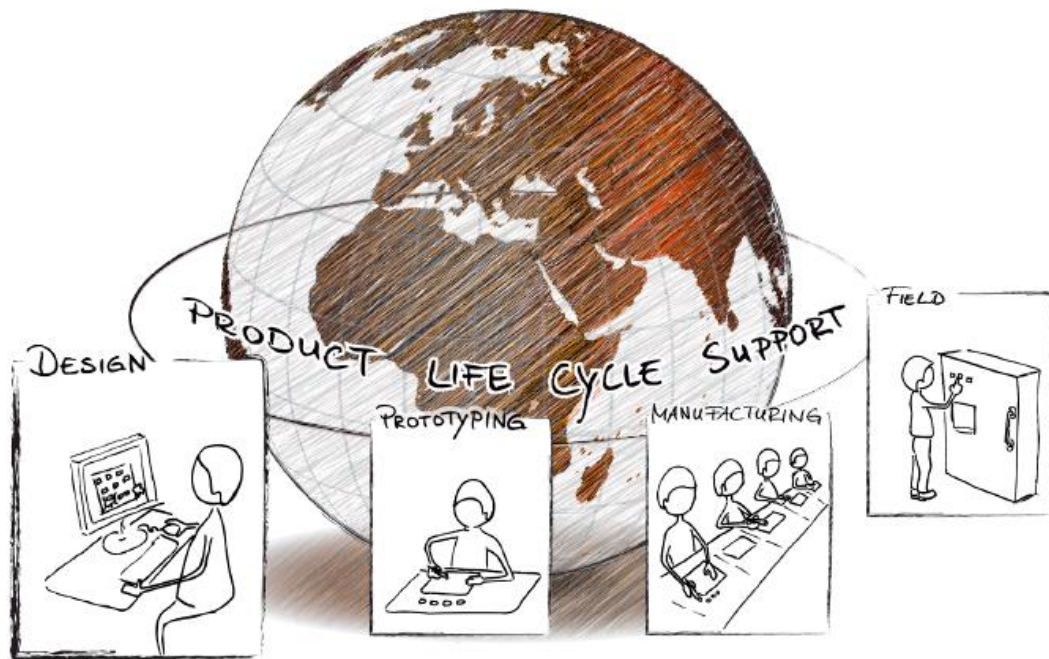




Design for Test (DfT) for Embedded Board Test

Foresighted Board Level Design for Optimal Testability and Coverage



Content

1.	Design for Test - Embedded Board Test	5
1.1	Why Design for Test?	5
1.2	Why should you care about Embedded Board Test?	5
1.3	Embedded Board Test Elements	7
2.	Chip Level DfT	9
2.1	Insert IEEE 1149.1 functionality into ASICs	9
2.2	Insert BIST functionality into ASICs	9
3.	Board Level DfT – Main ICs	10
3.1	Select IEEE 1149.1 compliant ICs, if possible or necessary.....	10
3.2	Request accurate BSDL files from IC manufacturers	11
3.3	Verify standard compliance of the BSDL files	11
3.4	Check BSDL files for compliance conditions and design warnings.....	11
3.5	Check BSDL files for BIST capabilities.....	11
3.6	IEEE 1532 – Advantages over SVF or JAM/STAPL	11
3.7	Place Board-Logic Functionality into FPGAs	12
3.8	Watch the watchdog!.....	12
3.9	Design for testability is team work	12
4.	Scan Chains & Debug Ports.....	13
4.1	Test/Debug Port design – be consistent.....	13
4.2	Test/Debug Port layout and placement	13
4.3	Ensure proper TAP pin connections.....	13
4.4	Use pull-resistors to set compliance pins, boot pins or test control pins.....	14
4.5	Mind the PCB layout of test bus signals.....	14
4.6	Optimize test bus signal terminations	14
4.7	Maximum scan chain device number	16
4.8	Scan chain voltage levels	17

4.9	Help test engineers in quickly locating scan chain errors	17
4.10	BScan ICs that power other BScan ICs & EXTEST	17
4.11	Testing of assemblies of multiple boards	17
4.12	Allow to bypass ICs that are not mounted or not compliant	18
4.13	FPGA configuration and Boundary Scan	18
4.14	FPGA Concatenation	19
4.15	FPGA/CPLD test designs.....	19
4.16	When to place devices in a separate scan chain?	18
5.	Non-Boundary Scan Signals and Devices	20
5.1	Provide access to control signals	20
5.2	Provide means to disable non-Boundary Scan ICs completely.....	20
5.3	Make logic clusters controllable, whenever possible.....	21
5.4	Spare pins – BScan vs. GPIO vs. analogue... ..	21
5.5	Access to all signals of memory ICs	21
6.	Programming	22
6.1	Access to programmable non-Boundary Scan ICs.....	22
6.2	Enhance programming speed with direct access to Write Enable pin.....	22
6.3	TCK frequency impact	23
6.4	On-chip Flash memory	23
7.	Extending Boundary Scan Test Coverage.....	24
7.1	Utilizing Boundary Scan throughout the product life cycle	24
7.2	Utilize test modes in non-Boundary Scan ICs (e.g. NAND-Tree)	24
7.3	Test of analogue circuitry	25
7.4	Test of optical components.....	26
7.5	Test of switches.....	26
7.6	Clock signal verification	27
7.7	Clock signal measurement	27

7.8	Improving coverage using I/O tests.....	28
7.9	Improving coverage using at-speed tests (VarioTAP® and VarioCORE®).....	28
7.10	Improving coverage by combining various test methodologies	29
7.11	System test – Scan Router ICs.....	31
8.	Select an Embedded Board Test System.....	32
8.1	Software	32
8.2	Hardware.....	33
8.3	Licensing.....	33
9.	Glossary	34
10.	Figures.....	36

1. Design for Test - Embedded Board Test

1.1 Why Design for Test?

Many of today's products contain quite complex components for information collection, processing and exchange. Besides, many of those components come in packages whose tens or hundreds of pins are either very narrow pitch or not even visible anymore.

For many products, quality is an issue, not only for cost reasons but also for customer satisfaction; losing a reputation might be costly at the end or even lead to bankruptcy of a company. And a good product quality can be assured by good testing and that includes some preparations, known as Design for Test or DfT.

But how much cost for test is good for a product? On the other side, how much cost will arise from not testing? The test efforts are different for every individual product and need to be calculated to find the optimum level in selecting the right type of ATE. Tools like Aster's TestWay (www.aster.fr) can help in this situation.

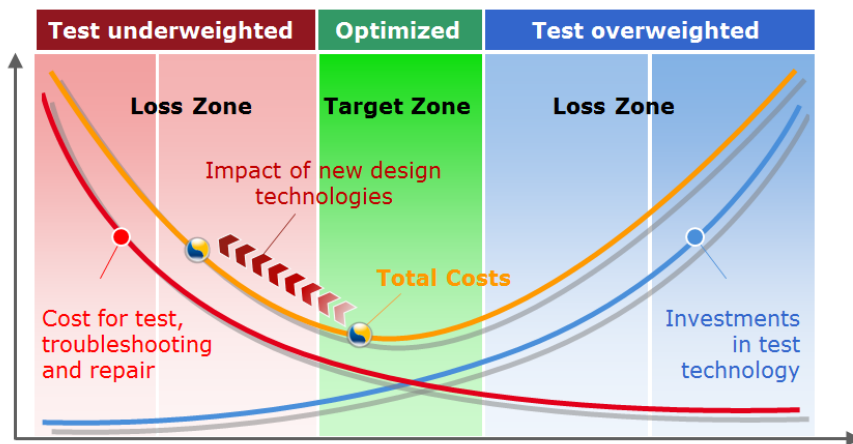


Figure 1-1 Troubleshooting and repair cost vs. test technology cost

1.2 Why should you care about Embedded Board Test?

Since years we see continuously reducing test access due to today's progress in silicon (MCM, SiP, SoC...), at component level (pin numbers and pitch) as well as in PCB manufacturing (buried layers, inserted components, flexible PCBs). Besides, signal transmission becomes more and more complex due to modern high speed demands.

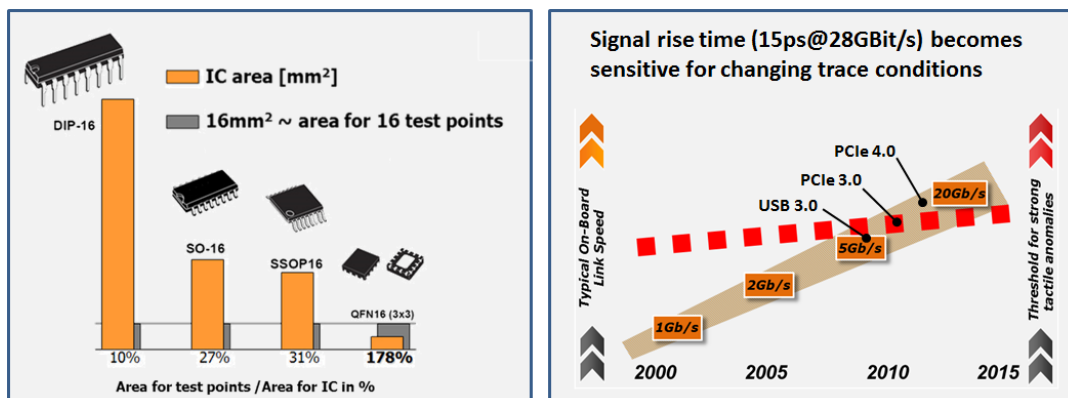


Figure 1-2 Device complexity & signal transmission evolution

Embedded Board Test combines non-invasive board test methodologies such as Boundary Scan, Processor Emulation, Chip-embedded Instruments, Embedded Diagnostics and others by accessing test resources through an electronic interface. Often it makes sense to combine EBT with traditional ATE in order to achieve optimal coverage and test performance.

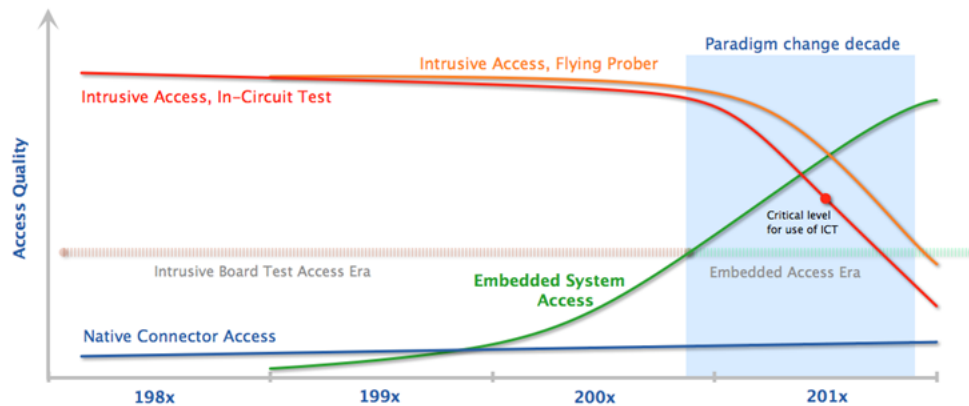


Figure 1-3 Board Test Paradigm Change

Here are some reasons why to care about EBT:

- **Regain test access** - think of test access problems with BGA, multi-layer PCBs, buried signal traces...
- **Reduce cost per test step** - reuse tests throughout the product life cycle; simplify fixtures; use less expensive equipment...
- **Reduce time-to-market** - debug prototypes very early, detect defects and diagnose faults down to pin level
- **Improve yield and productivity** – benefit from fast test execution and precise diagnostics in manufacturing tests
- **Reduce physical stress** - allow less risk for UUT damage by probing
- **Test and programming in one step** - In-system configuration/programming for CPLD/FPGA, Flash, EEPROM and MCU-on-chip memory allows for just-in-time programming, less inventory management and reduced costs compared to pre-programmed devices
- **Optimise your test strategy** - combine test methodologies and test less while achieving the same coverage
- **Reduce or eliminate NDF (No Defect Found) issues** - in-field, in-system test and error logging; reduce cost of repair and/or warranty returns

Summary: EBT increases product quality and lowers overall product cost.

