BOUNDARY-SCAN: AN INTRODUCTION

by James Stanbridge, Sales Manager of JTAG Technologies

Once considered to be something of a black art, and solely an aid to manufacturing, boundary-scan is ‘coming of age’ - thanks largely to the emergence of easy-to-use and highly automated tools for developing boundary-scan tests.

What’s more, boundary-scan is increasingly used as a means of in-system programming (ISP) of flash memory and programmable logic devices (PLDs); and also as a debug interface for microprocessors, Digital Signal Processors (DSPs) and System on Chip (SoC) designs.

Encouragingly, boundary-scan is now a standard feature on most micros (-processors and –controllers), 16-bit and above, for example, Freescale (formerly Motorola) QUICC, ColdFire and Dragonball series, Intel’s Pentium and Xscale families, Renesas (formerly Hitachi) SuperH, TI’s TMS 320 series DSPs. You may also find it in modern variants of some 8-bit systems. But what is boundary-scan?

Also known as JTAG, after the Joint Test Action Group that developed it, the technology has been around (conceptually) for more than 20 years.

It was officially approved (as IEEE Std 1149.1) in 1990 so, as electronics ‘standards’ go, it is safe to call it a mature technology.

Even so, there remains a considerable thirst for knowledge of the fundamental ‘nuts and bolts’ of JTAG and how it can assist in not only the post-manufacture testing of circuit boards, for which it was initially developed, but also a variety of board and system development applications.

Component issues

Where a boundary-scan version of a digital component exists it will typically have four or five additional pins/pads, which together constitute the Test Access Port (TAP). Excluding the TAP and power/ground pins/pads, the device’s other digital pins/pads will usually each have a boundary-scan cell or cells – which are ‘transparent’ during normal operation. It will also have three registers:

- Boundary-scan Register (BSR);
- Instruction Register; and
- ID Register.

These registers, and the chain of boundary-scan cells, are all accessible via the TAP.

Data comes into the device on TDI (Test Data In) and exits on TDO (Test Data Out). Two control signals, TCK (Test Clock) and TMS (Test Mode Select), control the state machine of the TAP controller: which synthesises the switch and mode signals that select internal registers via multiplexers and latches.

As mentioned, a fifth pin/pad may be present on the device. TRST offers an asynchronous reset capability to the TAP controller. However, device reset can also be achieved using the ‘soft method’ of holding TMS high for five clock cycles.

The above registers are mandatory for a device to be IEEE Std. 1149.1 compliant.

Instructions
A boundary-scan device must also accept three instructions.

- EXTEST which allows external testing via the BSR;
- SAMPLE/PRELOAD which can load or sample BSR cells without interference; and
- BYPASS which selects bypass register and puts the device into normal operation.

Three optional instructions were also included in the original standard.

- INTEST which allows internal [core logic] testing via the boundary-scan register;
- IDCODE which scans out the ID register's contents (a 32-bit unique device code); and
- RUNBIST which invokes device level BIST (e.g. using internal scan methods);

And two more optional instructions in the 1992 IEEE 1149.1a addendum:

- HIGHZ which selects bypass register - device tri-states all 3-state outputs; and
- CLAMP which selects bypass register - device outputs set to defined state.

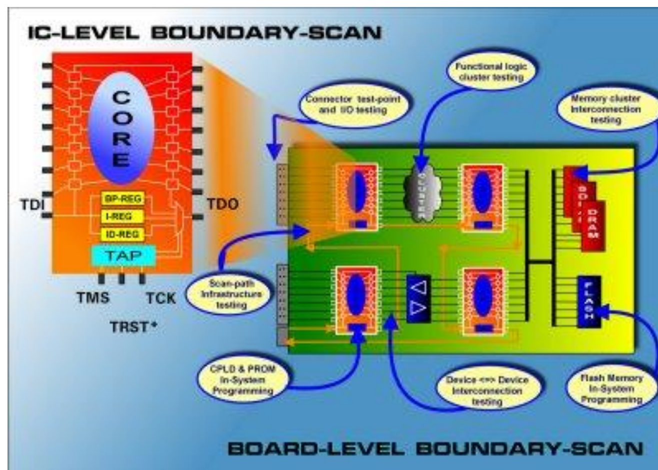
For every IEEE1149.1-compliant device, the manufacturer is obliged to produce a model for the boundary-scan logic known as the Boundary-Scan Description Language (BSDL) file.

Close to VHDL in syntax, the BSDL includes a port (pin) description, pin name versus number mapping, supported instruction set and a BSR listing showing the scan cell to pin relationship. These can typically be downloaded from manufacturers' web-sites.

