



BOARD DFT GUIDELINES

For PCB Testing and In-System-Programming

The logo for JTAG Technologies, featuring the letters 'JTAG' in a large, stylized, white serif font. Below 'JTAG' is the word 'TECHNOLOGIES' in a smaller, white, sans-serif font. The background is a dark blue gradient with abstract, glowing blue and white geometric shapes and bokeh light effects.

JTAG
TECHNOLOGIES®

We *are* boundary-scan.®

This booklet has been prepared with great care, but yet might contain inconsistencies. The reader is welcome to give any comment or suggestions. The illustrations and descriptions in the “Board DFT Guidelines” are for illustrative prupose only. The contents are subject to alterations without notice. No part of this document may be reproduced in any form without the prior written permission of JTAG Technologies B.V.

March 2019

© JTAG Technologies B.V.

PREFACE

Boundary-Scan Technology For PCB Testing and In-System Configuration

Why boundary-scan?

Throughout the electronics industry, manufacturers are turning to the latest device packaging technologies, such as ball grid arrays (BGAs), chip-scale packages, and other small outlines, to provide the functionality and miniaturization their customers demand. However, the new packages are increasing the difficulty of accessing printed circuit boards for In-Circuit Testing (ICT) and in-system device configuration. These difficult access problems have been addressed by the industry through adoption of the IEEE 1149.1 boundary-scan standard (aka JTAG), allowing pin-level access - independent of the device packaging technology - to even the most crowded printed circuit board assemblies.

Nowadays almost all popular complex ICs support IEEE 1149.1 test features. At the printed circuit board level it is the responsibility of the hardware designers and project managers to use the available device IEEE 1149.1 features for achieving a better board level testability. The guidelines described in this booklet help the designer to implement board level Design For Test (DFT).

Benefits of boundary-scan

The result of introducing boundary-scan can lead to a reduction in the number of test points needed on the board resulting in simpler board layouts, less costly test fixtures, reduced time (or even no time) on an in-circuit tester, and faster time-to-market. On the programming side, JTAG/boundary-scan provides a means for configuration of almost all types of cPLDs, serial PROMs, microcontrollers and flash memories, regardless of size or package type, on the board after PCB assembly.

By adopting JTAG/boundary-scan on your compliant designs you'll quickly realize substantial savings by reducing device handling, decreasing the burden of inventories of pre-programmed parts, and integrating the programming steps into the board production line.



GLOSSARY OF ABBREVIATIONS

ASIC	-	Application Specific Integrated Circuit
AW	-	AutoWrite
BDM	-	Background Debug Mode
BGA	-	Ball-Grid Array
BSDL	-	Boundary-Scan Description Language
BSR	-	Boundary-Scan Register
cPLD	-	Complex Programmable Logic Device
DDR	-	Double Data Rate (RAM)
DFT	-	Design for Test(ability)
DIOS	-	Digital Input Output Scan
DNP	-	Do Not Place
DSP	-	Digital Signal Processor
ECL	-	Emitter Coupled Logic
FPGA	-	Field Programmable Gate Array
Hi-Z	-	High Impedance
I/O	-	Input - Output
IC	-	Integrated Circuit
ICT	-	In-Circuit Test(ing)
IEEE	-	Institute of Electrical and Electronic Engineers
IP	-	Intellectual Property
IR	-	Instruction Register
ISP	-	In-System Programming
JEDEC	-	Joint Electron Device Engineering Council
JTAG	-	Joint Test Action Group
LSP	-	Local Scan Port
MCGR	-	Multi-Cast Group Register
OE	-	Output Enable
PCB	-	Printed Circuit Board
PLL	-	Phase Locked Loop
SDRAM	-	Synchronous Dynamic Random Access Memory
SRAM	-	Static Random Access Memory
SSRAM	-	Synchronous Static Random Access Memory
TAP	-	Test Access Port
TCK	-	Test Clock
TDI	-	Test Data Input
TDO	-	Test Data Output
TMS	-	Test Mode Select
TRST	-	Test Reset
TTL	-	Transistor - Transistor Logic
UUT	-	Unit Under Test
VLSI	-	Very Large Scale Integration
WE	-	Write Enable

TABLE OF CONTENTS

	Preface	4
	Glossary of abbreviations	5
	Table of Contents - Table of Figures	6
1)	Introduction	8
2)	Considerations for Implementing Board Level Boundary-Scan	9
2.1	Component Selection	9
2.1.1	Select IEEE Std. 1149.1 Compliant Devices	9
2.1.2	Dual Function JTAG Port Pins	9
2.1.3	Check BSDL Files for Non-Compliance of 1149.1 Standard	9
2.1.4	Supported Instructions	10
2.1.5	IEEE Std. 1532 Support	11
2.2	Scan Chain Layout - Design Partitioning	11
2.2.1	Connection of JTAG Control Signals	11
2.2.2	Partitioning - Third Party Debugging/Emulation Tools	12
2.2.3	Partitioning - Different Logic and Voltage Families	12
2.2.4	Partitioning - Optimized Test Vector Execution	12
2.2.5	Partitioning - Board-to-board Interconnect Testing	12
2.2.6	Signal Termination	12
2.2.7	Edge Connector Access	13
2.2.8	Power Distribution	13
2.2.9	IEEE Std.1149.6	13
2.3	Scan Chain Layout - Signal Integrity	13
2.3.1	Tracking of TCK and TMS	13
2.3.2	Buffering and Fan-out of JTAG Signals	13
2.4	Scan Chain Layout - Physical Bypassing	15
2.4.1	Physical Bypassing	15
2.4.2	Single Device Bypass	15
2.4.3	Multiple Device Bypass	16
2.5	Control of Non- Boundary Scan Devices	16
2.5.1	Access to Control Signals of Non-Scan Logic	16
2.5.2	Control of Clock Signals	17
2.6	Special Considerations for SRAM based FPGA Devices	18
2.6.1	Boundary Scan Execution - Pre and Post Configuration	18
2.6.2	Xilinx and Altera examples	18
2.7	FPGA Re-configurable I/O Considerations	19
2.7.1	Pre and Post Configuration BSDL Files	19
2.8	Flash Programming Optimization	20
2.8.1	AutoWrite™ Accessq	20
2.8.2	Eliminating Signal Contention	20
2.8.3	3-State and 2-State Output Cells	21
2.8.4	Multiple Output Cell Control	21

2.8.5	AutoWrite™ Gating Arrangement	22
2.8.6	Flash Programming Optimization - Reducing Chain Length	22
2.8.7	Flash Programming Optimization - Split Boundary-scan Data Register	22
2.8.8	Flash Programming Optimization - Embedded Flash Controller	22
2.9	Device Specific DFT Requirements	24
2.9.1	Dual Port Operation - Default Background Debug Mode (BDM)	24
2.10	Design-for-Test - Other Considerations	25
2.10.1	Control of Watchdog Circuits	25
2.10.2	Configurable Termination Resistors	25
2.11	Embedded JTAG Options	25
2.11.1	Embedded instruments for enhanced testing	25
2.11.2	Adding an embedded USB-to-JTAG driver device	27
2.11.3	Adding an embedded JTAG controller device	28
3)	Configuring a design for MIOS tester	29
4)	Board layout considerations	30
4.1	Track separation	30
4.2	Placing of clock terminator	30
4.3	Effect of BGAs	31
5)	Contact Information	32
6)	Boundary-Scan - Basic Principles	34

Table of Figures

Figure 1	JTAG Signal Terminations
Figure 2	Fan-out of TCK and TMS
Figure 3	Physical Bypass of Single and Multiple Devices
Figure 4	Control of Non-Boundary Scan Devices
Figure 5a	Discouraged clock distribution circuit
Figure 5b	Recommended clock distribution circuit
Figure 6	FPGA Devices - configuration hold-off
Figure 7a	Pre-configured IO Considerations
Figure 7b	Post-configured IO Considerations
Figure 8	AutoWrite Access
Figure 9	Eliminating Signal Contention
Figure 10	Multiple Output Cell Control
Figure 11	AutoWrite Gating Arrangement
Figure 12	Split Boundary-scan Register
Figure 13	Embedded Flash Controller
Figure 14	Embedded Flash Controller for high capacity (NAND) flash
Figure 15	Physical Forcing JTAG Operation on Power-up
Figure 16	FPGA configured with JTAG translator "bridge"
Figure 17	Embedded USB=> JTAG Controller/Driver
Figure 18a	Best termination point for TCK -standard layout
Figure 18b	Best termination point for TCK -star layout
Figure 18c	Best termination point for TCK -star layout with multiple R-C networks

1 INTRODUCTION

So, at last you've managed to grab a five minute break to take a closer look at what you can do with that puzzling interface people have been talking about - the JTAG port. If you've already studied it in some detail you will have observed that by using the completely autonomous JTAG port you can access most of the pins of a JTAG compliant part while the core logical function of that part is effectively disconnected or suspended. This means that via a minimal 4 wire bus you can use test patterns (vectors) to toggle signals at output or I/O pins and monitor inputs at sense or I/O pins to establish whether or not interconnects between devices (nets) are as per your design.

