufferverzögerungen der TCK- und TMS-Signale sollten vor allem gleichmäßig gehalten werden.

## ➤ Gewährung von Zugriff zur Unterstützung der Suche nach Scan-Kettenfehlern

Durch die Ermittlung des physikalischen Zugriffs zwischen dem TDO-Signal einer Komponente und dem TDI-Signal des nächsten wird eine einfache Erkennung der Komponente im Falle einer nicht funktionierenden Scankette ermöglicht. Dies kann von einem seriellen Abschlusswiderstand oder einem Testpunkt sein.

## ➤ Pufferung der TAP-Signale und Spannungspegel-Trennung

Wenn möglich, puffern Sie die primären TAP-Signale, um Rauschen und Impedanzfehlanspassungen zu minimieren und das Fan-Out zu verbessern. Im Allgemeinen ist es bei relativ kurzen Spurlängen akzeptabel vier bis sechs Komponenten fächerartig auszulegen. Bei längeren Spuren sollten dahingegen lediglich ein oder maximal zwei Komponenten fächerartig ausgelegt werden. Stellen Sie sicher, dass TCK, TDI, TMS und nTRST am Board-Eingang und das TDO am Boardausgang gepuffert werden. Es ist wichtig, dass die Puffer nicht invertierend sind.



Bei JTAG-Komponenten mit unterschiedlichen Spannungen empfiehlt es sich, diese in Komponentenketten zu unterteilen, die auf dem gleichen Spannungspegel arbeiten. Alternativ gruppieren Sie die Vorrichtungen aus der gleichen Spannungsgruppe zusammen in der Kaskade, z.B. 3,3 V, und verbinden diese dann mit den Komponenten der nächsten Spannung durch einen Spannungswandler, um das Signal entsprechend einzustellen, z.B. von 3,3 V auf 1,8 V.

Wenn es nicht möglich ist, TAP-Signale zwischen Spannungsdomänen zu puffern, dann entwerfen Sie die JTAG-Kette dergestalt, dass Spannungen heruntersgesetzt werden können, bspw. von 3,3 V-Signalen auf 2,5 V (z.B. mit Widerstandsteiler), da dies einfacher ist als Spannungen hochzusetzen.

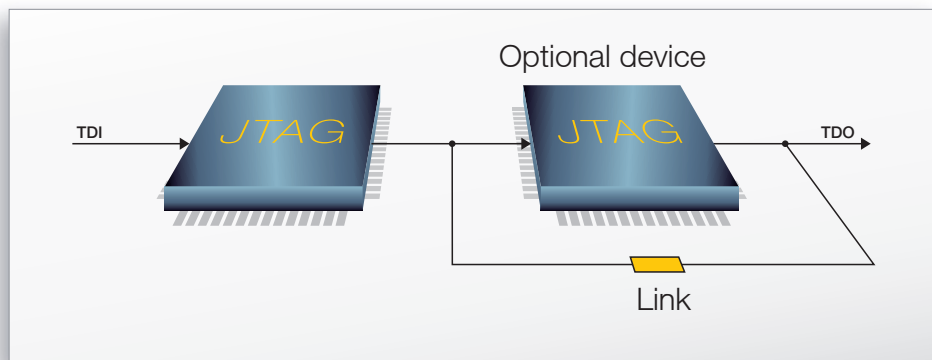
Es ist wichtig, dass alle gepufferten Signale wie zuvor beschrieben beendet werden. Der TCK-Eingang in einen Puffer benötigt eine Terminierung in der Nähe des Puffers, während die Ausgänge von beliebigen Puffern eine parallele Beendigung in der Nähe der letzten Komponente auf der Spur benötigen. Alle gepufferten TDO-Signale sollten einen Serienabschluss in der Nähe des Treibers haben.