1.3 Embedded Board Test Elements

This document focuses on design for test guidelines related to Embedded Board Test (EBT), combining various board level test technologies such as Boundary Scan (BScan), Processor Emulation Test (PET), Chip Embedded Instruments (CEI) and JTAG Embedded Diagnostic OS (JEDOS).

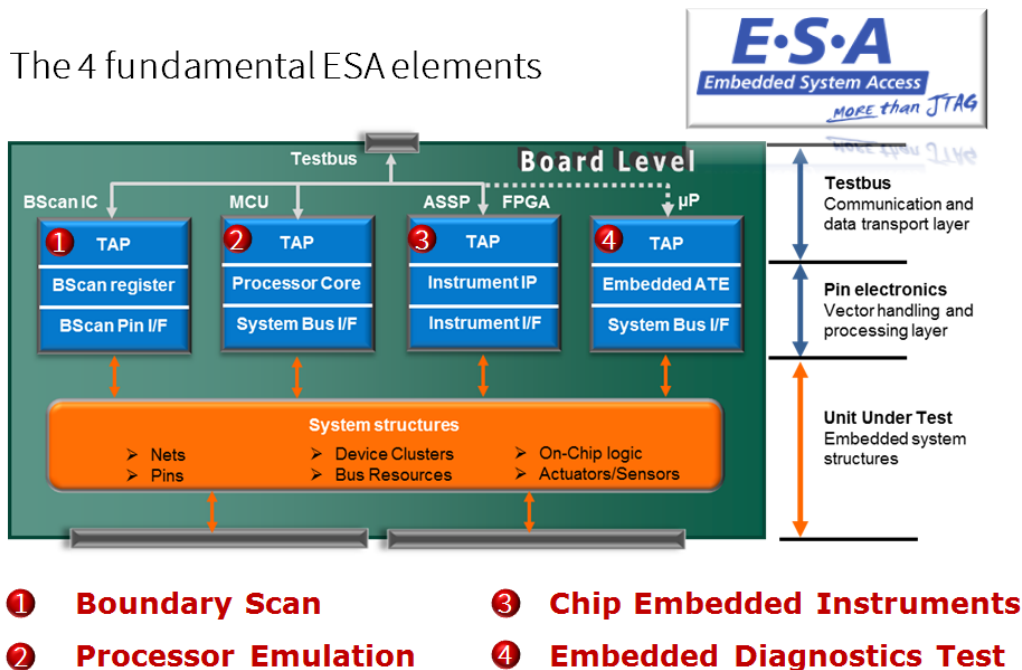


Figure 1-4 Embedded Board Test

Boundary Scan IEEE 1149.1

Boundary Scan was the first test methodology to become an IEEE standard. It was developed by the JTAG consortium; its initial version was balloted on and approved in 1990 and the standard number is IEEE 1149.1. JTAG stands for Joint Test Action Group, which was a group of interested parties that set out to develop the test methodology that became IEEE 1149.1. Meanwhile, the 2013 standard version is approved, dealing with today's challenges in board level testing such as dynamic scan chains, power domains, power-up and -down features in ICs and more.

The 1149.1 standard defines features to be designed into an integrated circuit that provides access to its digital I/O pins from the inside of the device. For the first time, this allows circuit nodes on the PCB to be accessed with device internal test features, rather than with a bed-of-nails fixture or with moving probes. The primary application, for which Boundary Scan was initially developed, was to detect and diagnose manufacturing defects related to connectivity at the board level, such as stuck-at-0 and stuck-at-1 faults, open solder joints, and shorted circuit nodes. Today, the test access port defined in IEEE 1149.1 is used for many additional applications, such as in-system programming, access to built-in self-test, on-chip emulation and debug resources, and system level test.

IEEE 1149.1 plays an important role because it detects shorts and opens on pin level without any functional precondition; an excellent prototype verification mechanism and outstanding manufacturing test tool!

Boundary Scan tests can be developed very rapidly and early in the design cycle, typically as soon as a schematic design of the UUT is available, even prior to having the layout of the PCB finished.

Many standard development efforts built on the original work by reusing features defined in 1149.1. One example is IEEE 1149.4, which defines analogue device test features or IEEE 1149.6, which defines testing of AC-coupled and differential networks. IEEE 1532 defines in-system programming features for FPGA devices, essentially providing a common method to program devices from different vendors. A number of additional standardization efforts related to JTAG / Boundary Scan have meanwhile been completed (e.g. IEEE 1149.7, IEEE 1149.8.1, IEEE 1500, IEEE 1581, IEEE 1687...) or are under way (e.g. IEEE P1149.1.1, IEEE P1149.10, IEEE P1838, SJTAG...)

Processor Emulation Test

Processor Emulation Test (= VarioTAP® technology in GOEPEL electronic) utilizes the debug interface (implemented in many µP for software validation) to transform the processor core temporarily into a test controller. In this case, the processor itself becomes the access point for the connected circuitry in the target system.

Remote-controlled through the JTAG interface or some other debug interface, the processor core utilizes write and read access to the system bus with respective test vectors in order to manipulate and test the connected internal and external resources and components. No operating system or flash firmware is necessary to accomplish this.

The technology can detect both static and dynamic defects; however, diagnostics are limited due to the functional test approach. PET complements Boundary Scan very well and enables or improves especially the test of dynamic components such as DDRSDRAM, high-speed I/O interfaces, and other non-scanable components at chip, board, and system level.

Chip Embedded Instruments

Chip-embedded Instruments are IP blocks integrated into ICs, often accessible through the JTAG port or another debug interface. The functionality of Chip-embedded Instruments is completely open and ranges from simple elements like sensors to complex signal processing units like analysis instruments and programming engines.

Chip-embedded Instruments have been utilised for years in chip testing, for instance, in the form of a built-in self-test (BIST). However, access to these instruments has not been standardised, a situation that has changed with the launch of the new standard IEEE 1687 (also known as IJTAG).

The IP is either integrated permanently in the chip (hard macro) or can be temporarily instantiated and configured (soft macro) in FPGA devices. This specific form is also referred to as „FPGA Assisted Test (FAT)“ or „FPGA Assisted Programming (FAP).“ Another example for FPGA embedded instruments are logic scopes integrated into FPGA's MGT pins (= ChipVORX® technology in GOEPEL electronic).

JTAG Embedded Diagnostic Operation System

JTAG Embedded Diagnostics Operating System (JEDOS) is introduced as a new technology for embedded test of complex electronic designs.

By its architecture, JEDOS represents a complete operating system that uses the natively integrated processor to execute embedded diagnostic functional tests in real time. It is loaded and controlled through the JTAG interface or an alternative debug interface, native firmware is not required. This offers the software developer pre-verified prototype hardware and more efficient fault isolation.

JEDOS offers a wide range of different functions for test, validation and calibration as well as programming. Of particular interest are calibration functions for a DDR RAM controller to verify access security, or to get optimized initialization parameters through appropriate margin tests.

For additional information please contact GOEPEL electronic or your local representatives.

2. Chip Level DfT

This document focuses only on board level topics. We assume the devices to be ready for testing and programming and take them "as is". Of course the chip manufacturers must have placed appropriate resources into the devices in use. Resources include the JTAG interface as defined in IEEE 1149.1 or any other debug interface allowing access to IC-internal register and memory structures.

2.1 Insert IEEE 1149.1 functionality into ASICs

If you have influence to the development of ASICs used on your UUTs we strongly recommend you insist on the implementation of Boundary Scan, which will improve test coverage and reduce test development time drastically.

2.2 Insert BIST functionality into ASICs

BIST can help to verify proper device internal circuitry functions. Alternatively, BIST can be executed from a device to connected components. BIST must be built in by the device manufacturer and can tremendously improve the fault-coverage by verifying at-speed functional behaviour. However, BIST provides typically go/no-go information only.

3. Board Level DfT – Main ICs

The design rules discussed in this document are guidelines that support optimal test coverage and reliable execution of Embedded Board Test technologies, in most cases referring to Boundary Scan. The JTAG interface is the reason because it has been defined and adopted meanwhile widely within the industry. PCB's "real estate" such as signal trace layout, functionality, cost considerations and others may necessitate deviations from these guidelines.

Because we are talking about Embedded Board Test and programming, main ICs may be components that feature different types of debug interfaces, such as SWD, DAP or others that are supported as well.

3.1 Select IEEE 1149.1 compliant ICs, if possible or necessary

But what does "compliant" mean? Well, 1149.1 compliant devices follow the standard 100%. Be carefully, if you read about Boundary Scan or 1149.1 "compatibility" or compatible devices. In such cases, search for additional information or request them from the chip manufacturer. Your test system provider may have some experiences as well but this is unlikely when it comes to very new or very specific ASIC devices, that's why: prefer **compliant** devices and models.

In general we can say: The more JTAG/Boundary Scan ICs are available on a UUT (and enabled for test), the better is the coverage thanks to the JTAG/Boundary Scan pins. Diagnostics can be improved with multiple Boundary Scan pins connected to the same net. There were times when you could choose between a IC version with and without Boundary Scan capabilities. These times are over; most complex devices today feature JTAG/Boundary Scan. However, if you face a situation with one or multiple components (with many pins) that do not have JTAG/Boundary Scan build-in, you may think about a different design.



Figure 3-1 PCB with multiple BScan devices, memory types and interfaces

3.2 Request accurate BSDL files from IC manufacturers

In case of ASICs or new ICs, however, you should request BSDL files directly from the device manufacturer. Do that early enough to have them on hand when you start UUT test development. Furthermore, inquire that the device vendor has verified the BSDL file, in many cases respective notification is made within the BSDL file.

3.3 Verify standard compliance of the BSDL files

Consider use of a BSDL Syntax Checker (e.g. www.goepel.com/bsdl-syntax-checker) to verify syntax and semantics of the BSDL file. Btw, the same verification is performed when importing a BSDL file into the SYSTEM CASCON™ device library.

3.4 Check BSDL files for compliance conditions and design warnings

Compliance conditions specify pins that allow Boundary Scan tests. This optional BSDL section is identified by the keyword COMPLIANCE_PATTERNS. Make sure the Pins are connected to the listed logic level.

Design warnings are optional paragraphs of specific requirements for the Boundary Scan ICs, identified by the keyword (attribute) DESIGN_WARNING. An example is the advice of altered test behaviour of a configured FPGA.