If you are a design engineer reading this then you have already acknowledged the role you must play in DFT (Design for Test). If however you are a test or production engineer it is important that you establish contact with designers at the earliest planning stages of a new design and gain their co-operation to implement some of the ideas mentioned in this text. In either instance it is clearly advantageous to have regular meetings on a product's test strategy (for prototype debug, production and field service) as soon as product concepts are known.

The scope of this document is to present a number of DFT guidelines that can be used for reference to support the implementation of a boundary-scan architecture within electronic assemblies. Once established the boundary-scan functions of the design can be utilised by design and test engineers alike to perform board-level structural testing and the in-system configuration of cPLD, FPGA's and flash memory devices etc..



CONSIDERATIONS FOR IMPLEMENTING BOARD LEVEL BOUNDARY-SCAN

2.1 Component Selection

2.1.1 Select IEEE Std. 1149.1 Compliant Devices

The first thing to be considered when implementing a boundary-scan test architecture within any PCB design, is careful component selection. Ensure that [IEEE] 1149.1 compliant components are selected where possible. It is normally the case that many modern day VLSI (very large scale integration) devices are already 1149.1 compliant, however, in the case that of small scale integrated devices there may be a choice between selecting compliant or non-compliant parts from differing silicon providers that perform the same core function. If possible select the 1149.1 compliant device as this will increase your boundary-scan test coverage.

2.1.2 Dual Function JTAG Port Pins

There are still some devices that assign dual functionality to the JTAG port pins. This is most often the case where the silicon provider has decided that it is not cost effective to add a minimum of four additional boundary-scan access pins, purely for test purposes. In this instance they typically multiplex the JTAG control signals with predefined core function signal pins. In these circumstances the device dual function pins will usually default to core function mode on power-up, and will have to be switched into the JTAG mode of operation by selecting the appropriate drive state on a pre-defined 'JTAG Enable' pin. Once boundary-scan tests have been completed the enable pin can revert to its non-active state.

2.1.3 Check BSDL Files for Non-Compliance to IEEE 1149.1 Std.

Ensure that the data sheets and BSDL files are thoroughly checked for information specifying non-compliance to the IEEE 1149.1 standard. This is often specified in the BSDL file as a design attribute or a compliance patterns attribute as detailed below.

Example of design warning ...

```
-- 1149.1 Design Warnings
attribute DESIGN_WARNING of shark: entity is
"PWRDWN pin should be kept low to allow proper operation" &
"of TAP circuitry. There is a compliance enable on this" &
"pin to force the safe value. The boundary scan cell" &
"associated with the PWRDWN pin has been changed to an" &
"internal pin. It is cell number 18 in the boundary scan" &
"register description and has a safe value of 0 specified" &
" " &
"Please note that the SD_TP pin is a two state output during" &
"boundary scan. Care should be taken to place the appropriate" &
"value (0 for copper, 1 for fiber) in the boundary scan cell" &
"so as not to conflict with the functional connection of the pin";
```

2.1.4 Supported Instructions

All IEEE 1149.1 compliant devices must support the mandatory instructions SAMPLE/PRELOAD, EXTEST and BYPASS . However, it is highly desirable if devices also support the optional instructions of HIGHZ, CLAMP and IDCODE. The IDCODE instruction is particularly useful for determining if the correct device has been placed during the manufacturing process by reading the 32-bit device IDENT register which will specify the manufacturer's details, the device type/variant code and a 4-bit revision code. The revision code is helpful in situations where there are multiple versions of devices that have the same package footprint, but the internal functionality differs. In this situation it is more cost effective to detect the revision status at the process verification test stage, instead of the functional or system test stages.

The HIGHZ instruction is useful for optimizing the overall length of board-level boundary-scan chains, particularly when programming flash memory on a target board. Keeping the overall boundary-scan data register length to the minimum number of cells required for accessing the target flash memory data, address and control lines, will speed-up programming throughput. If multiple boundary-scan devices can access address and data busses connected to the target flash, those not used will have to be held in a high impedance state during memory read/write cycles in order to avoid bus contention. The HIGHZ instruction is a quick and effective way to set all output pins of a device in high-impedance state and at the same time bypass the often lengthy boundary-scan register.

The CLAMP instruction, similar to HIGHZ, also allows bypassing of the boundaryscan register. Additionally CLAMP allows preloading the boundary-scan register with static logic 1/0 levels to be driven, or disable values to keep output pins in the highimpedance state while the device remains in the CLAMP function. This is useful for controlling non-boundary-scan devices managed by the device that supports CLAMP while still bypassing the boundary-scan register during long test or programming operations.

2.1.5 IEEE Std. 1532 Support

IEEE Std 1532 is a unified programming concept for CPLDs and FPGAs that is also based on the IEEE 1149.1 infrastructure. The use of this method relies on enhanced BSDL models that include not only information for testing but also for configuring the internal device logic. Such BSDL models must be compliant to the relevant chapters of IEEE 1532 standard. Devices supporting this standard can also be programmed using a standardized data file format (ISC). By merging ISC files it is possible to configure multiple cPLD devices from different silicon vendors simultaneously.

2.2 Scan Chain Layout - Design Partitioning

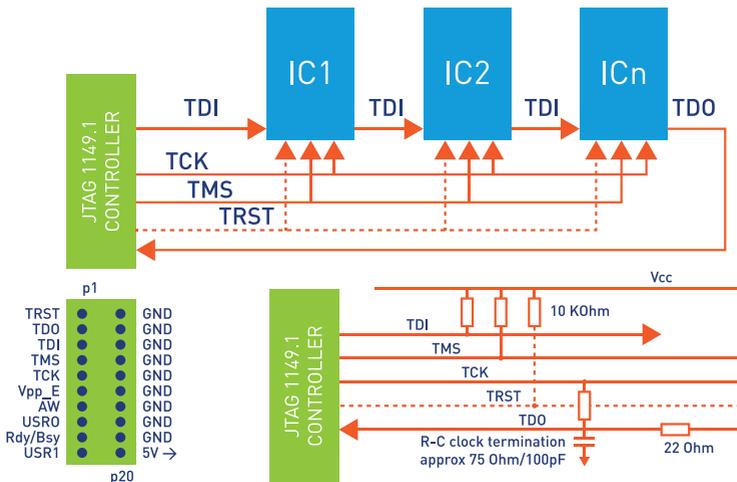


Figure 1 - JTAG Signal Terminations

2.2.1 Connection of JTAG Control Signals

Ensure that all the JTAG test access port (TAP) control signals; TCK, TMS and optional TRST are connected in parallel. TDI and TDO are used to “daisy chain” boundary-scan devices into a chain path as detailed in Figure. 1.



Note:- In some circumstances it may be necessary to construct more than one chain. For example:

2.2.2 Partitioning - Third Party Debugging/Emulation Tools

Some devices such as microprocessors and DSPs may need to be kept in a separate chain in order to maintain compliance with debugger or emulation tools. An example of this might be an IC vendor's DSP debugging software tool that expects every device in the chain to have an instruction register length equal to a specific number of bits. This of course cannot be guaranteed if other manufacturer's parts are used in the same chain(s), subsequently in this instance all the DSP devices would be laid out in a separate chain to allow the emulator/debugger to work effectively.