## ➤ Nebenschluss optionaler 1149.x-konformer Komponenten

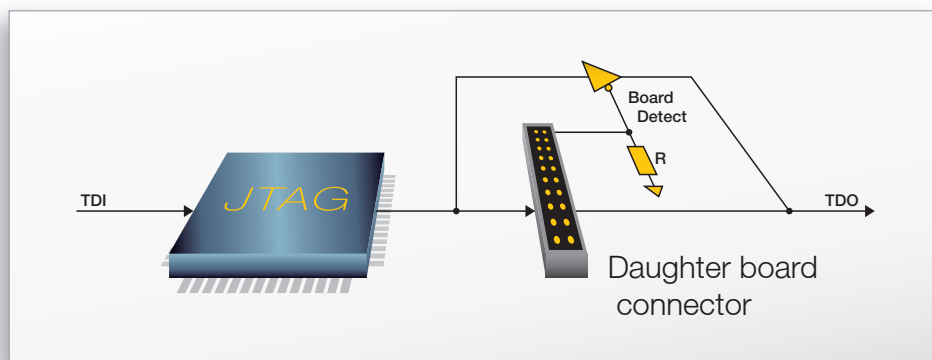
Wenn JTAG-Komponenten auf einer Platine optional sind, ist darauf zu achten, dass eine Verbindung, normalerweise ein 0 Ω-Widerstand, zur Verfügung steht, um die TAP-Datenkette zu vervollständigen, wenn die Komponente nicht verbaut ist.

Wenn dies nicht der Fall ist wird der unterbrochene Datenpfad in der Scan-Kette verhindern, dass alle Tests mit anderen Komponenten in dieser Kette durchgeführt werden können. Wenn eine Komponente wahrscheinlich nicht eingepasst wird, sollte sie besser in einer separaten Scan-Kette platziert werden.



## ➤ Fügen Sie der 1149.x-Kette Add-ons und optionale Boards hinzu

Wo Tochterleiterplatten benutzt werden, leiten Sie die 1149.x-Kette über die Leiterplattensteckerleiste durch die Komponenten auf den Tochterboards. Wenn die Tochterplatine eine Option ist, dann verwenden Sie eine Erkennungslogik für das Board, um den TAP-Datenpfad auf der Tochterplatine zu umgehen, wenn sie nicht wie unten gezeigt installiert ist.

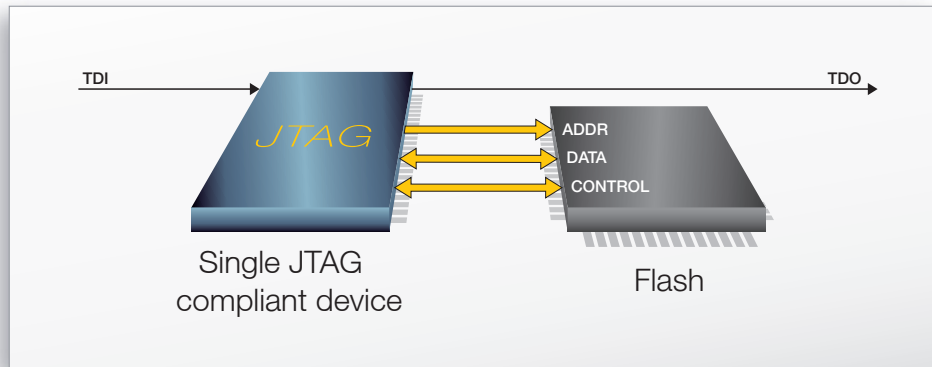


## ➤ Einbezug von 1149.x-Kompatibilität bei Entwurf oder Spezifizierung von ASICs

Wenn Ihr Design ASICs oder anderes benutzerdefiniertes Silicium enthält, sollten Sie in Erwägung ziehen, die benutzerdefinierten Geräte 1149.x-konform zu machen, da sie sonst wahrscheinlich untestbar sind.

## ➤ Machen Sie programmierbare Abschnitte von der 1149.x-Kette aus zugänglich

Wenn In-System-Programmierung von Komponenten wie Flash oder EEPROM erforderlich ist, stellen Sie sicher, dass alle notwendigen Pins der programmierbaren Geräte von den JTAG-Komponenten aus zugänglich sind.



Die Boundary-Scan basierte In-System-Programmierzeit ist sowohl von der Länge der Scan-Kette als auch von der TCK-Frequenz abhängig. Für die kürzeste Boundary-Scan-basierte Programmierzeit muss die TCK-Frequenz so hoch und die Scan-Kette so kurz wie möglich sein.