3.5 Check BSDL files for BIST capabilities

BIST functionality is built in by the device manufacturer and can tremendously improve the fault-coverage by verifying at-speed functional behaviour. However, BIST provides typically go/no-go information only. Information about availability and control of BIST is generally specified as a BSDL extension. However, many device vendors implement BIST for device level test but never advertise their existence in BSDL files or datasheets.

The optional keyword (attribute) RUNBIST_EXECUTION indicates the existence of BIST in a particular device and describes implementation details of RUNBIST; consider using them for board level test, especially if there is memory BIST available that typically verifies externally connected RAM access.

3.6 IEEE 1532 – Advantages over SVF or JAM/STAPL

IEEE 1532 specifies the in-system configuration features (programming, erase, read, etc.) of compliant PLDs and FPGAs. One key benefit of IEEE 1532 is that it allows the simultaneous handling and concurrent configuration of multiple devices, even devices from different vendors. Some PLD and FPGA vendors provide BSDL files with and without IEEE 1532 specifications. We suggest you use the files with IEEE 1532 specifications. This way you are prepared to use these functions within your project if you later choose to do so.

IEEE 1532 – Principle

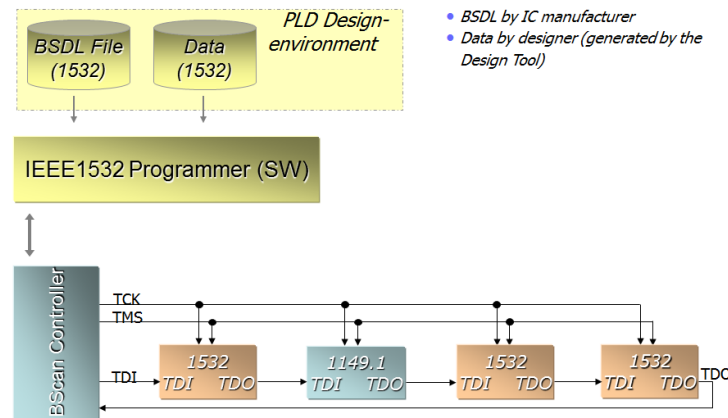


Figure 3-2 IEEE 1532 Scenario

3.7 Place Board-Logic Functionality into FPGAs

Instead of using single logic elements, like gates, multiplexers, flip flops etc., consider putting logic into FPGA devices, since those types of devices typically include IEEE1149.1 features as well as extended test possibilities (FAT), while simple logic components don't.

3.8 Watch the watchdog!

A watchdog device or function on a board can disturb test operation in many ways. Besides, they might be hard to discover because the board is reset from time to time during test execution, resulting in scan chain errors. Watchdog devices typically expect a trigger (e.g. an "alive" signal generated by the UUT itself). Provide means to disable the watchdog, using a test point or with a Boundary Scan pin.

3.9 Design for testability is team work

Assemble a team of beneficiaries to discuss the implementation of EBT early at the design stage (include design engineers, production test engineers, field service engineers). Don't forget to involve management and procurement to evaluate monetary savings throughout the product life cycle.

4. Scan Chains & Debug Ports

4.1 Test/Debug Port design – be consistent

If you use embedded test in multiple products, all of these UUTs should ideally use the same type of connector and pinout for the same interface connector. This way one type of cable can be used to connect the different UUTs to the test system hardware. The figure below shows the JTAG port pinout used by GOEPEL electronics hardware families.

Pin	Signal	Direction	Note
1	TCK	out	
2	GND		
3	TMS	out	
4	GND		
5	TDI	in	must be connected to the UUT's TDO
6	GND		
7	TDO	out	must be connected to the UUT's TDI
8	GND		
9	/TRST	out	
10			do not use




Figure 4-1 JTAG port pinout

4.2 Test/Debug Port layout and placement

Consider putting the test/debug port connector close to the edge of the PCB. This way a cable plugged into the connector has less impact within adaptations, e.g. in ICT or FPT environments.

In addition, you may also want to route the test/debug signals to spare pins on an edge-connector that is also used for functional connections. This would allow access to the scan chain even if the TAP connector is not mounted or if the TAP connector cannot be reached easily. Besides it can simplify the test setup (fewer cables to handle), e.g. when Boundary Scan and Functional Test are integrated in one test setup.

4.3 Ensure proper TAP pin connections

“Daisy-chain” is the most common scan chain design; the TAPs of all Boundary Scan devices are connected in series; TDO of one device is connected to TDI of the next device in the chain. However, TCK and TMS on all Boundary Scan ICs must be connected in parallel to all devices in order to allow proper TAP operation.

If any of the devices features a /TRST signal, make sure to connect this signal at the test bus connector. Often, the /TRST pin is tied to GND with a pull-down resistor, in this case the test system needs to force the /TRST signal to logic high in order use Boundary Scan. Never tie /TRST signal directly to GND, since this would permanently disable the test mode of a Boundary Scan component.

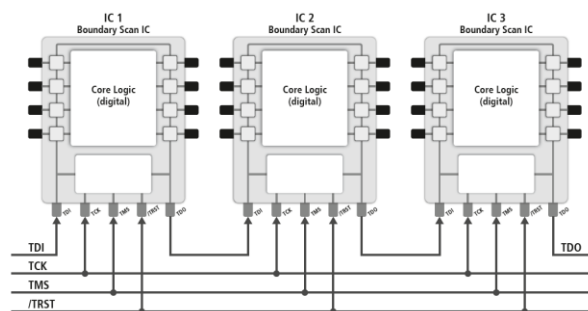


Figure 4-2 Daisy chain scenario and TAP pin connections

4.4 Use pull-resistors to set compliance pins, boot pins or test control pins

When certain control pins (e.g. compliance enable, boot or other control pins) of Boundary Scan devices need to be kept at a single logic level for normal operation (mission mode), make sure to use pull-resistors. Additionally, connect a test point or connector pin in order to force that pin to the appropriate level for entering and staying in test mode.

Keep in mind that compliance enable conditions specified in BSDL files (sometimes also in data sheets) need to be satisfied for respective test operation, in the BSDL file double check for the COMPLIANCE_PATTERNS attribute.

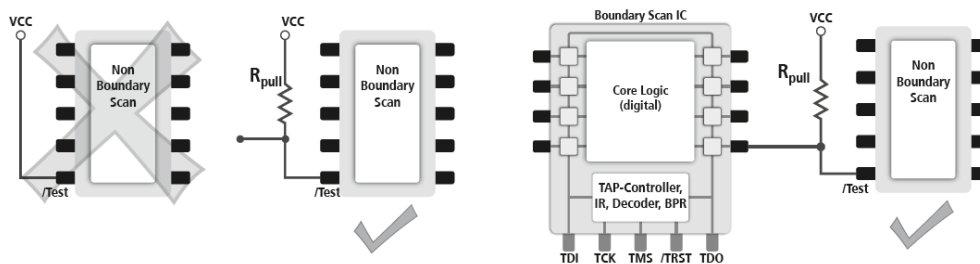


Figure 4-3 Control pins

4.5 Mind the PCB layout of test bus signals

When routing test bus signal traces on the PCB layout, keep in mind that the signals need to be clean, dramatic overshoots need to be avoided and ringing needs to be minimized. TCK, TMS, and /TRST should be laid out similarly (e.g. trace length and width). Avoid a star-type layout and provide proper signal termination to avoid reflections. Avoid crosstalk by keeping signals at an appropriate distance and/or by shielding signals. Follow the layout design rules that generally apply to high-speed signals.

4.6 Optimize test bus signal terminations

TCK, TMS, and /TRST need to be treated as high-speed signals because they are connected to all of the devices in a scan chain in parallel. Therefore, they require clean signal edges and little overshoots. The optimum is a bus layout, where the termination is provided at the last device. If your CAD tool provides a respective feature, use it.

In case of improper signal termination the TAP synchronization is ruined and results in various scan chain error messages. The system has no chance for stable operation.

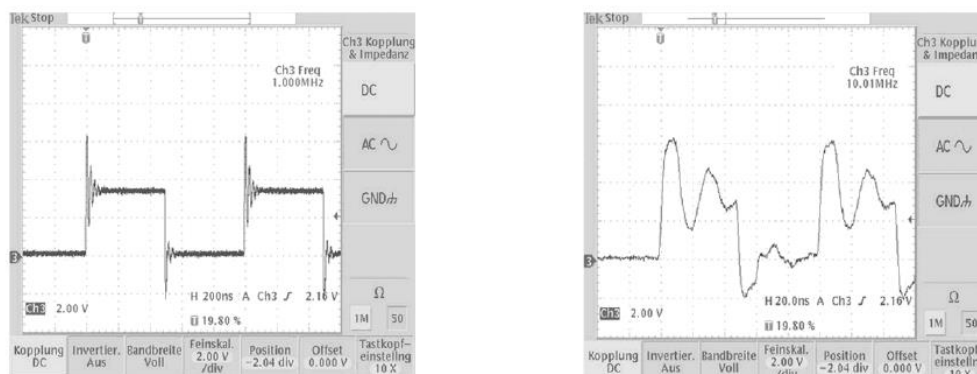


Figure 4-4 Signal overshoots and "extra clock cycles" due to reflections

A typical termination scheme that fits for many designs can be built by using a small serial resistor (e.g. 20-70 Ohm) close to the test system's output driver and a pull-down resistor (e.g. 220-330 Ohm, optionally in series with a pull-down capacitor) close to the last Boundary Scan device in the scan chain (farthest away from the UUT TAP connector).

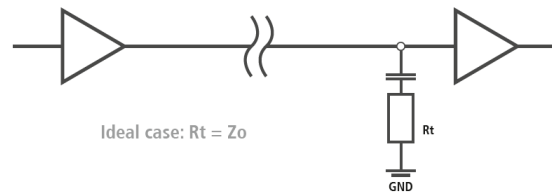


Figure 4-5 Termination example(s)

The ideal values for R_t and C_t shown in Figure 4-7 depend on the current UUT layout (voltage, signal length and layout...) and should be calculated as accurately as possible. Common values for R_t are 68 Ohm to 100 Ohm (R_t should match the impedance)

Ideal case: $R_t = Z_o$

Z_o of the transmission line, considering both the test bus cable and the trace on the UUT) and C_t is often in the range of up to 100 pF. Such a termination, including R_t and C_t , is beneficial only for the TCK signal and potentially for TMS. The purpose of C_t is the reduction of quiescent power dissipation in R_t .

The $/\text{TRST}$ signal should have a weak pull-down resistor (e.g. 10 kOhm) to ensure that Boundary Scan stays inactive during functional use of the UUT. Never tie $/\text{TRST}$ directly to GND as this would permanently disable Boundary Scan. The Boundary Scan test system drives the $/\text{TRST}$ signal high for the duration of Boundary Scan tests.

Even though required by IEEE 11491, not all Boundary Scan devices have an internal pull-up resistor at their TDI pin. To avoid or identify test problems later on, consider including a pull-up resistor (e.g. somewhere between 5 k Ω to 10 k Ω) on all TDO-TDI connections.