2.2.3 Partitioning - Different Logic and Voltage Families

It may be advantageous to segment different logic families e.g. ECL / TTL or voltage families e.g. 1.8V, 2.5V and 3.3V by placing the different families of devices into separate chains. Connect the external TAP connections to the individual chains and ensure that the external controller can be programmed for the appropriate drive levels.

2.2.4 Partitioning - Optimized Test Vector Execution

It may also be advantageous to segment devices into separate chains for improved test partitioning and diagnostic resolution, or for optimising the test vector execution. This is particularly important for flash memory programming, where care must be taken to optimise the overall boundary-scan register length so that the minimal number of boundary-scan register (BSR) bits are shifted for each flash memory read/write operation. It may be advisable to segment a particular functional block of boundary-scan devices into a separate chain, so that only the device accessing the address, data and control signals of the target flash memory is operating in EXTEST mode, with the other devices in the chain configured for BYPASS or HIGHZ.

2.2.5 Partitioning - Board-to-Board Interconnect Testing

It is also advantageous in system level applications to partition devices that provide access to the backplane interface to be segmented into a separate boundary-scan chain. This will optimise the test vector execution for board-to-board interconnect testing between multidrop boards configured within a system level environment.

2.2.6 Signal Termination

For high-speed test applications or Flash Programming configurations where TCK speeds greater than 10MHz are used, it is advisable to terminate the TCK line using an impedance matched RC network comprising a resistor (close to the characteristic impedance of the TAP cable, e.g. 60-100 Ohms and capacitor e.g. 100pF) in series to ground. All other JTAG inputs should be pulled high using a weak (10kOhm) pullup resistor. To damp reflections a 22 Ohm series resistor can also be implemented between the last device in a chain and the UUT TDO pin on the TAP connector (see figure one).

2.2.7 Edge Connector Access

Where possible access to the boundary-scan chain / chains should be via the edge connector or dedicated TAP connector as this will eliminate the need for bed-of-nails probing of test points or worse - vias, which can lead to unreliable testing due to poor pin contact resulting from no-clean manufacturing processes.

2.2.8 Power Distribution

On 'mixed-signal' boards featuring a hybrid analog/digital design it may be 'dangerous' to drive boundary-scan patterns on the digital portion of the circuit causing random behavior from the analog portion. For example if the analog circuit was providing some power handling or RF transmit capability, this could lead to the potential damage of devices or to the board. This hazard can be alleviated by designing analog and digital subsystems that allow the analog power to be shut-down while boundary-scan tests are executed.

2.2.9 IEEE 1149.6 (aka 'dot6')

To assist with the testing of high-speed serial busses based on LVDS technologies new steps have been taken to adapt boundary-scan to cope with this development. The so-called 'dot6' revision of the standard introduced new BSDL instructions and logic descriptions to perform tests using high speed pulses that could overcome the DC/LF 'obstacles' such as AC coupling capacitors and termination resistors that are part of LVDS networks. By adding a pulse generator circuit to differential driver pins and an edge detection pulse recovery circuit at the receiver/sensor end, 'dot 6' compliant interconnects can be tested. JTAG Technologies ProVision software features a specific test generator module that can process dot6 compliant BSDL models and a modified diagnostics system that has been specifically developed to address 'dot6' circuits. Note also that to fully test AC coupled circuits you must have 'dot6' compliant devices that are fully proven at both receiver and driver points. This may mean that if the signals go 'off board' you have to implement a 'dot 6' compliant 'DIOS' (Digital I/O Scan) module external to the target or undertake to loop-back driver and sensor pins.

2.3 Scan Chain Layout - Signal Integrity

2.3.1 Tracking of TCK and TMS

It is important that the tracking of the TCK and TMS signals are kept as short possible in order to eliminate any 'trace looping'. Specify the routing of these signals within the PCB layout auto route tools as CRITICAL. See also chapter 4 on board layout considerations.

2.3.2 Buffering and fan-out of JTAG Signals

Depending on the number of devices allocated to any given 'scan chain' it may be worthwhile to buffer all the boundary-scan input signals interfacing to the board to preserve signal integrity (in particular TCK and TMS). As a general rule of thumb, if track lengths are relatively short, then fan-out to 6 - 10 devices from a '244 type buffer is acceptable.

However, if track lengths between the buffer device and boundary-scan devices are fairly long (>10 cm), a fan-out to 4 - 6 devices would be advisable. Note also that in addition to the 'standard' R-C network on TCK a pull-up (e.g. 10kOhms) should also be implemented to ensure the TCK buffer inputs are never left floating. Refer to Figure 2). It is worthwhile considering buffering the primary asynchronous test reset signal (TRST) and terminating this signal with a weak pullup resistor (typically around 10K ohm) that can be easily driven by a PC based boundary-scan controller. The buffer ensures that the broadcast secondary TRST signal can overcome the parallel sum of internal device pull-up resistors.

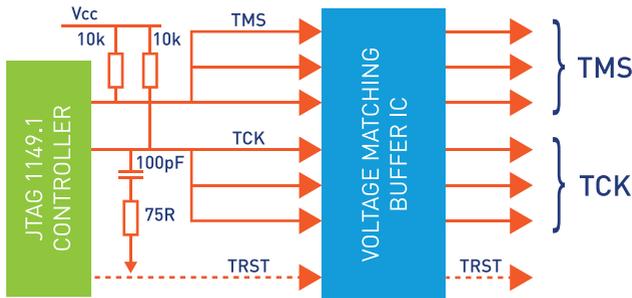


Figure 2 - Fan-out of TCK and TMS

The TRST debate

The original IEEE Std 1149.1 states that TRST~ pins within devices should feature a weak internal pull-up resistor. The reasoning for this is the optional requirement for the TRST~ signal. It was initially thought that if not all controller hardware supported TRST then, on mixed (TRST & NOTRST) designs, TRST on some devices could be left floating and thus in some cases enter the reset phase that shuts down the boundary-scan activity on a device. Providing the TRST pin with an internal pull-up therefore offers a fail-safe to ensure that boundary-scan will always work even if TRST signal is left N/C. The counter argument states that a pull-down on TRST will always mean that boundary-scan activity is disabled and thus device cannot inadvertently enter boundary-scan mode during 'normal' board operation. This approach is often promoted by silicon vendors as a specific DFT guide for their particular device and is not a generic design rule. Designers should assess their own requirements and note that a pull-up on TMS will ensure a soft reset will always remain following power-up or any inadvertent TCK clocks.

2.4 Scan Chain Layout - Physical Bypassing

2.4.1 Physical Bypassing

For designs that are using first issue (working sample) devices it may be sensible to place zero ohm bypass resistors so that via a combination of DNP (do not place) and placed resistors, boundary-scan devices can be physically bypassed within a boundary scan chain as shown in Figure 3.

This cautious implementation is recommended in case the boundary scan logic [of the device] has either not been implemented or has not been fully tested. In which case the result would be an incomplete chain and subsequently reduced or no boundary-scan capability.