Es ist ratsam, alle benötigten Knoten von einer einzigen JTAG-Komponente aus zugänglich zu machen. XJTAG optimiert automatisch den Programmiervorgang, indem alle JTAG-Komponenten, die für einen bestimmten Scan nicht benötigt werden, in einen Modus namens CLAMP gesetzt werden, der den Datenpfad durch jedes Gerät zu einem einzigen Bit verkürzt.

Es empfiehlt sich, Adressspeicher-Konfigurationen zu vermeiden, da sie die Anzahl der erforderlichen Scans zum Schreiben jedes Bytes/Worts erhöhen, was wiederum die Programmierzeit signifikant erhöht.

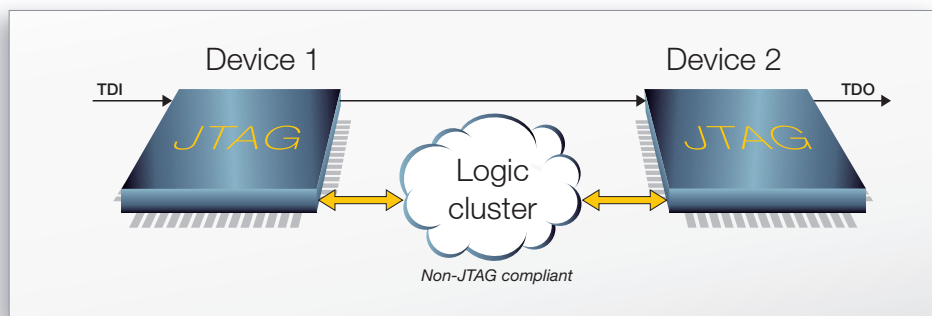
## ➤ Stellen Sie direkten Zugriff auf Programmiersignale zur Verfügung

Die Programmierzeiten können durch direkte Steuerung von Programmiersignalen wie der Schreibfreigabe oder sogar ganzen Protokollen wie SPI und IIC, weiter optimiert werden. Durch das Routing der relevanten Netze zu einem Konnektor auf der Platine können sie mit Ersatz-I/O-Pins anstatt durch Boundary-Scan auf dem JTAG-Controller gesteuert werden. Dies hilft Programmierzeiten zu reduzieren, da Signale, die häufig geändert werden müssen und normalerweise zwei vollständige Scans erfordern, unabhängig von Kettenlänge und Frequenz geschaltet werden können.

Um diese Funktionalität nutzen zu können, muss es möglich sein, alle Boundary-Scan-Treiber auf diesen Netzen zu deaktivieren.

## ➤ Umschließen Sie Logikgruppen mit JTAG-Komponenten

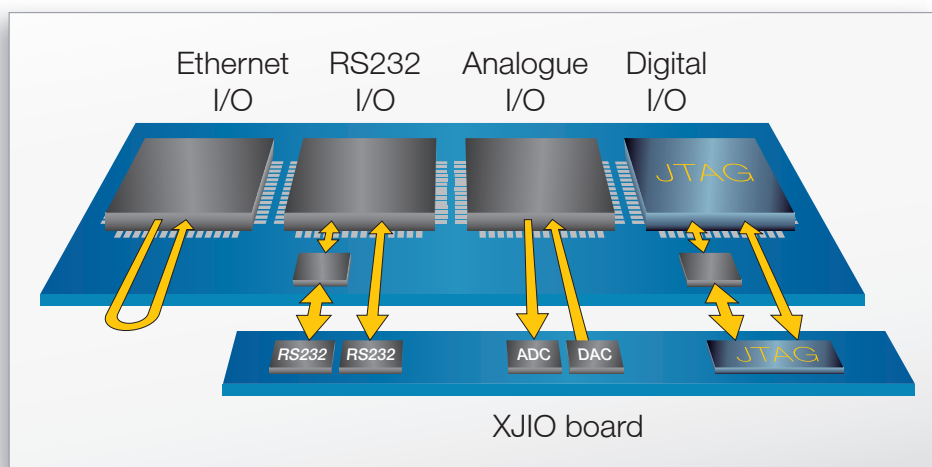
Ordnen Sie die nicht-JTAG-Logik in Gruppen an, die von JTAG-Komponenten umgeben sind, die alle Eingänge steuern und alle Ausgänge zu und von der Gruppe überwachen können. Versuchen Sie sicherzustellen, dass genügend Zugriff auf Knoten innerhalb der Gruppe verfügbar ist, um den Betrieb der Gruppe so vollständig wie möglich zu testen. Fügen Sie ggf. zusätzliche JTAG-Verbindungen hinzu, um Zugriff auf die erforderlichen Clusterknoten zu erhalten. Alternativ kann es vorteilhaft sein, das Logiksystem in einer programmierbaren Logikkomponente zu haben, die Boundary-Scan unterstützt.