Boundary-scan in action
For test purposes the above instruction set has proved more than sufficient over the years, allowing users to:

- Check scan-path infrastructure and integrity. This confirms the presence and location of the boundary-scan devices. The check is 'non-disruptive' and can be done whilst the circuit board is functioning normally.
- Scan-pin to scan-pin interconnect test. This is used as a post-manufacture assembly test and verifies the electrical connectivity of the nets between components accessible via boundary-scan (where 'accessible' does not necessarily mean that boundary-scan components can only test other boundary-scan components– see later).
- Memory cluster tests. Memory (read and write) test patterns can be applied to most memory devices and technologies.
- Apply 'functional' test vectors (patterns) to non-scan logic elements.

In addition, thoughtful use of test access provided by boundary-scan has enabled users to create flash ISP applications with relative ease.



The above diagram illustrates how boundary-scan might be routed around a circuit board, and zooms in on a typical device to show how boundary-scan is implemented within a component.

Programming

Historically, programmable logic vendors - such as Altera, Lattice, Xilinx and Cypress - were the early adopters of boundary-scan; but not always in order to use its test features. The attraction of boundary-scan to these silicon vendors was as a programmer interface that could allow register accesses in order to program the EEPROM and/or Flash memory which stored the macrocell and gate 'fuse maps'.

Of course, in the early days, each vendor devised its own schemes and instructions to implement this, adding so called PRIVATE [JTAG/boundary-scan] INSTRUCTIONS to those listed above. So, whilst the interface itself was a standard, the data formats and instruction codes were not. Indeed, several attempts were made by a few vendors to introduce 'industry standards' for the configuration codes and procedures (e.g. JAM, XSVF, etc.).

By the late 1990s however a common PLD programming standard working group was established, which took input from test system developers such as JTAG Technologies as well as the aforementioned CPLD vendors. By 2001, the new standard IEEE Std 1532 for In-System device Configuration (ISC) was approved.

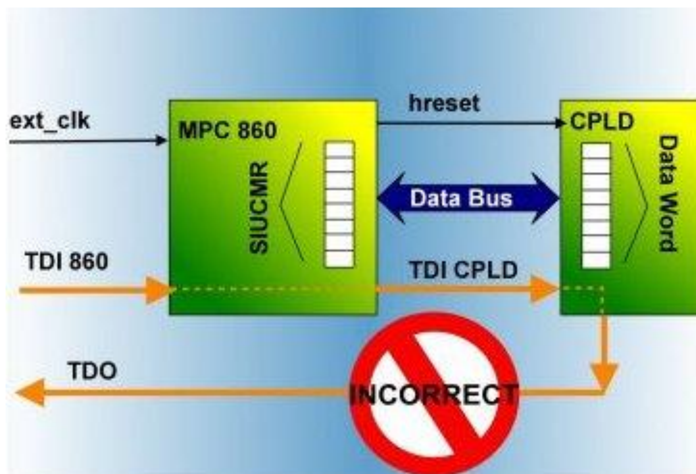
This standard documents both the approved data-format and the extensions required for compliance to a new BSDL model format. A new minimum set of ISC registers is defined, plus the minimum length for the JTAG instruction register (IR) is now set to 4 bits - in order to decode the now 10 mandatory instructions for this standard.

Micros

Just as the PLD vendors used (and built on) the capabilities of boundary-scan for *their* needs, so too did the microprocessor and DSP vendors; 'fitting' the basic boundary-scan functionality and thus allowing structural board tests to be performed prior to board and system-level functional tests.

However, whereas the PLD vendors were focussed on adding functions like ISP, the micros and DSP vendors added emulator and debug functions - accessible via the standard JTAG/Boundary-scan port.

In some instances (e.g. Motorola/Freescale PowerQUICC series) JTAG's TDI and TDO pins are actually multiplexed with the pins DSDI and DSDO (used for an embedded debug port and which allow a device to operate in 'Background Debug Mode').



In theory this is a smart way to save on pin-count but in practice it requires some design effort to create a system which can switch easily between the two modes due to the convoluted method of reading a field data word into the set-up register (known as the SIUCMR) some 512 clock cycles after an asserted hard reset

With the set-up above a problem will result if the CPLD is assembled on the board in a blank state. With no config' data word programmed, the MPC 860 cannot be switched into boundary-scan mode and the device becomes inaccessible. Only a fresh layout with a separate TAP for the CPLD could solve this problem

Coming of age

Initially JTAG/boundary-scan could only be found on larger, 'top-end' 32-bit (now 64-bit) microprocessors and a few 16-bit microcontrollers. However, many silicon companies are now implementing JTAG/boundary-scan across the board. SI Labs (formerly Cygnal) for example, offers an 8051 derivative micro-controller device C8051F120 with full boundary-scan and on-board JTAG debug: plus embedded flash which can also be programmed by register access from the JTAG port.