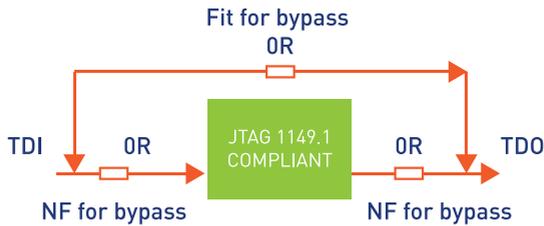


Figure 3a

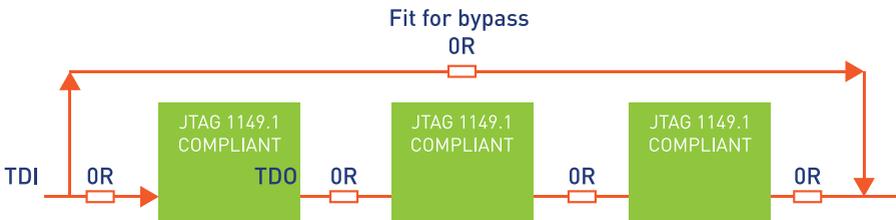


Figure 3b

Figure 3 - Physical Bypass of Single and Multiple Devices

2.4.2 Single Device Bypass

In configuration (Figure 3a) the zero ohm series resistors connected to the TDI and TDO pins will normally be fitted and the BYPASS resistor will be a DNP. However, if this device needs to be bypassed in order to get the chain to function, the BYPASS resistor will be placed and the TDI and TDO resistors removed. The TDI resistor is removed to prevent erroneous instructions from being shifted into the bypassed device instruction register (remember that the TCK and TMS signals will still be connected, subsequently the device TAP controller will still be active). The IEEE 1149.1 Std, specifies that all compliant devices should have an internal pull-up on the TDI pin, which will ensure that an all 1's instruction will be shifted into the bypassed device instruction register - the all 1's value is the BYPASS instruction. The TDO resistor is removed to prevent signal contention between bypass data and TDO data.

2.4.3 Multiple Device Bypass

Configuration (Figure 3b) shows the bypassing of multiple devices, which may be a group of memory devices that are beta release silicon. In this instance it is advisable that the facility to bypass all these devices is implemented within the board design. The only difference between this implementation and the single device implementation, are the additional TDO/TDI zero resistors between each device within the bypass group. This is to ensure that by removing these resistors the Capture-IR instruction values will not be shifted into the downstream device, which may cause erroneous operation of the device causing the device IO pins to be driven to an unknown state.



Note: - These configurations should only be necessary for prototype designs that utilize unknown silicon such as 'engineering samples', as it should be anticipated that any silicon problems are resolved prior to commencing full production.

2.5 Control of Non-Boundary Scan Devices

2.5.1 Access to Control Signals of Non-Scan Logic

Wherever possible ensure that the control signals of non-boundary scan devices are connected to boundary scan enabled pins. Under these conditions devices can be disabled during test thus preventing any signal contention which could cause damage to devices or cause the tests to be unreliable.

In the example below (Figure 4) the non-boundary scan device also accesses the data and address busses that are being tested as part of the interconnect test between the two boundary scan devices. In this instance the OE control pin of the nonboundary scan device is connected to a boundary scan pin so that it can be driven to a logic "1" state to disable this device during interconnect testing. In this way its output pins are placed in a high impedance state and will not impede testing of the data and address busses. This signal is not driven during normal operation (it is held low via the pull-down resistor), but for test purposes it is controlled via a boundary scan cell.

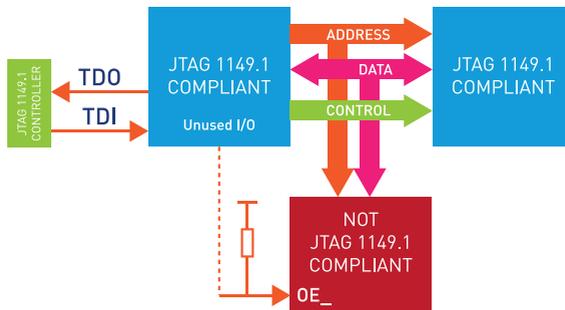


Figure 4 - Control of non-scan devices

It may also be necessary to gain control of clock signal pins that are utilized to synchronize logic or memory accesses. Consequently it is advisable that the on-board clock can be disabled and replaced with a boundary-scan derived test clock as detailed in Figure 5.

2.5.2 Control of Clock Signals

The clock distribution circuit in the left-hand side example (Figure 5a) is an indication of a commonly implemented clock distribution circuit designed without boundary-scan in mind. In this instance the on-board oscillator cannot be disabled because its OE control line is tied directly to VCC, and there is no facility for boundary-scan access to the primary clock distribution signal.

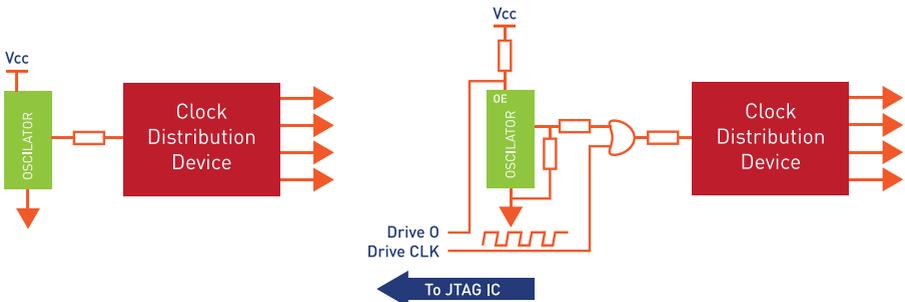


Figure 5a - Discouraged clock distribution circuit

Figure 5b - Recommended clock distribution circuit

However, the implementation in the right hand side example (Figure 5b) allows the disabling of the on-board oscillator by connecting the OE pin to a spare boundary-scan cell (this could be an unused IO pin on an FPGA or CPLD) and driving this signal pin to a logic "0" to place the OUT pin of the on-board oscillator in the high impedance state. This will allow the on-board clock to be 'replaced' by a test clock that is driven from a spare boundary-scan cell functioning in EXTTEST mode via a simple gate. The test clock will be distributed to all the devices controlled via the clock distribution device which may include synchronous memory devices to be tested.



Note: - When using this technique, check on the minimum operating frequency of the clock distribution device as this may require a minimum operating frequency to synchronize with the internal PLL.

2.6 Special Considerations for SRAM based FPGA Devices

2.6.1 Boundary Scan Execution - Pre and Post Configuration

SRAM based FPGA devices are often subject to changes to their boundary scan behavior at different stages during the programming / configuration cycle. During the power-up cycle these devices will enter a configuration phase which will prevent the boundary scan circuitry from being accessed, unless the configuration sequence can be held-off.



Note: - Carefully read the manufacturer's data sheets / application notes to determine whether this may be a problem. In particular any reference to a compliance pattern attribute within the BSDL model file e.g. "(INIT,PROGRAM) (01)"; This specifies that INIT and PROGRAM pins should be held in the logic '0' and logic '1' states respectively to halt configuration place the device in boundary-scan mode.

2.6.2 Xilinx and Altera examples

In older Xilinx FPGAs PROG pins and the INIT pin required attention (the PROG pin needs to be held high to prevent the TAP reset while the INIT pin needed to be tied low so that during power-up these devices are prevented from entering the configuration cycle). In the case of the more recent Xilinx parts only PROG_B needs to be held high. Also if PROG_B transitions high-low during a test the TAP controller is reset and a total test failure is likely. If the FPGA configuration source (e.g. serial PROM) cannot be erased its associated BSDL model can be amended to reflect the post configuration behavior of the board, however this is long-winded and subject to change. An alternative therefore is to allow the FPGA to 'boot', then clear it's configuration by toggling PROG_B via JTAG control, allowing the standard BSDL model to be used.

In the case of the Altera FPGAs you should carefully read the documentation (Altera application note AN039) which is regularly updated. In most cases for the device to exhibit pre-configuration behavior the nCONFIG signal line is held low from power-up (Figure 6) .

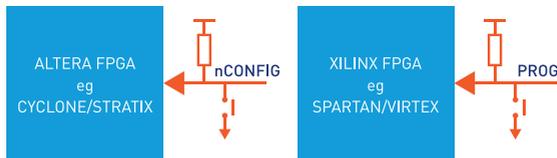


Figure 6 - Critical pins to hold off FPGA configuration - check vendor datasheets at all times

The alternative to executing tests pre-configuration is to wait until the FPGA has configured, at which time JTAG functionality will be available. However, it is often necessary to generate a BSDL file that describes the post-configuration status as some of the I/O pins may now be re-defined following configuration. Refer to section 2.7.

2.7 FPGA Re-configurable I/O Considerations

2.7.1 Pre and Post Configuration BSDL Files

All the I/O pins of FPGA devices are defined as bi-directional in the pre-configured state, which means that each of the I/O pins is connected to three boundary-scan cells as defined in Figure.7a; input cell, output cell and control cell.