## ➤ Erweitern Sie 1149.x-Testen durch Steckverbinder

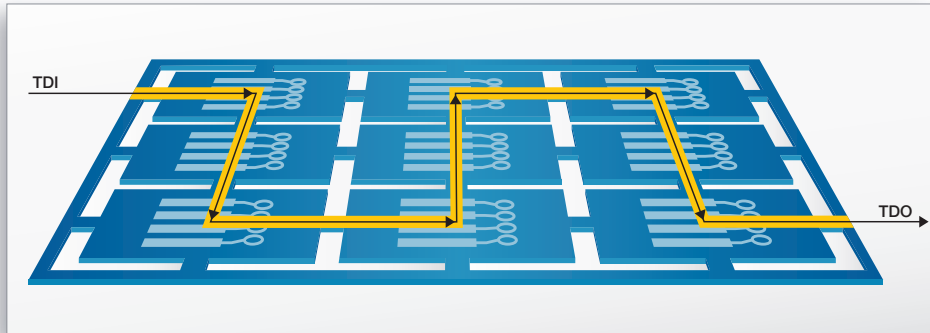
Die Testabdeckung ist stark erhöht, wenn externe Konnektoren an zusätzliche 1149.x-zugängliche Netze angeschlossen werden können. Die Prüfung von Anschlusspins, die direkt mit Netzen mit einem einzigen 1149.x-Pin verbunden sind, kann auf offene Schaltkreise erweitert werden, wenn sie an externe, 1149.x-steuerbare I/O, wie die XJIO-Platine, angeschlossen sind.

Diese Form der Prüfung ist nur für Standard-Digitalsignale möglich. Wo solche Tests nicht möglich sind, z.B. bei Signalen wie Ethernet und RS485, designen Sie I/O-Ports, so dass Rückschleifen verwendet werden können.



## ➤ Ziehen Sie das Testen von Mehrfachleiterplatten als Einheit in Betracht

Wenn mehrere Boards auf einem einzigen Paneel konstruiert werden, sollten Sie die Übertragung von Leistung und TAP-Signalen über einige oder alle Platinen auf einem Paneel in Betracht ziehen, um das gleichzeitige Testen aller Platinen vor dem Trennen zu ermöglichen.



## ➤ Testen Sie analoge Schaltungen

Obwohl 1149.1 und 1149.6 keine explizite Funktionalität für das Testen von analogen Schaltungen enthalten, können viele Tests dank sorgfältigen Designs durchgeführt werden. Ziehen Sie die Verwendung von Verbindungen mit 1149.x-kompatiblen Ausgängen in Betracht, um Komparator- oder Verstärkereingänge auf feststehende Zustände zu treiben und verwenden Sie die Ausgänge dieser Geräte, um bekannte Eingänge für JTAG-Komponenten bereitzustellen. Der Zusatz von kostengünstigen DAUs und ADUs kann zusätzliche Steuerungen für und die Überwachung von analogen Gruppen durch die Verbindung der digitalen Seite des ADU und des DAU mit Komponenten in der Boundary-Scan-Kette bereitstellen. Mit wenigen zusätzlichen, geeigneten Schaltungen können große Mengen an analogen Tests durchgeführt werden.

## ➤ Verlassen Sie sich nicht auf programmierbare Pull-Widerstände

Versuchen Sie nicht die programmierbaren Pull-ups zu verwenden, die an den I/O-Kontaktstellen einiger Prozessoren und FPGAs bereitgestellt werden, um die externe Logik zu bedienen. Wenn die auf einem IIC-Bus benötigten Pull-ups nur durch programmierbare Pull-ups bereitgestellt werden, dann können alle an den IIC-Bus angeschlossenen Komponenten nicht durch Boundary-Scan getestet werden. Wenn dies nicht vermieden werden kann, dann stellen Sie die IIC-Bus-Pins an einem externen Stecker zur Verfügung, so dass Pull-ups extern für Testzwecke hinzugefügt werden können.