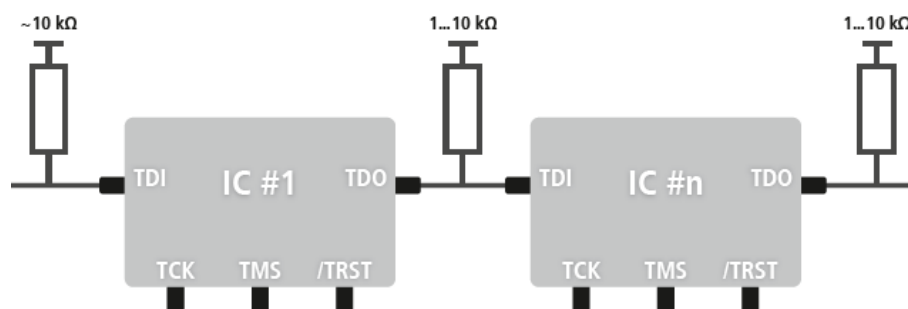


Figure 4-6 TDI-TDO connection example

Option: Use buffered test bus signals for more complex (more than 2 or 3 Boundary Scan devices) designs: The buffer (74'244, 74'125...) is supplied by on-board VCC! If it's not possible to use a buffer on-board try to buffer the signals externally (near to the board connector). In this case you can supply this buffer by the 3 V or 5 V from your GÖPEL electronic Boundary Scan controller.

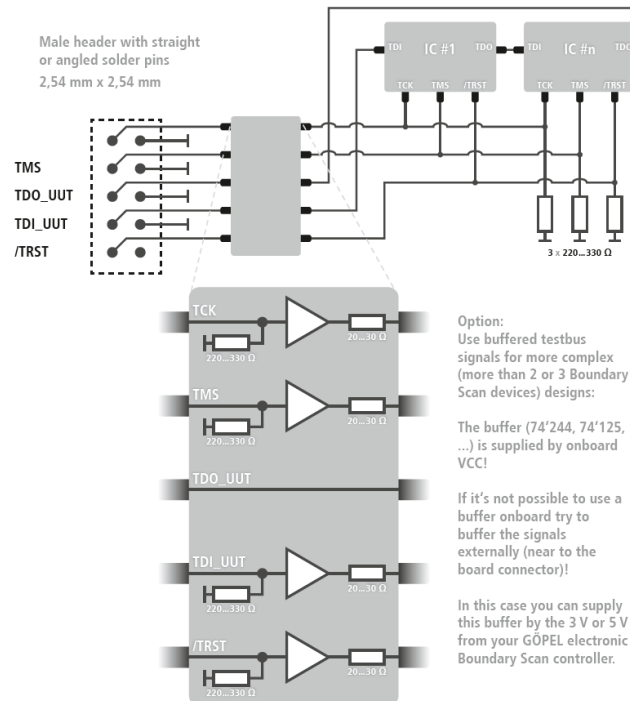


Figure 4-7 Scan chain design example

4.7 Maximum scan chain device number

If there are multiple devices in a scan chain, the Boundary Scan controller output driver must be able to drive all input pins (fan-in/-out). Scan chains featuring more than three to five devices (depends on the signal voltages and driver strengths) generally require a buffer on the UUT or close to it in order to drive the test bus signals. If not, the TAP synchronization may be ruined as shown in the picture below. Buffer devices must be non-inverting types.

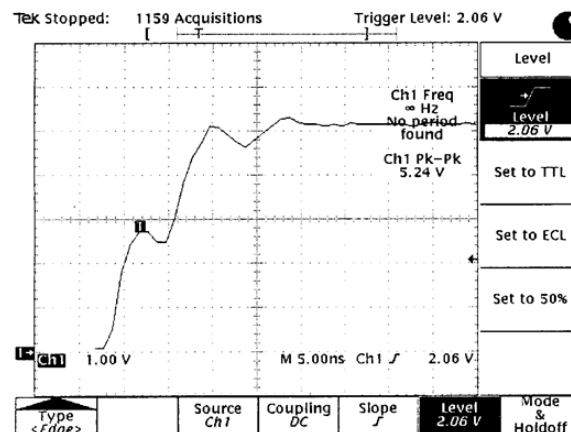


Figure 4-8 „Extra Clock“ at the rising TCK edge

4.8 Scan chain voltage levels

If the Boundary Scan devices use different I/O voltage levels, double check first that the lower voltage components are compliant to the higher voltages, if not, make use of level shifters to adjust the signal levels accordingly or keep the respective devices in separate scan chains. The decision of when to shift signal levels or when to buffer the test bus signals depends on conditions on the UUT (number of Boundary Scan ICs, I/O technologies, etc.)

4.9 Help test engineers in quickly locating scan chain errors

If TDO-TDI connections are buried in middle layers and the respective pins are hidden, add a test point in each of the signals. If possible, place the test point close to TDO. IEEE 11491 provides means to diagnose scan chain errors, e.g. a simple instruction shift verifies connectivity of TCK, TMS and TDO.

4.10 BScan ICs that power other BScan ICs & EXTEST

A typical design contains a device (FPGA/ μ C) that controls the power supply of the major part of the board which can include other Boundary Scan devices. Especially if the powering device is a Boundary Scan device as well some additional rules should be double checked for:

- If possible the powering device should be in a single scan path.
- If the powering device has to be located in the same scan chain with other Boundary Scan devices, the powering device must be first in the scan chain.

This is necessary as the powering device has to be configured and set to EXTEST before any other device can use the Boundary Scan resources.

4.11 Testing of assemblies of multiple boards

If the UUT is an assembly of multiple boards or modules, and there are UUT variants where some boards or modules may not be mounted, with Boundary Scan devices included on the various modules, it is important that any missing modules do not break the scan chain for the whole assembly. One way to handle this is illustrated in the figure below.

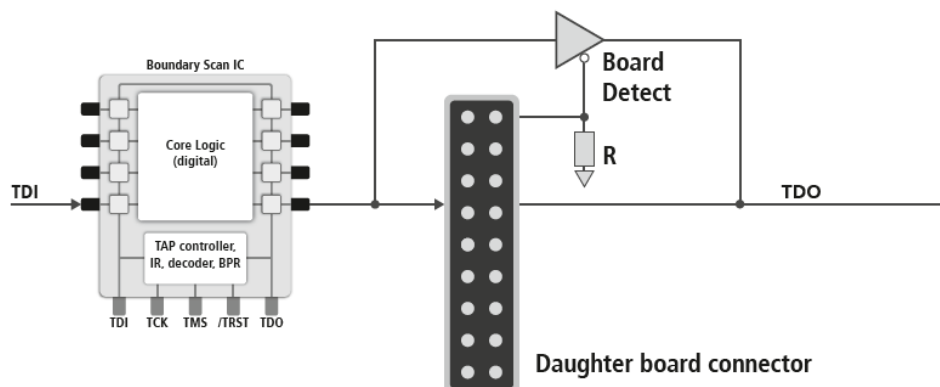


Figure 4-9 A missing daughter card

4.12 Allow to bypass ICs that are not mounted or not compliant

Some UUT variants may call for one or more Boundary Scan devices not to be mounted. In order to keep the scan chain intact in such cases, at least a bypass resistor connecting the respective TDI and TDO signals is needed. If Boundary Scan ICs not fully compliant to IEEE 11491 are included on a UUT, it is best to keep them in separate scan chains, so that they can be excluded or included in tests as appropriate and possible, without impacting other devices on the UUT.

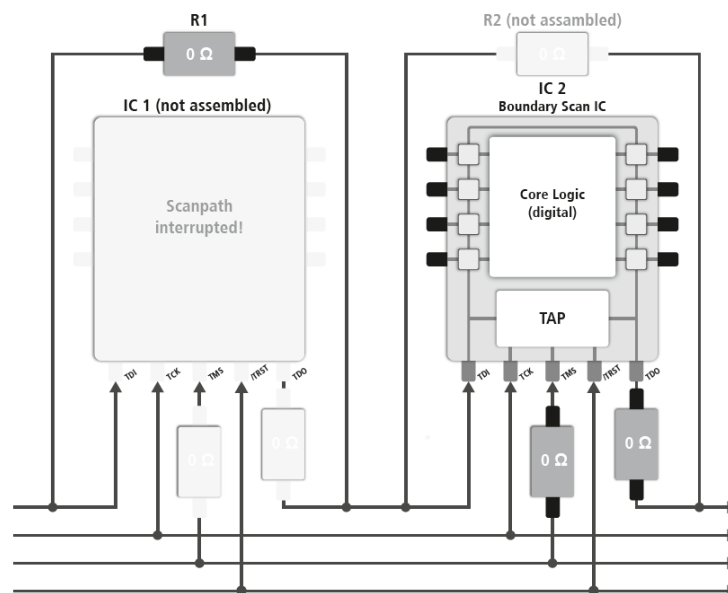


Figure 4-10 Bypass devices

4.13 When to place devices in a separate scan chain?

In general, we can say that devices that are compliant to 1149.1 can be used in the same scan chain and devices not compliant or featuring a different test and debug interface shall be placed in a separate scan chain as other Boundary Scan devices.

In some cases it may make sense to keep such devices in a separate scan chain, e.g. when other test/programming equipment is used that requires a single device within the scan chain, some more examples can be seen in following list:

- Boundary Scan devices with high TCKmax for Flash programming
- Boundary Scan devices that control UUT power
- μ Controllers, especially if used in debug mode (VarioTAP/JEDOS)
- μ Controllers not using the JTAG interface but another debug interface

By the way, using the right test equipment (GOEPEL SCANFLEX hardware), different interfaces can be handled in parallel within the same test sequence.

5. FPGA related DfT

5.1 FPGA configuration and Boundary Scan

FPGA devices generally do not allow for Boundary Scan testing while they are configured. This means the configuration of such FPGA devices needs to be inhibited or prevented for Boundary Scan testing. Typically there is an FPGA control signal that must be held to a certain level.

Alternatively, the FPGA configuration needs to be completed before Boundary Scan testing. Sometimes the FPGA configuration is required for proper I/O configurations (e.g. for communication with low voltage memory). In these cases configurations changes FPGA I/O behaviour and the BSDL file must be modified for proper operation. Fortunately the FPGA manufacturers provide software tools to create post-configuration BSDL files.

If not required by the functional design or extended test purpose the FPGA should be used un-configured because every I/O pin features bi-directional Boundary Scan resources, ideal for achieving a high test coverage.

5.2 FPGA/CPLD test designs

Consider test specific designs for FPGAs (especially if the functional design reduces the Boundary-Scan test coverage). Such designs could even provide test functions that cannot be obtained with pure Boundary Scan.

Even small CPLDs providing a TAP for in-system configuration only (but no Boundary Scan cells and no EXTEST instruction) can be reconfigured temporarily for test purposes, e.g. providing a simple pass through network or some test logic functions (e.g. IEEE 1581). After completing the Boundary Scan test such FPGAs and CPLDs can then be configured with their functional design.

5.3 FPGA Concatenation

Avoid concatenations of FPGA devices in a way that one FPGA needs to be configured (and drive a signal pin) before the next FPGA's scan chain can be enabled. This can lead to complex enable procedures and limited debug capabilities. Make sure you can enable all FPGA with test start and at the same time.

5.4 FPGA I/O signal behaviour

Some FPGA have pins that allow enabling/disabling implemented pull resistors. An example is Xilinx' Spartan 3A series; when HSWAP_EN is HIGH, then the internal pull-up resistor is enabled. This corresponds to the default BSDL description. Make sure that for testing you have this pin controlled in the correct way. Alternatively, handle it in your test setup appropriately.