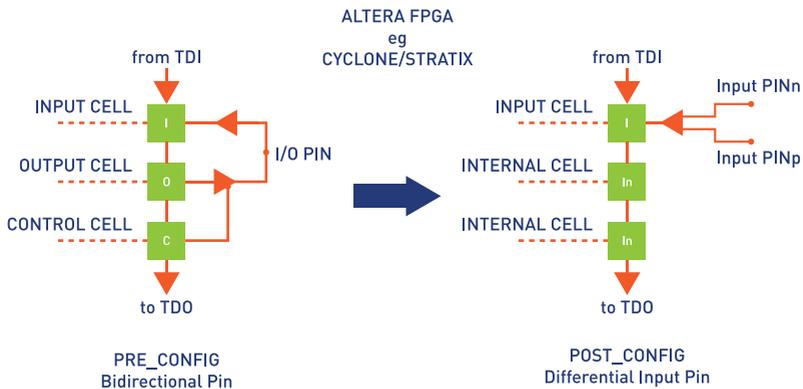


Figure 7a - Pre-configuration

Figure 7b Post-configuration (LVDS - Rx)

A number of the FPGA devices available from the silicon vendors now allow reconfiguration of the I/O blocks (IOB's) to support a wide variety of I/O standards. However, the selectable I/O resource can change the function of the boundary-scan cell architecture behind the reconfigurable IOB. The example in Figure.7b for a LVDS input receiver configuration shows how the boundary-scan cell status has changed to that of a single input cell, with the remaining cells having no boundary-scan function other than that of internal cells to complete the chain.

This means that boundary-scan tests developed for the pre-configured state cannot be used once the device has been configured. In this case tests will need to be regenerated that comply with the post-configuration boundary-scan status as described in a post-configuration BSDL file. In addition for some other boundary-scan functionality such as the 1149.6 support, it is required to configure the FPGA before this functionality becomes available.



Note: The FPGA tool vendors may provide a utility to automatically create a postconfiguration BSDL file post-configuration interconnect testing.

2.8 Flash Programming Optimization

2.8.1 AutoWrite™ Access

The importance of optimizing the flash programming sequence was mentioned earlier (sub-paragraph 2.2.4) which emphasized the need to keep the chain (combined BSR) length to an absolute minimum. This optimization can be enhanced further by utilizing JTAG Technologies AutoWrite™ (AW) feature which utilizes a WE strobe pulse for each flash memory write cycle access, instead of accessing each address three times in order to toggle the WE signal. As shown in Figure 8) this is achieved by routing the WE signal to the edge connector (or to a header if this is feasible) so that it can be connected to the AW (pin 13) pin on the JTAG Technologies TAP pod.

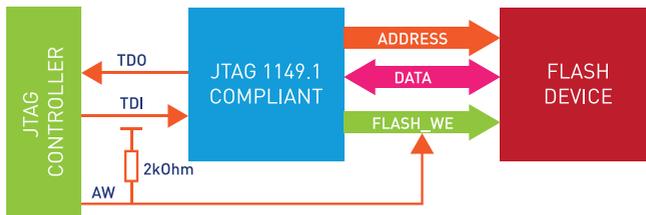


Figure 8 - AutoWrite™ Access

2.8.2 Eliminating Signal Contention

In order to utilize the AutoWrite facility, it is essential that the boundary scan cell that is driving the WE signal (in this example from the processor) can be placed in the high impedance state. If not there will be contention between the processor WE signal and the AW signal generated via the TAP pod. Figure 9) below shows how this is controlled via the control cell associated with the output cell driving the WE signal, and how this is represented within the BSDL file boundary scan register description.

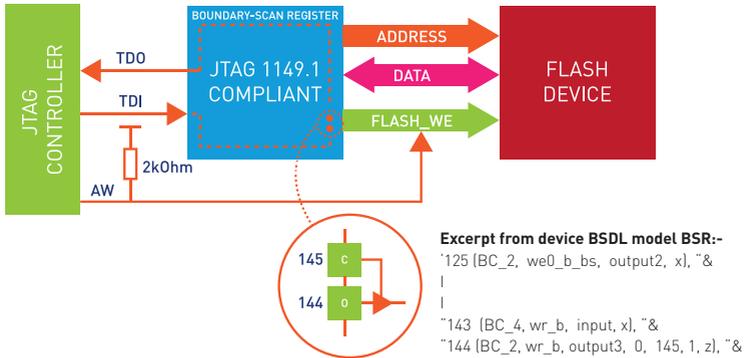


Figure 9 - Eliminating Signal Contention

2.8.3 3-State and 2-State Output Cells

In the above example of the boundary register description the wr_b signal is described as a 3-state output cell in which the signal on this pin can be driven to logic "1", logic "0" and also "HIGHZ". This is also depicted by the cell diagram within figure 9 in which a safe logic "1" value has been shifted into the cell 145 of the boundary scan register so that the output cell can be disabled. However, in an alternative instance the we0_b_bs (cell 125) signal is described as a 2-state output cell in which the signal on this pin can only be driven to logic "1" or logic "0" and cannot be placed in high impedance. In this situation there would be contention between the we0_b_bs signal and the AW signal.

2.8.4 Multiple Output Cell Control

Alternatively the situation may occur where each output cell does not have a unique control cell, and one control cell may control a number of output cells as shown in Figure 10) In this example boundary scan cell 74 not only controls the WE signal but also controls other signals that are connected to the target flash memory device. Subsequently if a safe logic "1" value is shifted into cell 74, all signals associated with this cell will be placed in the high impedance state.

Cell#	type	p_name	p_type	safe_val	ctrl_cell	ctrl_val	dis_state
74	[BC_2,	*	output2,	1), "&
73	[BC_1,	WE,	output3,	0,	74,	1,	Z], "&
72	[BC_1,	OE,	output3,	0,	74,	1,	Z], "&
71	[BC_1,	nSDRAS,	output3,	0,	74,	1,	Z], "&
70	[BC_1,	nSDCAS,	output3,	0,	74,	1,	Z], "&
53	[BC_1,	A25,	output3,	0,	74,	1,	Z], "&
52	[BC_1,	A24,	output3,	0,	74,	1,	Z], "&
51	[BC_1,	A23,	output3,	0,	74,	1,	Z], "&
50	[BC_1,	A22,	output3,	0,	74,	1,	Z], "&

Figure 10 - Multiple Output Cell Control

2.8.5 AutoWrite™ Gating Arrangement

In the scenarios where a two-state WE or a WE which is included in a 'block' tristate set (2.8.4) is used, the AW can only be connected by using an additional gate arrangement. Thus during operational mode the WE signal connects directly from the processor and during manufacturing test, the AW signal can be utilized to optimize the flash programming process. This gating arrangement can either be implemented using glue logic or could be embedded within a cPLD core. An example of this is shown in Figure.11.

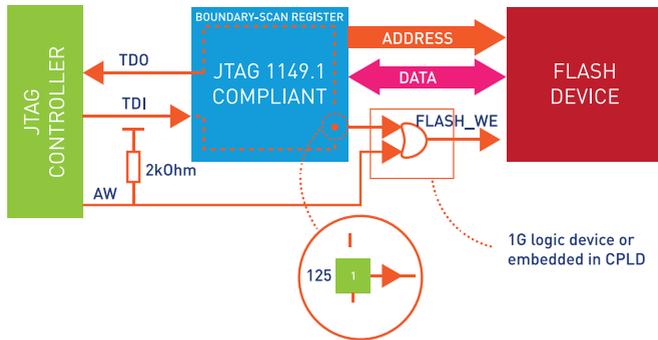


Figure 11 - AutoWrite™ Gating Arrangement

2.8.6 Flash Programming Optimization - Reducing Chain Length

Flash programming times may be affected by a number of factors such as maximum clock frequency, device programming mode (e.g. use of page buffers), use of AutoWrite™ (see above) and also total boundary-scan register length. By significantly reducing the length of the boundary scan chain so that fewer bits are shifted for each flash memory read/write cycle, programming times can be vastly improved. In many instances however overall scan chain length is fixed at the design stage and so to take advantage of this factor it may be necessary to use alternative device types. For example if a large FPGA with a →2000 bit register were accessing a flash via standard buffers why not replace the standard buffers with 1149.1 compliant versions or even low-cost low-pin count CPLDs that accomplish the same task and only use a 100 or so scan cells. In this instance the large FPGA can be bypassed using HIGHZ and the short chain alternative parts accessed.