## ➤ Ziehen Sie die Nutzung der Funktionalität von JTAG-Komponenten in Betracht

Es gibt immer einige Teile jeder Schaltung, die nicht mit Standard 1149.x-Techniken getestet werden können, während andere Teile des Designs „at speed“-Tests erfordern können. In diesen Situationen können eventuell Programme verwendet werden, die in Komponenten der JTAG-Kette geladen werden, um die Funktionalität der Leiterplatte auszuführen.

Konfigurierbare Komponenten wie CPLDs und FPGAs können mit SVF- oder STAPL-Dateien programmiert werden. Für Prozessoren besteht eventuell die Möglichkeit, die Flexibilität der Programmiersprache XJEase von XJTAG zu nutzen, um die Konfigurationen zu laden.

## » Verwenden Sie die Testfunktionalität in Nicht-JTAG-Komponenten

Einige Nicht-JTAG-Komponenten verfügen über spezielle Testfunktionen, mit denen die Testabdeckung erhöht werden kann. Diese Merkmale, wie z.B. NAND-Baumgraphen oder Testmuster-Generierung, ermöglichen das Testen von Netzen, die mit Signalen verbunden sind, die nicht mit Boundary-Scan getestet werden können, um mit der Komponentenfunktionalität zu interagieren. Sie können durch Schreiben an Registern mit Schnittstellen wie SPI, IIC und MDIO aufgerufen werden, die durch Boundary-Scan gesteuert werden können.

## » Interaktive tests

Eine Reihe von Tests kann nur mit Hilfe des Testbedieners durchgeführt werden. Visuelle Komponenten können getestet werden, indem der Bediener ihre Funktion beobachtet und ihre Ergebnisse eingibt, Schalter, indem ein JTAG-Gerät den Zustand des Schalters überwacht, während er vom Bediener betätigt wird. Andere Geräte wie Schrittmotorsteuerungen und Lautsprecher Systeme können in ähnlicher Weise getestet werden.

## » Programmieren Sie nicht-flüchtige Speicherkomponenten

XJTAG kann verwendet werden, um viele Komponenten wie NOR- und NAND-Flashes mit seriellen und parallelen Schnittstellen zu programmieren. Dies ist nützlich beim Laden von Anwendungsbildern oder zum Speichern von Konfigurationsoptionen, Seriennummern, Ethernet-MAC-Adressen usw. als Teil des Testverfahrens.

## » Schnelle Flash-Programmierung

Schaltungs-Programmierung von Flash-Komponenten über Boundary-Scan kann sehr langsam sein, da jedes Netz nur einmal mit jedem JTAG-Scan umgeschaltet werden kann und jeder Scan mehrere tausend Bits lang sein kann. Eine Lösung für dieses Problem besteht darin, entweder FPGA-unterstützte oder prozessorgestützte Programmerroutinen zu verwenden. Diese Routinen laden zuerst das FPGA oder den Prozessor mit einer Programmiermaschine, die die Flash-Programmierung steuert. Der JTAG-Port wird dann verwendet, um Daten an die Programmiermaschine zu übertragen.

Die FPGA-unterstützte Programmierung kann mit XJTAGs XJFlash-Technologien umgesetzt werden. Dadurch können Flash-Komponenten mit Geschwindigkeiten in der Nähe ihres theoretischen Maximums programmiert werden. Wenn Ihr Design eine Flash-Komponente enthält, die nicht mit einem FPGA verbunden ist, aber das Design ein FPGA mit Ersatzpins beinhaltet, können Sie durch Verbindung der Flash-Komponente mit diesen zusätzlichen Pins die nötige Flexibilität erreichen, um XJFlash zu verwenden.

Prozessor unterstützte Programmierung, langsamer als XJFlash, ist immer noch deutlich schneller als mit Boundary-Scan. Die Debug-Schnittstelle, die auf vielen Prozessoren zur Verfügung steht, verwendet JTAG als Kommunikationsprotokoll. Solche Prozessoren können über JTAG konfiguriert werden, um ihre Flash-Controller zur Programmierung eines Bildes in Flash zu verwenden.

Für weitere Informationen über die Implementierung der schnellen Flash-Programmierung wenden Sie sich bitte an [support@xjtag.com](mailto:support@xjtag.com).