This, along with analogue inputs and DIO ports, all in a 64-pin package:

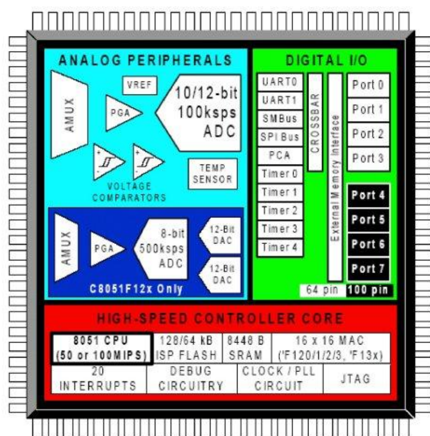


Diagram copyright SI Labs Inc.

With this burgeoning availability of IEEE 1149.1-compliant devices in evidence, more and more engineers are harnessing the power of JTAG/boundary-scan (and employing formalised test techniques) for the first time.

Until recently the available industry software tools for JTAG/boundary-scan application development have been at best esoteric and at worst long-winded, text-based systems requiring in-depth knowledge of programming syntax and the unit under test.

However, 'low-cost and easy-to-use' (a phrase not previously associated with JTAG/boundary-scan development tools) are now the watchwords of several newly launched development suites; for example, JTAG ProVision from JTAG Technologies.

Clearly, because of the way boundary-scan works (controlling the I/O pins of JTAG-compliant devices on a powered board) it is important to consider what affect running the boundary-scan tests will have on non-boundary-scan parts (often called 'clusters'). In the past it has been necessary to craft the tests such that they do not risk damaging non-boundary-scan parts (which may share nets with boundary-scan devices) when performing interconnect tests.

Now though, tools like ProVision can automate JTAG/boundary-scan test generation, as models are provided (or can be built/requested) for all non-boundary-scan compliant parts too. This is a major plus for engineers, in that it saves time and de-risks the tests.

Cluster models can not only be used to disable active parts but also include the test patterns (or vectors) needed for functional checking: for example it is possible to clock devices *via* boundary-scan to verify functional behaviour.

In addition to boundary-scan development tools (software), the associated hardware has also come a long way since the introduction of the standard. For example, controllers with multiple TAPs now allow engineers to design circuit boards with more than one boundary scan chain: thus giving the option for a dedicated chain for programming flash devices via a micro for example

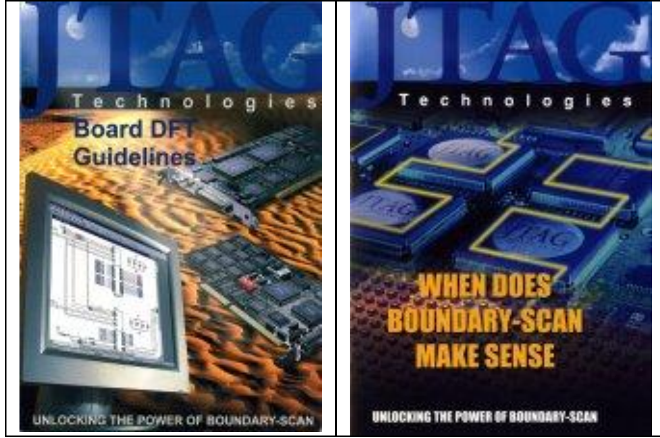
Additional tester TAPs also make it easier to 'test through connectors', using a Digital I/O Scan (DIOS) hardware module and accessing the edge of the board and/or test points. Such modules contain JTAG/boundary-scan compliant parts that synchronise with those on the board – and the use of modules can lead to the near 100% test coverage of a digital design.

Conclusion

JTAG/boundary-scan is playing an increasingly important role throughout products' life-cycles.

Whether your concerns are prototype debug, manufacturing process tuning or field service and firmware upgrades, JTAG/boundary-scan has a part to play. Moreover, as the technology disseminates through to all level of circuit board complexity then more engineers will be exposed to this invaluable test and programming technique.

If you wish to learn more about boundary-scan (and we have only skimmed the surface in this introductory piece), please note that two useful booklets are available from JTAG Technologies'. **Board DFT Guidelines** and **'When does boundary-scan Make Sense?**, are available for free. Simply email JTAG Technologies at info@jtag.com, giving your name and full address and setting the Subject as 'Guidelines'.


JTAG
Technologies
Board DFT
Guidelines
UNLOCKING THE POWER OF BOUNDARY-SCAN
JTAG
Technologies
WHEN DOES
BOUNDARY-SCAN
MAKE SENSE
UNLOCKING THE POWER OF BOUNDARY-SCAN