2.8.7 Flash Programming Optimization - Split Boundary-scan Data Register

Some ASIC designers have addressed the vector optimization issue by designing the boundary scan architecture so that the data register can be accessed as a split register Figure 12). In this design the mandatory EXTEST instruction selects the complete boundary-scan data register comprising, for example, of 650 cells, whereas the device specific FLASH_PRG instruction selects the shorter (shown in dashed red) data register comprising of only 125 cells. This type of configuration is extremely beneficial where the flash memory can be accessed via a custom device and the ASIC designer has the freedom to implement the device specific boundary-scan architecture to meet the board level DFT requirements.

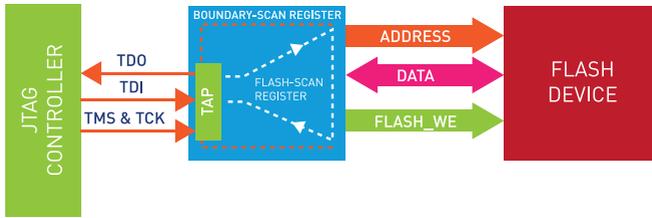


Figure 12 – Split Boundary-scan Register

2.8.8 Flash Programming Optimization - Embedded Flash Controller – NAND Flash

Many alternative techniques for optimizing the flash programming process are presented annually at Test Conferences throughout the world. In the next example a flash memory controller function is embedded within a programmable device (either a cPLD or FPGA) in the form of VHDL code. Access to this embedded controller is subsequently provided via the 1149.1 Test Access Port, Figure 13.

The serial data path TDI/TDO provides access to the Data Register for loading the flash memory address and data values, while the embedded controller includes the programming algorithm for performing read and write access to an externally mapped flash memory device. Programming data is shifted into the embedded data register, while other boundary-scan devices in the chain are placed in BYPASS or HIGHZ. In this particular design the pipelined Data Register can be kept full at a TCK frequency of 8MHz, while read and write access to the flash memory is achieved at system clock speeds, thus allowing write cycles of typically 10us.

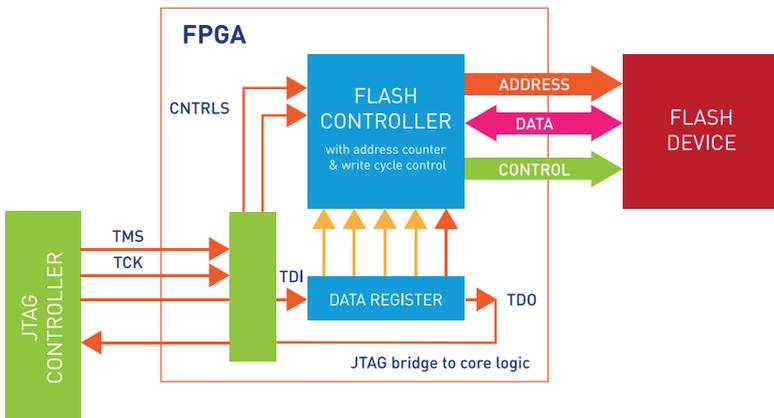


Figure 13 - Embedded Flash Controller

A further mechanism for embedding a flash programming core can be to use Altera's VJI (Virtual JTAG Interface) 'megafunction' or Xilinx equivalent, to access customized core logic via the regular JTAG port. Furthermore by coupling a flash controller core for NAND memories to an additional (existing) high speed interface (e.g LAN, SD-Card or USB) high capacity NAND flash devices can be programmed at systems speeds from external memory stores. In this case the data is not transmitted via the JTAG port, however JTAG control instead orchestrates the overall process.

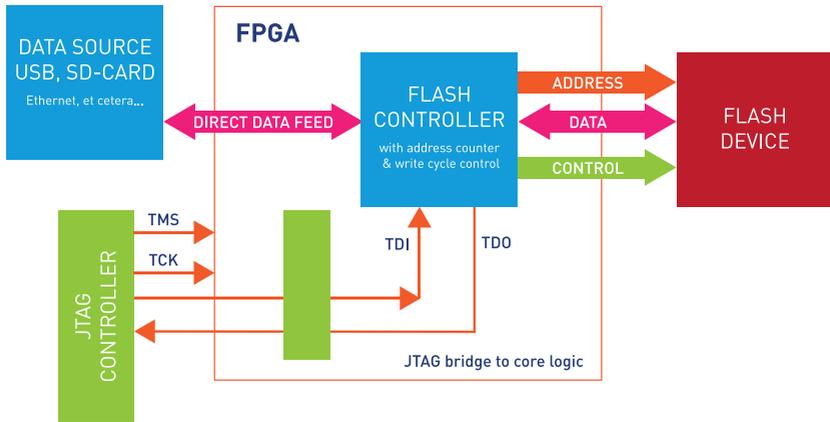


Figure 14 – Embedded Flash controller for high-capacity flash (e.g. NAND)

2.9 Device Specific DFT Requirements

2.9.1 Dual Port Operation - Default Background Debug Mode (BDM)

Some older boundary scan devices require special consideration in order to select the JTAG mode of operation. This is particularly the case where the boundary scan control pins have a dual (multiplexed) function and can be utilized for accessing the IEEE Std. 1149.1 TAP controller and for accessing a proprietary debug port. A problem arises when the default mode on power-up enables access to the proprietary debug port and not the 1149.1 TAP port. Under these circumstances it is not possible to gain access to the 1149.1 TAP controller in order to execute any form of boundary scan tests.

An example of this power-up scenario is shown in Figure 15 below where the device will default to background debug mode unless specific data pins within the device Configuration Mode Register are forced to the logic "1" and "0" state on power-up reset, at which point the 1149.1 TAP controller port will be selected.

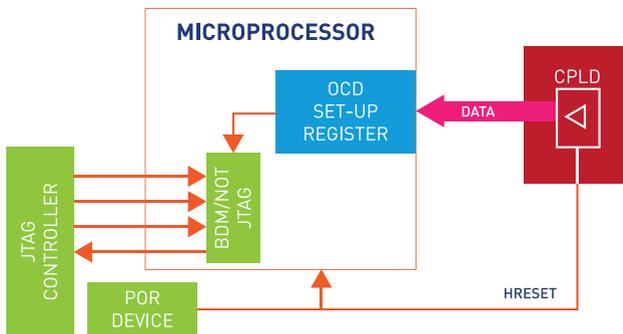


Figure 15 - Forcing JTAG Operation on Power-up

2.10 Design-for-Test - Other Considerations

2.10.1 Control of Watchdog Circuits

Under certain test conditions it is recommended that any on-board watchdog circuit can be disabled throughout the duration of boundary-scan testing. Otherwise the watchdog circuit may reset the onboard processor, which will place this device in the JTAG Test-Logic-Reset (TLR) state causing the device to return to its normal operational mode and subsequently breaking the boundary-scan chain. This situation is most likely to happen during a prolonged test sequence which will exceed the watchdog timeout period e.g. flash programming. If possible, control the watchdog disable circuit via an unused boundary-scan cell or by placing a jumper in the prescribed position on a test header.

2.10.2 Configurable Termination Resistors

It is advisable not to rely on the use of configurable pull-up/pull-down resistive termination within FPGA devices to control logic within a board design because these programmable terminations will not be active while the device is in the pre-configured state. This may lead to unreliable operation during test.