6. Non-Boundary Scan Signals and Devices

6.1 Provide access to control signals

Important control signals include reset, watchdog, power control and clock signals and maybe some more.

Keep in mind that reset signals potentially can inhibit one or more Boundary Scan devices from functioning in test mode, resulting in scan chain errors.

In certain time intervals, watchdog devices expect a trigger, e.g. an “alive” signal generated by a controller device on the UUT. If such a trigger is not detected by the watchdog device, it typically generates a reset pulse or puts the UUT into a safe state. Intermittent scan chain errors may be the result. Provide means to disable the watchdog, if possible using a Boundary Scan pin.

Power control signals may enable or disable power for the whole board or just for parts of the UUT. If Boundary Scan devices are affected this will result in scan chain errors, if other components are affected, it may result in other failing test steps.

Clock signals should be accessible and controllable by Boundary Scan in order to control synchronous devices such as DRAM.

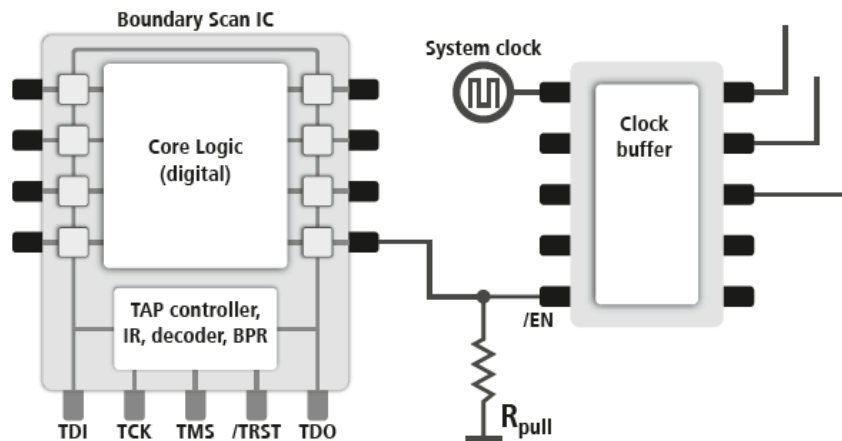


Figure 6-1 Example for clock control

6.2 Provide means to disable non-Boundary Scan ICs completely

In order to verify bus structures for shorts, it is important to deactivate outputs of non-Boundary Scan devices connected to those busses. Make sure that respective control lines can be controlled by Boundary Scan.

Bus contentions and collisions must be avoided by design anyway but in some cases test logic differs from functional logic. Therefore, try to be as flexible as possible.

6.3 Make logic clusters controllable, whenever possible

As many as possible cluster inputs and outputs need to be connected to Boundary Scan pins. Try to keep clusters as small as possible for good test diagnostics. In some cases it may be beneficial to program the cluster logic into a small PLD device supporting Boundary Scan testing itself.

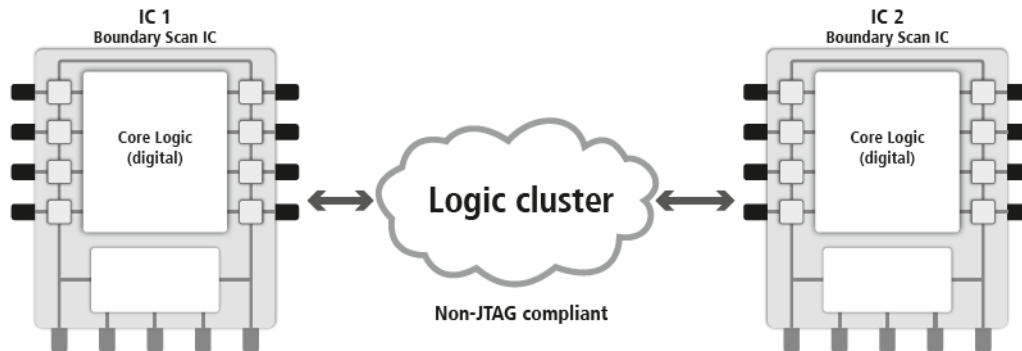


Figure 6-2 Logic cluster surrounded by Boundary Scan ICs

6.4 Spare pins – BScan vs. GPIO vs. analogue...

Take advantage of spare Boundary Scan pins, e.g. unused μ Controller pins or FPGA pins or I/Os from your tester electronics. Connect them to control lines (enable or direction pins) or other signals that were not testable without or control signals on other or even to signals inside logic clusters for. Even if this approach does not improve test coverage significantly, it may improve diagnostic messages in more details.

6.5 Access to all signals of memory ICs

Make sure that all signals on memory devices can be controlled by Boundary Scan pins. This includes clock input(s) on synchronous memory.

In general, buffer, latches, or logic devices may be available and can be handled in some cases automatically by the test equipment. However, mostly such "clever" ideas result in more manual test development effort. Direct Boundary Scan access to the memory device simplifies test development and provides best diagnostic messages possible.

Future memory devices may implement additional test features, such as defined in the IEEE 1581 standard. Again, as long as all pins can be controlled by Boundary Scan, a connectivity test may be implemented easily.

7. Programming

7.1 Access to programmable non-Boundary Scan ICs

Ensure that all control signals on programmable devices (such as serial EEPROM or Flash EEPROM) are accessible by Boundary Scan pins. The programming time for Boundary Scan based in-system programming primarily depends on two parameters: the length of the scan chain and the TCK frequency. Both parameters have an impact on the data throughput between Boundary Scan IC and EEPROM. For the shortest possible programming time with pure Boundary Scan access, the scan chain must be as short as possible and the TCK frequency must be as high as possible. If possible, void address latch configurations, as they complicate the access sequence to the EEPROM and increase programming time significantly.

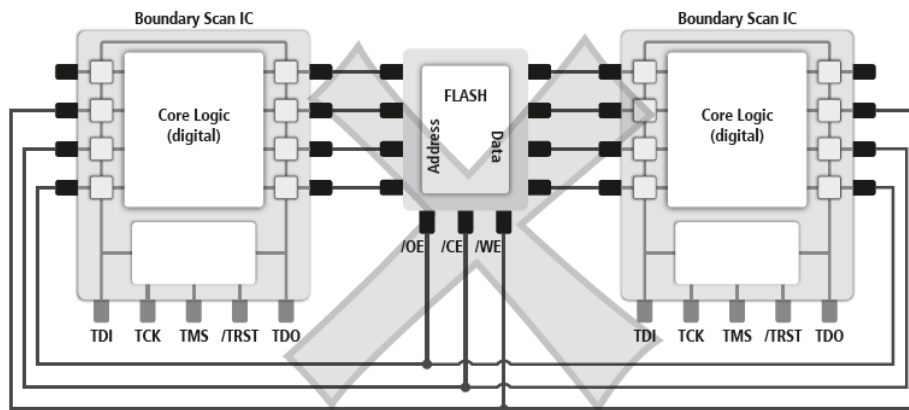


Figure 7-1 Avoid Flash memory access from multiple Boundary Scan devices

7.2 Enhance programming speed with direct access to Write Enable pin

The Flash programming time can be reduced significantly (about 35%) by utilizing direct access of the Flash device's write enable signal (through a test point or an edge connector pin) in order to control this signal with a parallel I/O from the test system rather than controlling this signal with Boundary Scan access. A precondition for this approach is the ability to disable any other Boundary Scan driver in this net.

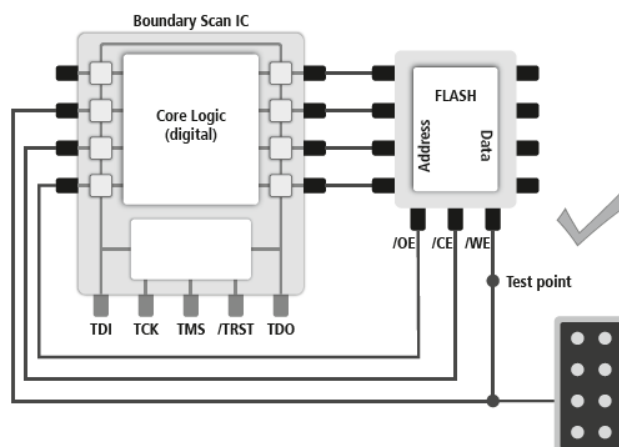


Figure 7-2 External control of /WE Signals

7.3 TCK frequency impact

If multiple Boundary Scan devices are available in a single scan chain, the device with the lowest TCKmax determines how fast pattern can be shifted. Make sure to use the highest possible TCK for programming flash memory. It may be beneficial to create a separate scan chain for the fast Boundary Scan device controlling the flash memory. In this case the programming performance can be optimised.

Another aspect to be considered is the signal transmission time, in particular the time between a falling edge on the TCK signal on the Boundary Scan controller and the corresponding signal change on TDO as detected by the Boundary Scan controller. Make sure your Boundary Scan hardware can compensate for these signal transmission delays so that they don't limit the maximum TCK frequency that can be applied to a particular UUT, especially if the test bus cable is quite long. SCANFLEX hardware equipment uses this TCK-TDO delay compensation (ADYCS®).

7.4 On-chip Flash memory

On-Chip Flash of μ Controllers can only be accessed by using the μ Controller itself which requires the use of the debug or emulation mode of the device. Hence, the pins for activating this mode have to be controllable. As these pins usually work in conjunction with the power on reset (/PORST) of the μ Controller, this signal has to be controllable as well. Furthermore, be aware of possible impact of watchdog timer mechanisms. Such should therefore be deactivated.

8. Extending Boundary Scan Test Coverage

8.1 Utilizing Boundary Scan throughout the product life cycle

For quality management system (QMS) purposes, it is useful to reserve a few bytes of Flash memory for board specific data, such as PCBA type and version, manufacturing date, serial number, rework/repair history, and other important information. The respective Flash device should be accessible via Boundary Scan for in-system programming purposes. Boundary Scan, in general, is applicable throughout the entire product life cycle. Once various PCBA and modules are assembled, Boundary Scan can provide an access mechanism for system level connectivity test, in-system programming and reconfiguration, power-on self-tests, and more. For PCBAs that are assembled for the first time or are returned from the field for repair, Boundary Scan can be an important and powerful debug and diagnostic tool.

8.2 Utilize test modes in non-Boundary Scan ICs (e.g. NAND-Tree)

Some non-Boundary Scan devices offer special test features that often can be beneficial for board-level manufacturing tests and for diagnostics. Don't inhibit the use of such test modes by permanently disabling them. Boundary Scan access to complex non-Boundary Scan devices can be used to provide at least some basic test capabilities, such as reading a device ID or some register values. If such non-Boundary Scan devices include a test mode, such as a NAND-Tree or NOR-Tree, this device can actually be included in Boundary Scan controlled connectivity tests, allowing the detection and diagnosis of open pins and shorted signals (e.g. due to solder bridges) that otherwise may not be easily diagnosed.