## Glossar

---

<b>ADU</b>	Analog-Digital-Umsetzer (engl. ADC – Analog to Digital Converter)
<b>ASIC</b>	Anwendungsspezifische integrierte Schaltung (engl. Application Specific Integrated Circuit)
<b>BSDL</b>	Boundary Scan Description Language
<b>CPLD</b>	Complex Programmable Logic Device
<b>DAU</b>	Digital-Analog-Umsetzer (engl. DAC – Digital to Analog Converter)
<b>DFT</b>	Design for Testability
<b>EEPROM</b>	Elektrisch löschbarer programmierbarer Nur-Lese-Speicher, auch E <sup>2</sup> PROM (engl. Electrically Erasable and Programmable Read Only Memory)
<b>FPGA</b>	Field Programmable Gate Array
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>1149.x</b>	IEEE Standard 1149.x: 1149.1: Test Access Port and boundary scan architecture 1149.6: AC coupled and differential signals (see <a href="http://www.ieee.org">www.ieee.org</a> )
<b>IIC</b>	Inter-Integrated Circuit, auch I <sup>2</sup> C
<b>I/O</b>	Input/Output
<b>JTAG</b>	Joint Test Action Group
<b>nTRST</b>	Test Reset (Active low TAP Signal)
<b>STAPL</b>	Standard Test And Programming Language
<b>SVF</b>	Serial Vector Format
<b>TAP</b>	Test Access Port
<b>TCK</b>	Test-Takt, TAP-Signal (engl. Test Clock)
<b>TDI</b>	Testdaten-Eingang, TAP-Signal (engl. Test Data Input)
<b>TDO</b>	Testdaten-Ausgang, TAP-Signal (engl. Test Data Output)
<b>TMS</b>	Test Mode Select-Anschluss, TAP-Signal
<b>XJTAG</b>	Eine Reihe von Werkzeugen, die die Entwicklung und den Test von elektronischen Systemen unterstützen

# Spezialangebot

## Kostenloses Board-Setup + kostenloses Testen von XJTAG



- Designen Sie Leiterplatten mit BGAs?
- Umfasst Ihre Hardware FPGAs, CPLDs, DSPs oder Mikroprozessoren?
- Möchten Sie Ihre Leiterplatten debuggen, Fehler identifizieren und Ihr Design schnell und einfach bestätigt wissen?

Dabei kann Ihnen **XJTAG-Boundary-Scan behilflich sein:** wir bieten Ihnen eine **kostenlose 30-Tage-Testversion von XJTAG auf Ihrer eigenen Leiterplatte.**

Sie haben richtig verstanden: XJTAG wird Ihnen im Rahmen der 30-tägigen Testversion auf Ihrer eigenen Leiterplatte kostenlos ein Testsystem einrichten.

**Melden Sie sich noch heute an und finden Sie heraus, wie XJTAG Ihnen helfen kann, Zeit und Geld zu sparen**

**Erfahren Sie, warum führende Unternehmen XJTAG nutzen**

“XJTAG ist ein absolutes Muß für jedes Unternehmen, das komplexe Schaltungen entwirft, die hochpolige BGAs oder Chip-Scale-Komponenten beinhalten.”

“XJTAG ist einfach zu bedienen und unglaublich schnell, was es uns ermöglicht hat, viele Wochen vom Entwicklungsplan für unser RFeye-Modul einzusparen und damit unser Entwicklungsteam von zeitaufwändigen Debugging-Aufgaben zu befreien.”

Alistair Massarella, CEO – CRFS

### Fallstudie ARM



**ARM wählt XJTAG zur Fehlersuche und zum Test der „RealView“ Entwicklungswerkzeuge**

ARM® (weltweit führender IP-Lieferant für „semiconductor intellectual property“), und reduzierte die Entwicklungszeit und -kosten der RealView® Entwicklungswerkzeuge, mit Hilfe des XJTAG-Boundary-Scan-Entwicklungssystems. Die Prozesse der Fehlersuche und des Testens von hochintegrierten, mehrlagigen Entwicklungsbaugruppen wurden mit dem XJTAG-System verbessert und beschleunigt.

**Melden Sie sich jetzt an!**

[www.xjtag.de/testversion](http://www.xjtag.de/testversion)