2.11 Embedded JTAG - Instruments & Control

2.11.1 Embedded instruments for enhanced testing

Since 2014 two new standards have emerged that promote the use of so-called 'embedded instruments' usually within FPGAs, SOCs or the newer combinational devices (Gate arrays with MCU cores) that are now available. The standards are IEEE 1687 and IEEE 1149.1 2013 and came about following a period of intense activity around 2010, with the two separate groups proposing similar updates to the existing 1149.1 standard, which was by then 20 years old.

Both groups identified deficiencies in the existing standard and addressed these through the introduction of more 'dynamic' IC infrastructures. In the case of 1149.1 2013 the driver for the changes was to standardise some of the design practices that IC vendors had introduced on a unilateral basis, such as initialisation protocols, individual device id codes and power management scenarios. While in the case of 1687 the main driver was to improve board-level 'testability' through the greater use of embedded test cores (BIST IP) accessed via an extended standardised infrastructure.

The new 2013 extension to IEEE 1149.1 has more than doubled the size of the original standard document and includes the syntax of a new procedural description language (PDL) that is used to define the usage of the dynamic register segmentation and device IP hierarchy for a given application. IEEE 1687 meanwhile also features PDL, however there is only a basic level of compatibility between the two PDLs - apparently due to the different focus of each new standard.

PDL is designed to document the procedures for stimulating and observing test data register fields for 1149.1-2013 and in IEEE 1687, the procedures for stimulating and observing data to an instrument. Not much of a difference except that in IEEE 1687 a second language is required to describe the [embedded instrument] access networks - ICL (Instrument Control Language) while in 1149.1-2003 the access network descriptions are embedded in an extended BSDL model. For complex networks that make extensive use of embedded instruments IEEE 1687s ICL is claimed to be better suited.

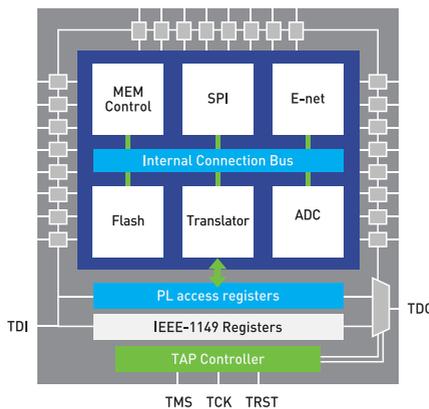


Figure 16 - FPGA configured with JTAG translator 'bridge'.

To date there are no commercial devices in widespread use that comply to either standard, however advanced JTAG users are using embedded instruments through proprietary techniques. One such technique is the use of JTAG Technologies Core-Commander FPGA product. Using CoreCommander FPGA uses can create a 'bridge' between the native JTAG interface of the device and an internal bus (e.g. CoreConnect, Wishbone, AMBA etc.. - see figure 16). CoreCommander routines to implement Memory writes and reads can be embedded in a work-flow such as a Python script program.

The script can then initialize 'instruments' that can be dedicated for a test purpose or that can be an existing IP element that will remain part of the FPGAs 'mission mode'. An example might be a DDR memory controller that can be accessed by the JTAG \leftrightarrow IP bus bridge which is then used for external (discrete device) memory testing, another might be a bus interface controller for CAN I2C Ethernet etc..

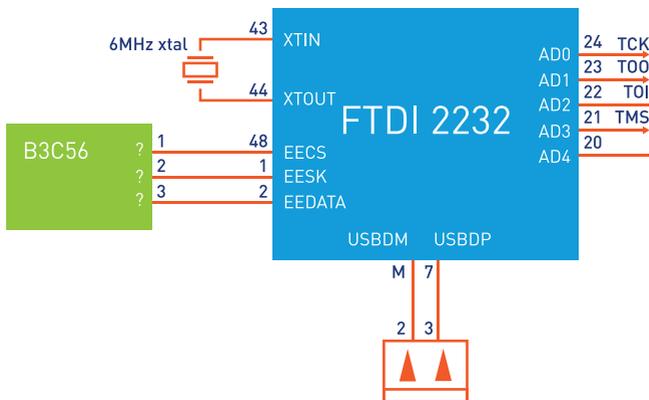


Figure 17 - Embedded USB => JTAG controller/driver

2.11.2 Adding an embedded USB-to-JTAG driver device.

For convenient access to the JTAG infrastructure of a board while it is in service or following a field return, it can be worthwhile to add a built-in JTAG interface rather than use an external instrument. One such device, supported by several boundary-scan vendors including JTAG Technologies, is the FTDI 2232 part. To implement this controller requires only a modest design effort since only the 2232, a serial PROM and 6MHz oscillator are needed. The block diagram in figure 17) gives an overview of the requirements, the designer is however encouraged to contact local JTAG Technologies support for more detailed guidance if required.



Note that it is also possible to use some proprietary FTDI-based programmers as interfaces to JTAG Technologies tools, however care must be taken with regard to the control of TRST pin as this is often arbitrarily assigned. In the cases where TRST appears not to function (pull high during a test) then action must be taken to pull board TRST 'manually'.

2.11.3 Adding an embedded JTAG controller device.

In 2.11.2 the boundary-scan controller function is part of the design yet the test patterns are streamed onto the board via a USB interface that is usually connected to a PC or similar. It is however also possible to add an embedded boundary-scan controller that uses internal resources (e.g. flash memory) to store test vector data that can be invoked on demand or as part of a POST (Power-On Self Test) system. While it is not too difficult to build a 'master' JTAG controller into existing logic (e.g. FPGA or microcontroller with sufficient spare I/Os) deriving test vector data and compressing it into a convenient format is not a trivial task, especially for a complex board or system. For this reason designers looking to employ embedded JTAG control often look at proprietary solutions such as the TI (was National Semiconductor) STA101 or the Firecron JTS01. As well as featuring all the state logic needed to control the boundary-scan chain(s), these devices are supplied with compilers that will take test vector data from principal vendors, such as JTAG Technologies, and compress it into formats that can easily be read out and 'fed' into the controller's parallel interface either directly or via an on-board processor.

CONFIGURING A DESIGN FOR MIOS TESTER

A MIOS tester is a combination of a standard JTAG controller equipped with Mixed-signal IO channels. In most MIOS systems the channels have some fixed functions digital IO, analog IO, frequency measure and pulse generator. JTAG Technologies MIOS systems are also FPGA based and thus can be re-configured to support other tester interfaces such as BDM (background debug mode), CAN, LIN and so forth.

Designing a board with existing knowledge of your tester's capabilities can be advantageous - for example it maybe possible to specify a test pad for a clock to be place on the underside (probe side) of the UUT so that it can be measured by a MIOS channel. Similarly if you are looking to test power rails of the UUT then making a easily accessible test point per voltage will be helpful.

Since MIOS units can also be configured to emulate other test/debug interfaces such as BDM (Background Debug Mode) which may be muxed onto other signal pins some mechanism to perform the switch would be required.

BOARD LAYOUT CONSIDERATIONS

4.1 Track separation

To mitigate the effects of cross-talk interference between data signals (TDI, TDO) and control signals (TMS, TCK) consider separating these lines by use of a ground plane. If possible allow access to TDI/TDO signals by routing them predominantly on the top or bottom layer of the board such that they can be accessed for infrastructure test debug purposes.

4.2 Placing of clock terminator

Every layer change or stub line will give an impedance change with a corresponding reflection coefficient. $(Z_o - Z_i)/(Z_o + Z_i)$. In the picture below (figure 18a) termination close to the end of the clock trace, point 1, provides optimum characteristics. Termination at points 2 and 3 at the end of long 'stubs' will result in ragged signals and possible false clocks. Therefore wherever it is feasible make 'stubs' as short as possible, alternatively terminate the clock at a central star point (see figure 18b).