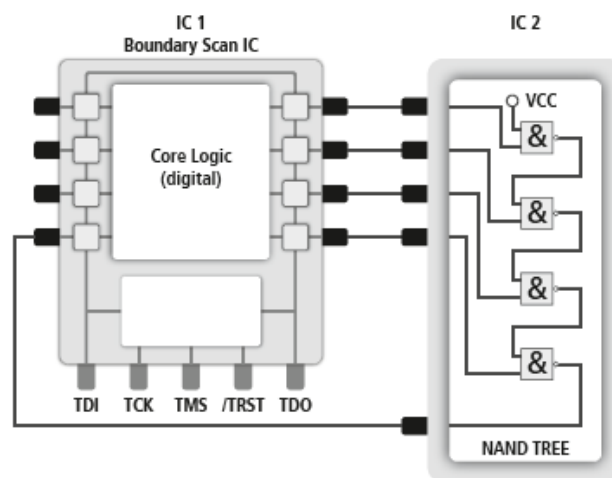


Figure 8-1 NAND-Tree test device

8.3 Test of analogue circuitry

Connect the digital side of analogue/digital converters (ADC) and digital/ analogue converters (DAC) with Boundary Scan pins. This way analogue circuitry can be involved indirectly into Boundary Scan tests.

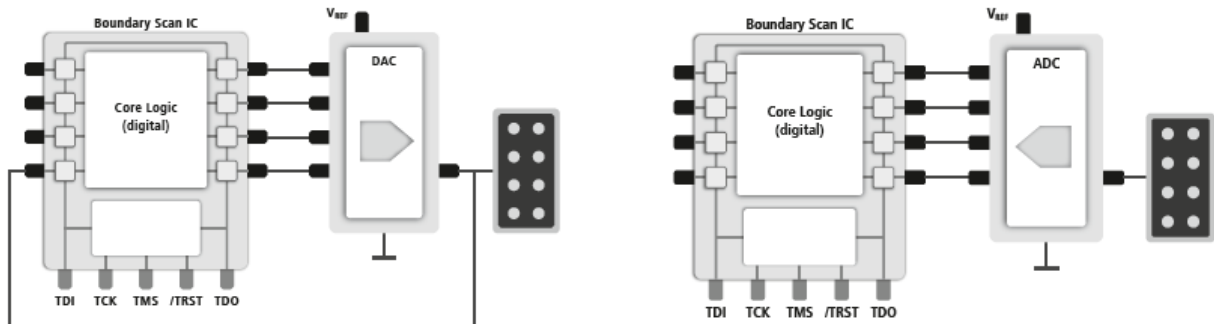


Figure 8-2 Analogue testing using DAC and ADC

IEEE 1149.4 defines test resources for analogue and mixed-signal Boundary Scan. However, the industry did not adopt the possibilities so that off-the-shelf components are nearly not to find, you may think about including such capabilities in your own mixed-signal ASIC.

Another option is the usage of the CION-LX with its analogue capabilities. This requires that the signals can be connected to the CION-LX which would usually be done via test points and special I/O modules. This opens the option for verifying frequency depending devices or circuits, e.g. LC resonators. The CIO-LX main features are following:

- Extended IEEE 1149.1 instruction set
- IEEE 1149.6 capabilities on ports DIFF, A-D and HC
- IEEE 1149.8.1 support
- TCK speed of 100Mhz
- 0,9V-3,6V (24mA) operating voltage at A-D, each group can have different voltage
- 0,9V-2,0V (40mA) operating voltage at HC
- LVDS or CML at DIFF port
- Frequency measurement on all user pins
- Cycle duration measurement
- Toggle recognition
- ADC (12bit SAR-ADC) on all single ended user pins (1k RAM)
- DAC (10bit) on all single ended user pins (1k RAM)
- Configurable pin properties
- Unstress feature
- Hot swap
- Tristate after Power-On

8.4 Test of optical components

Optical components, such as LEDs or LCD displays, for example, can be included in Boundary Scan tests by having an operator observing their proper function. Even a camera can be used. However, such components must be controlled directly or indirectly by a Boundary Scan device.

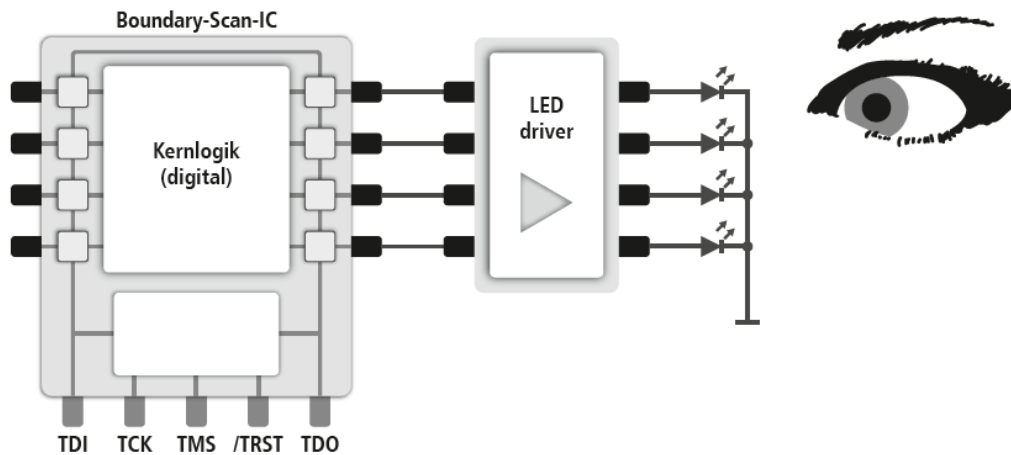


Figure 8-3 LED test

8.5 Test of switches

Switches (including push buttons, flip-flops, relays etc.) can be tested as part of Boundary Scan test. Such cluster tests are either written by the operator manually or are based on a device model. For example, a Boundary Scan IC can capture the pin connected to the switch several times, expecting a signal change within a certain time frame, the switch, of course, must be actuated by the operator or some other means.

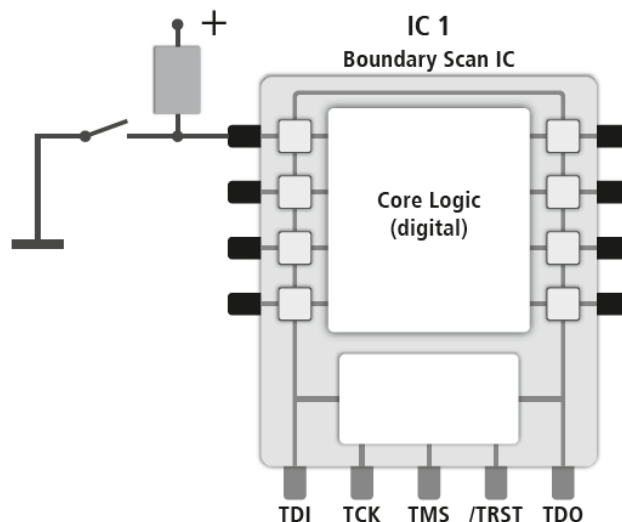


Figure 8-4 Switch test

8.6 Clock signal verification

Clock signals can be tested with Boundary Scan in regards to whether or not a clock pin is actively driving. Boundary Scan can detect if a clock signal is toggling, but it cannot measure the clock frequency. A Boundary Scan IC reads the pin connected to the clock signal a number of times, expecting a number of signal changes (e.g. detecting a minimum of X logic high and X logic low states out of Y signal captures) within a certain number of attempts. If the clock signal does not change state, a respective stuck-at0/1 error message is provided.

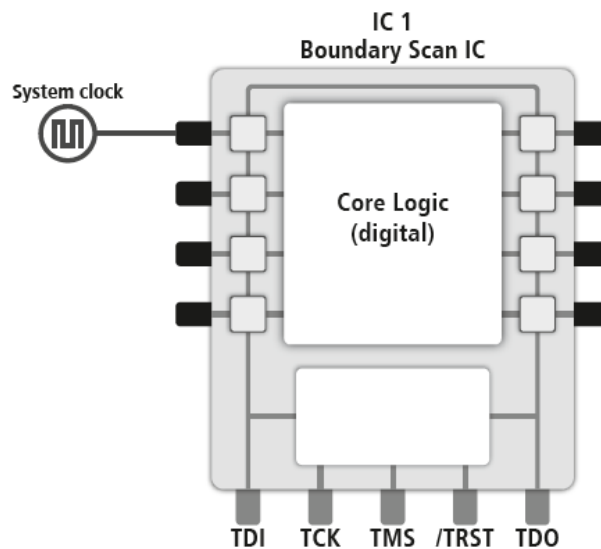


Figure 8-5 Clock signal verification

8.7 Clock signal measurement

Depending on the product, clock signals may be measured directly in today's board setups, in some cases there is not even an need for a test point or edge connector. Using Embedded Test, clock values can be measured in several ways.

If connected to an FPGA, use the appropriate ChipVORX[®] instrument to directly measure the clock value at the FPGA I/O pin (Xilinx) or dedicated clock input (Altera).

Alternatively, connect the clock signal to a CION-LX pin. However, that requires a test point or connector pin to be available at the clock net. Alternatively, connect a component pad using a spring probe.

Currently, there is no way to automatically measure clock values with a μ Controller if the clock signal is used for that component.

8.8 Improving coverage using I/O tests

The UUTs test coverage can be extended by connecting peripheral connectors. There are different ways; the simplest one is to use simple shortcut cables. Other setups use loopback cables or tester I/O channels, including them in the connectivity test. While loop-backs are an inexpensive and simple solution, the best test coverage and diagnostic support can be obtained with I/O modules.

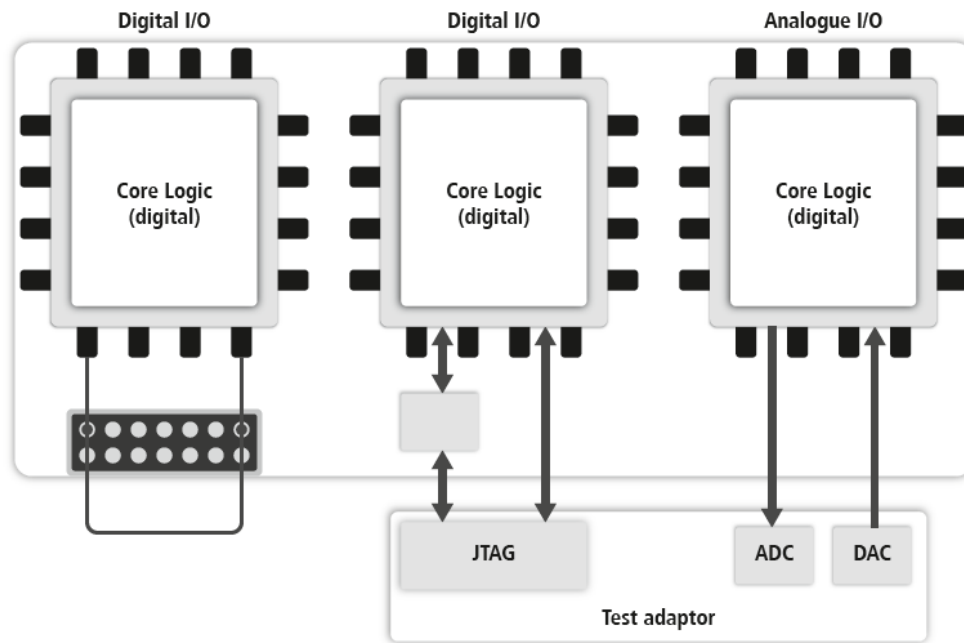


Figure 8-6 Including peripheral connectors in connectivity tests

8.9 Improving coverage using at-speed tests (VarioTAP® and VarioCORE®)

Boundary Scan is a quasi-static test methodology which may not be able to detect dynamic defects. Mixed-signal circuitry and widely used high-speed serial interfaces, for example, cannot be tested with the standard Boundary Scan connectivity test. Such parts must be covered by using extended test technologies such as VarioTAP® or VarioCORE® or JEDOS.

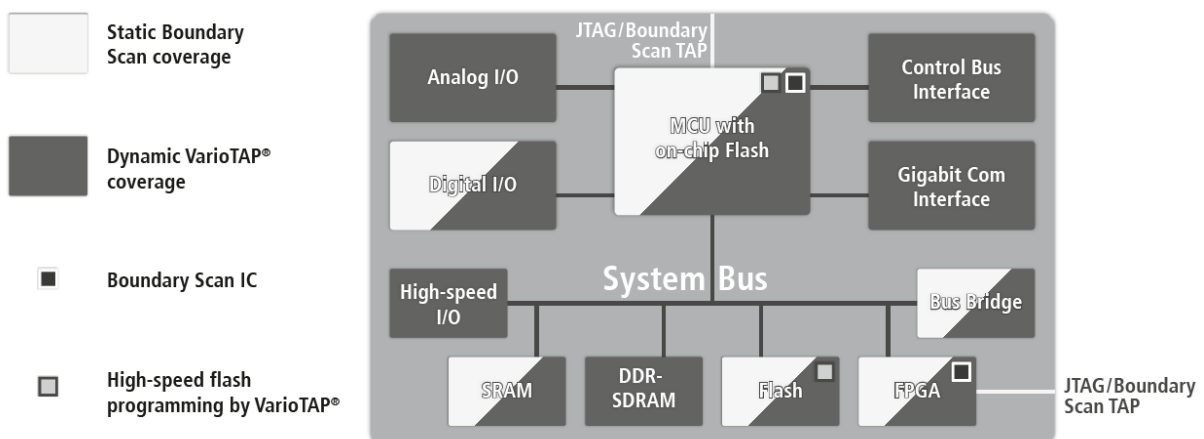


Figure 8-7 VarioTAP® on-chip programming, on-board programming, emulation test, and interface test

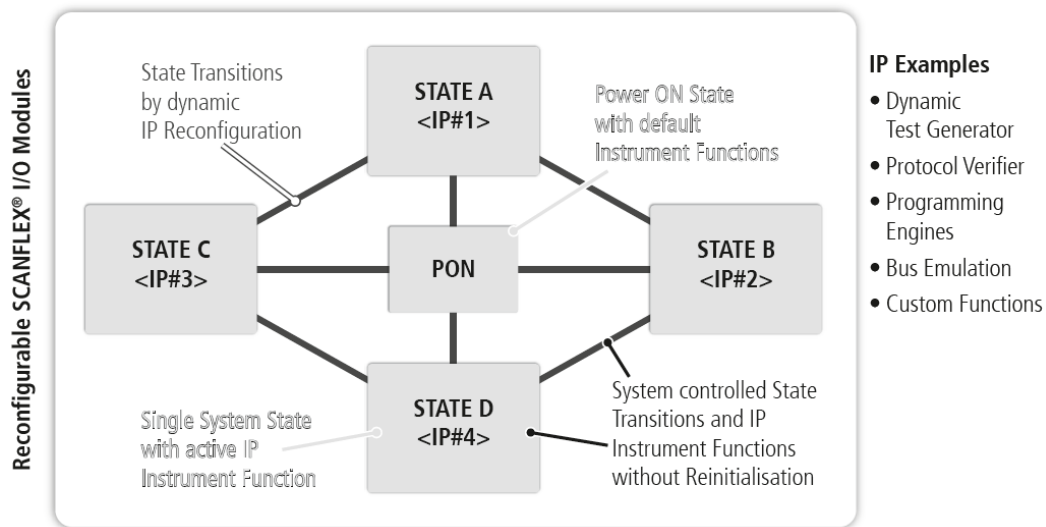


Figure 8-8 Reconfigurable VarioCORE® I/O module

8.10 Improving coverage by combining various test methodologies

Depending on the technology and complexity of the UUT and the test philosophy in place, Boundary Scan can be combined with other test methodologies. Such combinations can eliminate disadvantages of the individual test methodology.

In case of In-Circuit Testers, Boundary Scan can reduce the number of nodes that need to be accessed with nail probes, simplifying the bed-of-nail fixture and reducing its cost. Try to place all test points (test pads) on one side of the PCB (typically the bottom side).

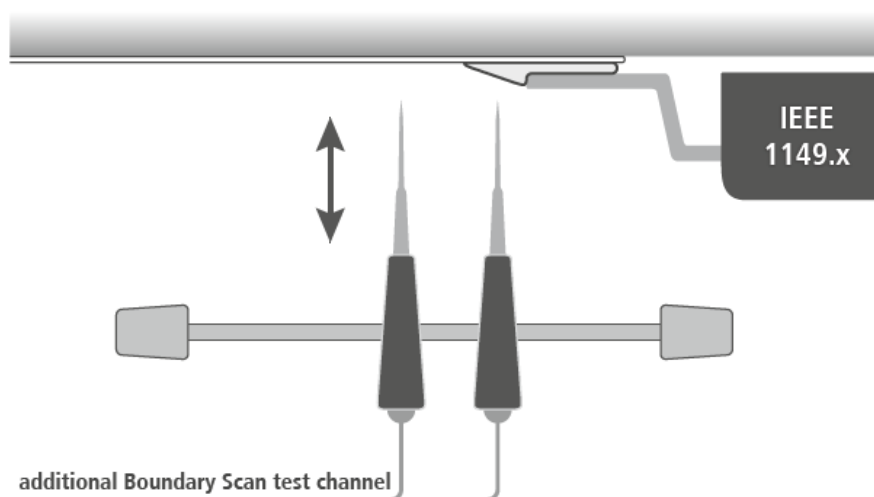


Figure 8-9 Combining Boundary Scan and In-Circuit Test

For flying probe testers the test time can be reduced by eliminating test steps for parts of the UUT that are already tested by Boundary Scan (such as tests for opens and shorts on digital circuit parts). Place test bus connector and power supply connections on the opposite side (from test points) of the board. This way the test bus cable and power cable are kept outside of the probing area, allowing the flying probes to move around with no further restrictions.

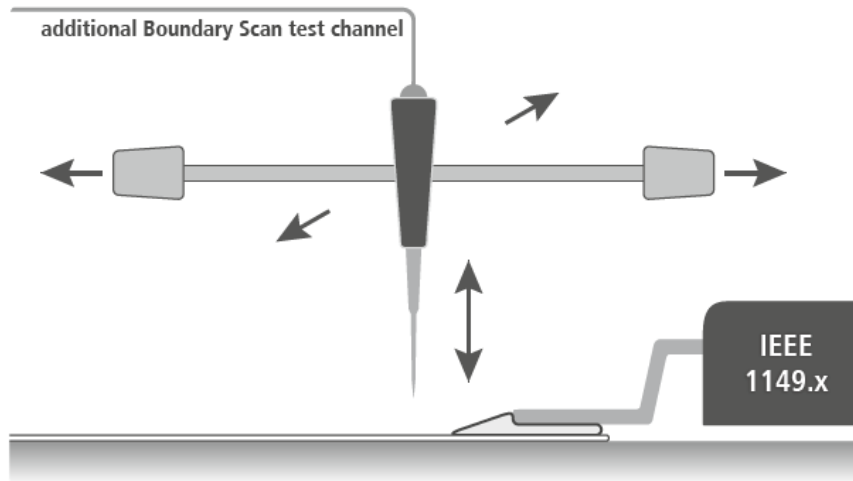


Figure 8-10 Combining Boundary Scan and Flying-Probe Test

Functional testers can benefit from the diagnostics provided by Boundary Scan, simplifying fault isolation and trouble-shooting at the functional test stage.

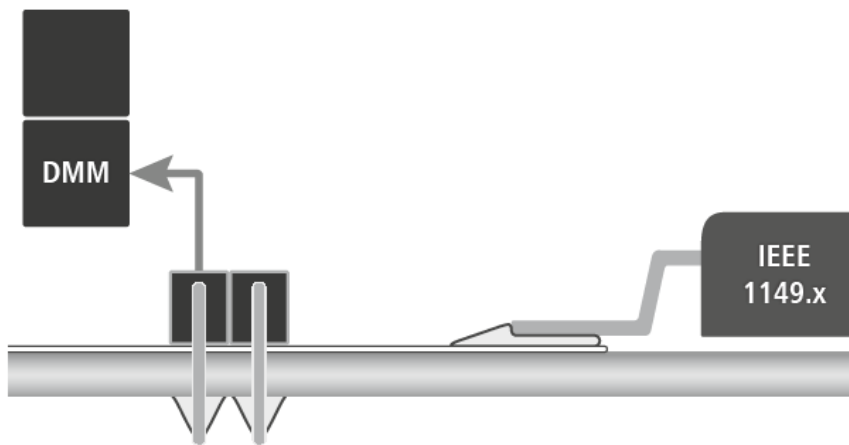


Figure 8-11 Combining Boundary Scan and Functional Test

Automated optical inspection can be used to check for presence, orientation, and alignment of components prior to electrical tests, which can enhance the accuracy of diagnostics. Furthermore, combining AOI and Boundary Scan allows the automated and more thorough verification of optical components (such as LEDs and LCD displays).

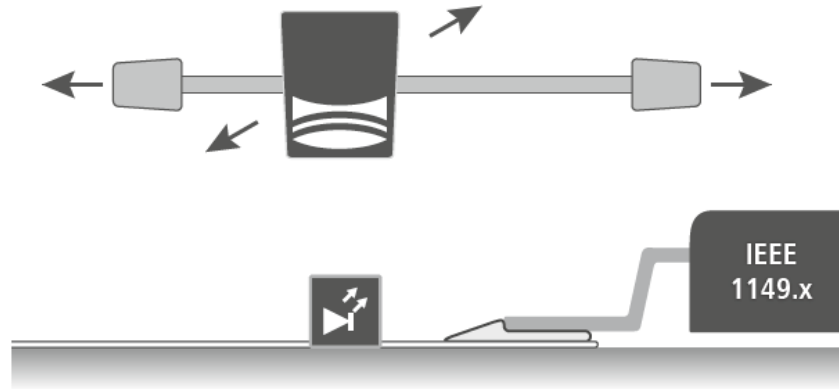


Figure 8-12 Combining Boundary Scan and Optical Inspection

Use test coverage reports to determine where test points for physical probe access are not needed, either from your Boundary Scan tool or from independent solutions like Tools like Aster's TestWay (www.aster.fr).

8.11 System test – Scan Router ICs

Embedded board test access can be very beneficial for system level test applications, where a PCBA is part of a larger assembly (e.g. multiple boards/modules plugged into a backplane). In such applications, the scan chain infrastructure needs to allow the test of individual system modules as well as the test between system modules.