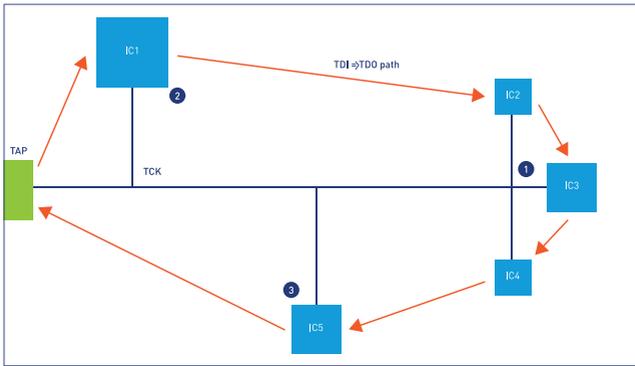


Figure 18a - TCK termination points

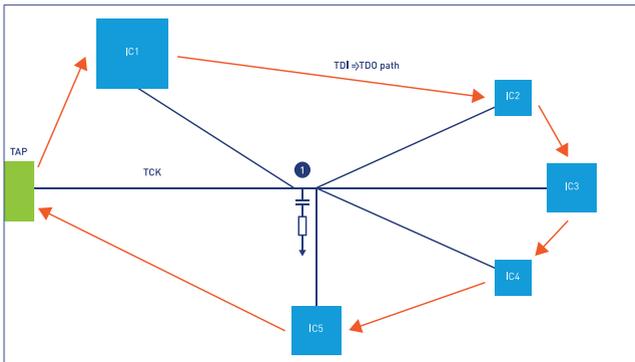


Figure 18b - TCK central termination

A further solution is to introduce separate clock lines from the point of entry to the PCB and terminate each line separately (see figure 18c) however the sum of the resultant capacitance from the terminations can 'load' the clock driver. In this case an additional buffer device could be employed.

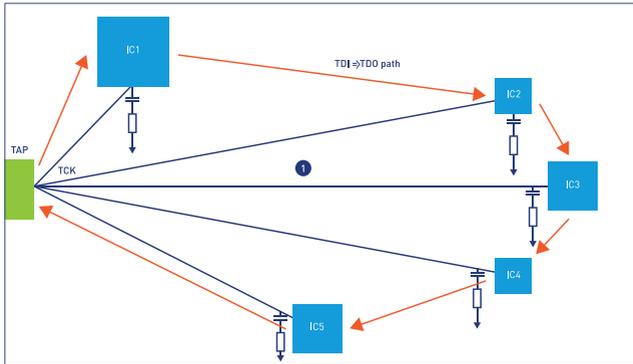


Figure 18c - Best termination point for TCK –star layout with multiple R-C networks

4.3 The effect of BGA packages

The use of BGA (ball grid array) parts in a design will also introduce additional effects. So-called 'power plane killers' they effectively punch square holes on power and ground planes due to clearance issues between holes, pads and traces. This ultimately has a significant effect on the impedance of signals around the holes/vias and thus the designer must be aware that different termination values must be considered for JTAG signals (specifically TCK) at trace entry points to these areas.



For more information of JTAG Technologies:

If you want to apply boundary-scan for testing or in-system programming and you need more help or you need product information, please contact:

JTAG Technologies' Sales and Customer Support



Offices

For contacting JTAG Technologies' local sales representatives, see our website at www.jtag.com/contact-us

IEEE Standards

- IEEE Std 1149.1-2013 - IEEE Standard Test Access Port and Boundary-Scan Architecture (Supersedes former issues IEEE 1149.1-1990 (Including 1149.1a-1993, 1149.1b-1194) and IEEE 1149.1-2001)
- IEEE Std 1149.6-2015 - IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks (Supersedes IEEE 1149.6-2003)
- IEEE Std 1532-2002 - IEEE Standard for In-System Configuration of Programmable Devices (Supersedes IEEE 1532-2000, 2001)

For more information on the IEEE Standards

IEEE Customer Service

445 Hoes Lane, PO Box 1331, Piscataway NJ 08855-1331 USA

T : (800) 701 4333 (within the US and Canada) F: (732) 981 9667

T : (732) 981 0060 (outside the US and Canada) E: customer.service@ieee.org

W : www.ieee.org

For more information on JEDEC

If your company designs its own ASICs with boundary-scan and you want to get a JEDEC Manufacturer ID Codes (JEP106), please contact JEDEC - Technical Affairs of the Electronic Industries Alliance (EIA) at:

JEDEC - Technical Affairs, 2500 Wilson Blvd, Arlington VA 22201-3834, USA

T : +1 703 907 7558 E: kenm@eia.org

F : +1 703 907 7583 W: www.jedec.org

Europe and ROW

T: +31 (0) 40-2950870

F: +31 (0) 40-2468471

E: info@jtag.nl

United Kingdom & Ireland

T: +44 (0) 1234-831212

F: +44 (0) 1234-831616

E: sales@jtag.co.uk

USA and Canada

T: (Toll Free) 877-FOR-JTAG

F: 410-604-2109

E: info@jtag.com

China (Malaysia, Singapore, Taiwan, Thailand)

T: +86 (021) 5831-1577

F: +86 (021) 5831-2167

E: info@jtag.com.cn

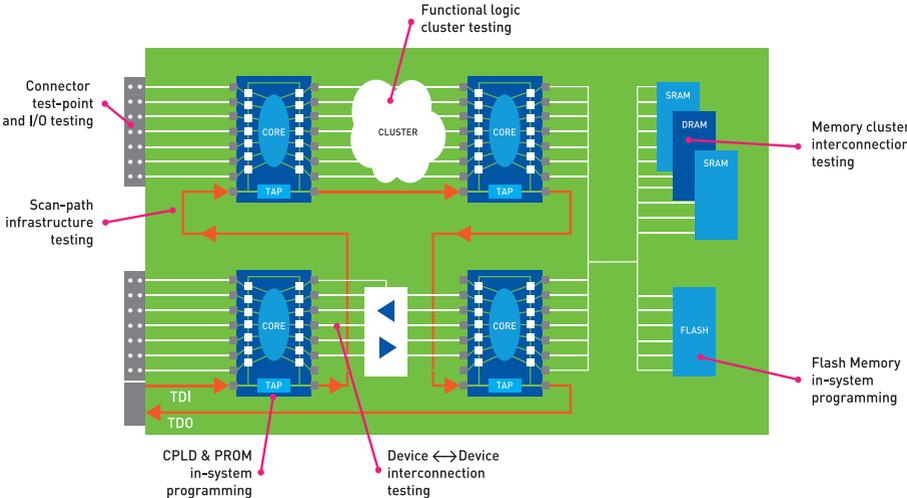
6

BOUNDARY-SCAN - BASIC PRINCIPLES

The IEEE 1149.1 standard defines a four-wire serial interface (a fifth wire is optional) to access complex integrated circuits (ICs) such as microprocessors, DSPs, ASICs, and CPLDs. Most of today's complex ICs are already supporting this feature. Any compatible IC contains shift registers and a state machine to execute the boundaryscan functions. Data enters serially the chip on the TDI pin and leaves the chip on the TDO pin. The boundary-scan logic is clocked by the TCK signal, independent of the system clock. The TMS signal controls the state machine and TRST* is optional for a hardware reset signal.

Multiple scan-compatible ICs may be serially interconnected on the printed circuit board, forming the boundary-scan chain, and a board may contain more than one scan chain (see picture at next page). The scan chain provides electrical access, from the serial TAP interface, to every pin on every IC that is part of the chain.

In normal operation, a scan-compatible IC performs its intended function as though the boundary-scan circuits were not present. However, when testing or in-system programming is to be performed, the device's scan logic is activated. Data can then be sent to the IC and read from it using the serial interface. This data may be used to stimulate the device core, drive signals outward to the PCB, sense the input pins from the PCB, or sense the device outputs. The scan modes of operation provide the ability to test a board for manufacturing structural faults and to perform in-system device programming all via the standard JTAG TAP.



The logo for JTAG Technologies, featuring the letters 'JTAG' in a large, stylized, white serif font. Below 'JTAG' is the word 'TECHNOLOGIES' in a smaller, white, sans-serif font. The background is a dark blue gradient with abstract, glowing blue geometric shapes and bokeh light effects.

We *are* boundary-scan.®



0020-Board DFT-E

© 2002-2019 The JTAG Technologies logo and other trademarks designed with the symbol "®" are registered trademarks of JTAG Technologies. JTAG Technologies reserves the right to change design and specifications without notice.

JTAG
TECHNOLOGIES®

www.jtag.com