Missing modules must not inhibit the Embedded Board Test of other modules within the system. These requirements can be satisfied with so called scan-router devices, available from various device vendors, including Firecron, Lattice Semiconductors, Telefunken Semiconductors, and Texas Instruments. Such scan router ICs are addressable and link a primary scan chain to one or more secondary scan chains.

Some of the manufacturers provide so called scan controller devices that allow taking over the scan chain control.

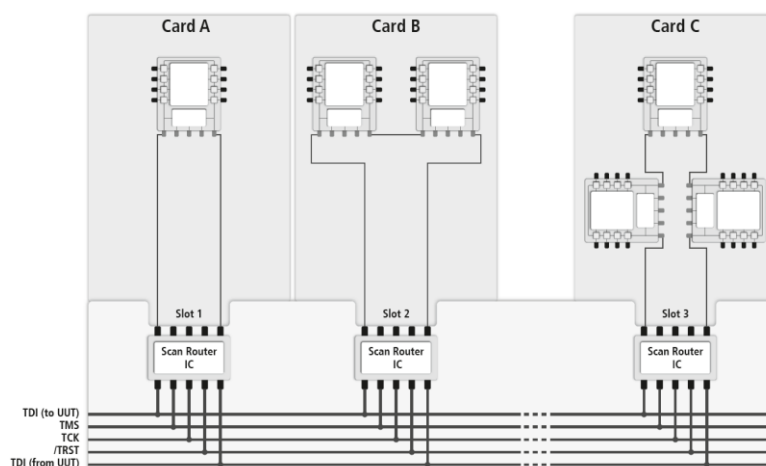


Figure 8-13 Utilization of Scan Router devices

9. Select an Embedded Board Test System

9.1 Software

Embedded Board Test software needs to understand the whole UUT, including intelligent components, Boundary Scan devices, non-Boundary Scan circuitry, logic clusters, interfaces as well as mixed signal areas. There are a few ***Intelligent Reasons*** that you should investigate for, depending on your requirements they can be of great importance:

Intelligent Tools

- data import, ATPG, viewers, TCR analyzers, diagnostic visualization, reports

Make sure the software provides sufficient flexibility and allows adjustments to the test pattern generation, in particular to avoid "Ground-Bounce" effects.

Intelligent Data Base

- UUT information referenced in one system

Such a data base is used by all tools and assures consistency as well as test program reusability.

Intelligent Programming Language and Device Library

- low- and High level commands allow for test, programming and debug

Test programs written in a high-level programming or scripting language provide the necessary flexibility to work around non-compliant device behaviour and to handle Boundary Scan test applications beyond the basic connectivity tests.

The Boundary Scan test system of your choice should already include a sizable library of Boundary Scan enabled devices as well as memory, logic and other ones.

Intelligent Protection

- model based functions and descriptions, safe values, compliance conditions

By analysing the surrounding circuitry and how it interacts with the Boundary Scan devices and Microcontrollers, ATPG tools can generate safe test pattern that do not harm the UUT.

Intelligent Fault Diagnostics

- precise and systematic fault isolation; layout viewer; schematic viewer, debugger

It may be helpful to edit the generated test vectors or to debug test programs. The Boundary Scan software should offer means to execute test programs in a step-mode. The user should be able to get access to and full control over all test resources (BScan cells, I/O pins, interfaces, registers, nets...).

Intelligent SW Interfaces

- user profiles, users environment, consistent GUI, project archive transfer, 3rd party ATE support, future proof

Boundary Scan software should allow integrations in third-party test equipment (such as in-circuit testers or flying probe testers) in order to support extended Boundary Scan applications. Such integrations can be realized at different levels, the most advanced of which provides the Boundary Scan software access to tester resources of the third-party equipment to extend the Boundary Scan test coverage.

And last but not least: The Embedded Test System should be able to handle non-compliant situations. Very often, the test engineer faces Boundary Scan components that are not compliant to the IEEE 1149.1 standard. Being able to handle such situations anyway, makes a big difference in test coverage!

9.2 Hardware

If the Embedded Board Test controller provides multiple TAPs, make sure those TAPs are truly independent and do not get chained together inside the controller.

Furthermore, check for programmability of TAP voltage parameters, terminations and TCK clock speed, and make sure the hardware provides parallel I/Os for additional test access.

Embedded Board Test hardware should be able to compensate for signal delays, especially between TCK and TDO. If not, the test bus cable length will limit the maximum TCK speed which may be influencing the test performance.

Special hardware modules can be very important upgrades to a basic Boundary Scan test system, e.g. for extending the test coverage (e. g. mixed signal I/O modules). There should be hardware items for integration into 3rd party ATE test equipment and for debug and trouble-shooting assistance.

Some Embedded Board Test hardware can be upgraded without any physical hardware exchanges; beneficial if one wants to start with a basic set of hardware functions and add performance and capabilities at some later time, without losing the initial investment.

9.3 Licensing

Flexible, granular licensing schemes enable the customization of system configurations.

Consider network licensing which allows software to be shared between multiple users or even departments, eventually even worldwide if you work in a respective company. Such floating licenses can offer a better utilization of a test system than node-locked licenses.

10. Glossary

ADC	analogue to Digital Converter
ADYCS	Active DelaY CompensAtion
AFPG	Automatic Flash Program Generation
ASIC	Application Specific Integrated Circuit
ATE	Automatic Test Equipment
ATPG	Automatic Test Program Generation
BScan	Boundary Scan - Test methodology defined in IEEE 1149.1
BIST	Built-In Self-Test
BSDL	Boundary Scan Description Language
CAD	Computer-Aided Design
CASCON	Computer Aided SCan based Observation and Node control – software environment for developing Embedded Board Test applications
CEI	Chip Embedded Instruments
CION	Complex Input Output Network – test IC providing digital test channels
CION-LX	2 nd generation CION providing also mixed signal test features
CPLD	Complex Programmable Logic Device
DAC	Digital to analogue Converter
DDR-RAM	Double Data Rate - RAM
DfT	Design for Test
DUT	Device Under Test
EEPROM	Electrically Erasable and Programmable Read-Only Memory Flash Type of EEPROM – name derived from method used to erase the memory
ESA	Embedded System Access
EBT	Embedded Board Test – BScan, PET, CEI, JEDOS...
FAP	FPGA Assisted Programming
FAT	FPGA Assisted Test
FCT	Functional Test
FLASH	Non-volatile memory for memory intense applications (cheap but slow)
FPT	Flying Probe Test
FPGA	Field Programmable Gate Array
Ground Bounce	Possible board test problem; many simultaneously switching pins may raise the IC-internal GND level which resets the IC immediately
ICT	In-Circuit Test
IEEE 1149.1	IEEE Standard Test Access Port and Boundary-Scan Architecture – digital board test – digital connectivity test
IEEE 1149.4	IEEE Standard for a Mixed-Signal Test bus - analogue board test
IEEE 1149.6	IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks - test of differential and AC coupled signals
IEEE 1149.7	IEEE Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture - cJTAG or Compact JTAG
IEEE 1149.8.1	IEEE Standard for Boundary-Scan-Based Stimulus of Interconnections to Passive and/or Active Components – Selective-Toggle or A-Toggle
IEEE 1500	IEEE Standard Testability Method for Embedded Core-based Integrated Circuits – Multi core silicon, IC-internal test & scan structures

IEEE 1581	IEEE Standard for Static Component Interconnection Test Protocol and Architecture - test features within non-BScan ICs, such as memories
IEEE 1687	IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device - iJTAG or Internal JTAG
IEEE P1838	IEEE Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits - real 3D stacking
IP	Intellectual Property
JTAG	Joint Test Action Group – Initiators of IEEE Std. 1149.1
JEDOS	Jtag-Embedded Diagnostic Operating System – Embedded ATE
LAN/WAN/GAN	Local/Wide/Global Area Network – local/wide/global Ethernet
MCM	Multi-Chip Module – multiple dice in a single package, pure horizontal approach, no 3D stacking
PCB/PCBA	Printed Circuit Board/ Printed Circuit Board Assembly
PET	Processor Emulation Test
PLD	Programmable Logic Devices
QMS	Quality Management System
RAM	Random Access Memory – SRAM, DRAM, others (fast but expensive)
SiP	System in Package - multiple dice in a single package, including vertical stacking (3D)
SoC	System on a Chip - single chip integrating multiple functionalities (µP, RAM, logic, interfaces...)
TAP	Test Access Port – IEEE 1149.1 test bus signals
TCK	Test Clock
TDI	Test Data In
TDO	Test Data Out
TMS	Test Mode Select
/TRST	Test ReSeT
UUT	Unit Under Test

11. Figures

<i>Figure 1-1 Troubleshooting and repair cost vs. test technology cost</i>	5
<i>Figure 1-2 Device complexity & signal transmission evolution</i>	5
<i>Figure 1-3 Board Test Paradigm Change</i>	6
<i>Figure 1-4 Embedded Board Test</i>	7
<i>Figure 3-1 PCB with multiple BScan devices, memory types and interfaces</i>	10
<i>Figure 3-2 IEEE 1532 Scenario</i>	12
<i>Figure 4-1 JTAG port pinout</i>	13
<i>Figure 4-2 Daisy chain scenario and TAP pin connections</i>	13
<i>Figure 4-3 Control pins</i>	14
<i>Figure 4-4 Signal overshoots and "extra clock cycles" due to reflections</i>	14
<i>Figure 4-5 Termination example(s)</i>	15
<i>Figure 4-6 TDI-TDO connection example</i>	15
<i>Figure 4-7 Scan chain design example</i>	16
<i>Figure 4-8 „Extra Clock" at the rising TCK edge</i>	16
<i>Figure 4-9 A missing daughter card</i>	17
<i>Figure 4-10 Bypass devices</i>	18
<i>Figure 5-1 Example for clock control</i>	20
<i>Figure 5-2 Logic cluster surrounded by Boundary Scan ICs</i>	21
<i>Figure 6-1 Avoid Flash memory access from multiple Boundary Scan devices</i>	22
<i>Figure 6-2 External control of /WE Signals</i>	22
<i>Figure 7-1 NAND-Tree test device</i>	24
<i>Figure 7-2 Analogue testing using DAC and ADC</i>	25
<i>Figure 7-3 LED test</i>	26
<i>Figure 7-4 Switch test</i>	26
<i>Figure 7-5 Clock signal verification</i>	27
<i>Figure 7-6 Including peripheral connectors in connectivity tests</i>	28

<i>Figure 7-7 VarioTAP[®] on-chip programming, on-board programming, emulation test, and interface test</i>	28
<i>Figure 7-8 Reconfigurable VarioCORE[®] I/O module</i>	29
<i>Figure 7-9 Combining Boundary Scan and In-Circuit Test</i>	29
<i>Figure 7-10 Combining Boundary Scan and Flying-Probe Test</i>	30
<i>Figure 7-11 Combining Boundary Scan and Functional Test</i>	30
<i>Figure 7-12 Combining Boundary Scan and Optical Inspection</i>	31
<i>Figure 7-13 Utilization of Scan Router devices</i>	31

[illegible]

GOPEL
electronic
Enjoy Testing

GOEPEL electronic GmbH

sales@goepel.com
www.goepel.com

 sales@goepel.in

Design for Testability/E/01-2017