

PCB Test, Debug & Programming made easy with Universal Test Jig

GIREESHA B

Student, Dept. of VLSI and EMBEDDED Systems Engineering, UTL Technologies Ltd, #19/6, Ashokpuram School Road, Industrial Suburb, Bangalore, Karnataka 560022,INDIA

Abstract - In this paper, we focus on the Universal test Jig using XJTAG, which is developed for testing Printed circuit board assemblies (PCB's). XJTAG is not just a technology for PROCESSOR debug/emulation but also for programming FPGAs/CPLDs. XJTAG project contains the description of the circuit you want to test; it references the board net list(s), BSDL files for the JTAG enabled devices and other files that describe how individual types of devices that do not have JTAG capability on the board are to be treated.

Key Words: XJTAG, universal test jig, I/O adapter, XJ Developer, JTAG, XJLink 2.0, DUT, FPGA, BSDL.

INTRODUCTION

Universal test jig is a comprehensive solution for any type of PCB with different types of connector and components mounted either JTAG or non-JTAG. XJTAG provides suitable hardware and software platform for testing, troubleshooting & programming JTAG(Joint Test Action Group) based printed circuit boards.

I/O board is required along with XJLink 2.0 adapter to drive and sense signals to DUT, which is incorporated in the jig. Each I/O board can accommodate up to maximum of 208 signals making a total of 416 I/O signals.

Software platform i.e. XJDeveloper is used to develop, execute and debug test programs.

Hardware platform i.e. XJLink 2.0 is connected to PC using USB 2.0 cable.

JTAG specifies the use of a dedicated debug port implementing a serial communications interface for low-overhead access without requiring direct external access to the system address and data buses. The interface connects to an on-chip test access port (TAP) that implements a stateful protocol to access a set of test registers that present chip logic levels and device capabilities of various parts.

The JTAG standards have been extended by many semiconductor chip manufacturers with specialized variants to provide vendor-specific features.

In the 1980s, multi-layer circuit boards and non-lead-frame integrated circuits (ICs) were becoming standard and connections were being made between ICs that were

not available to probes. The majority of manufacturing and field faults in circuit boards were due to poor solder joints on the boards, imperfections in board connections, or the bonds and bond wires from IC pads to pin lead frames. It is providing a pins-out view from one IC pad to another so these faults could be discovered. "XJTAG has reduced the amount of time required to bring new boards to market, due to the ability to re-use existing device-centric tests." "XJTAG provides the speed and accuracy of diagnosis that we need when testing complex, densely populated boards with multiple BGAs and limited test points."

"Using the XJTAG system, we can now isolate a faulty device immediately and rework or re-flow the failed component to rectify the problem. This saves us time and increases our yields."

"Using XJTAG we can now test and debug our boards in minutes as opposed to days and we don't need dedicated people"

"XJTAG has exceeded our expectations. Its unparalleled speed, accuracy and ease-of-use have enabled us to save days, if not weeks, off the development phase."

"XJTAG is a fast, extremely versatile and cost-effective tool for generating high test coverage on PCBs containing both JTAG and non-JTAG devices."

Although JTAG's early applications targeted board level testing, the JTAG standard was designed to assist with device, board, and system testing, diagnosis, and fault isolation.

JTAG allows device programmer hardware to transfer data into internal non-volatile device memory (e.g. CPLD and FPGA's). Some device programmers serve a double purpose for programming as well as debugging the device. In the case of FPGAs, volatile memory devices can also be programmed via the JTAG port, normally during development work. In addition, internal monitoring capabilities (temperature, voltage and current) may be accessible via the JTAG port.

A JTAG interface is a special interface added to a chip. Depending on the version of JTAG, two, four, or five pins are added. The four and five pin interfaces are designed so that multiple chips on a board can have their JTAG lines daisy-chained together if specific conditions are met. The

two pin interface is designed so that multiple chips can be connected in a star topology. In either case a test probe need only connect to a single "JTAG port" to have access to all chips on a circuit board. Daisy-chained JTAG (IEEE 1149.1) the connector pins are

1. TDI (Test Data In)
2. TDO (Test Data Out)
3. TCK (Test Clock)
4. TMS (Test Mode Select)
5. TRST (Test Reset) optional.

1. PROPOSED SOLUTION

1.1 Block Diagram

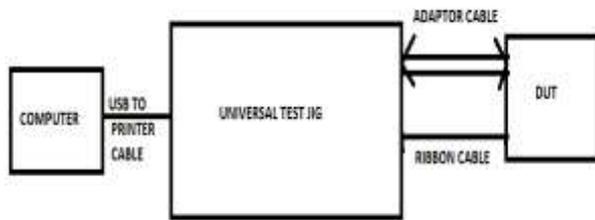


Fig 1: Hawk eye view of the testing setup

Universal test jig makes interface between computer and DUT. DUT is given power through jig power supply section which includes +/- 12 V, +/- 5V, +/- 3.3 V and 2 variable power supplies.. All connectors of DUT are connected to Universal Test Jig. Three VME type of connectors are available to connect to maximum of 480 pins. Adaptor cable is unique for each DUT hence we have to switch this cable for different DUTs .VME, EURO type of connector is supported for mating to the jig using adaptor cable.

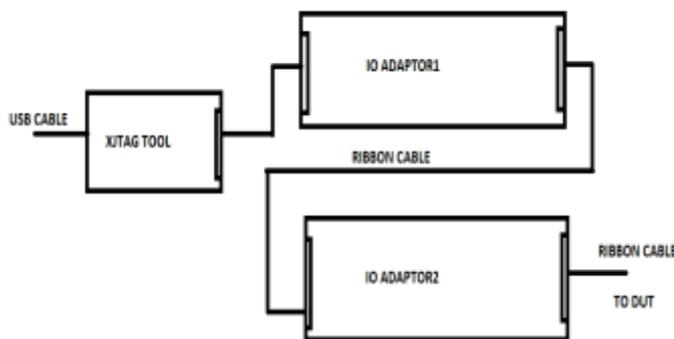


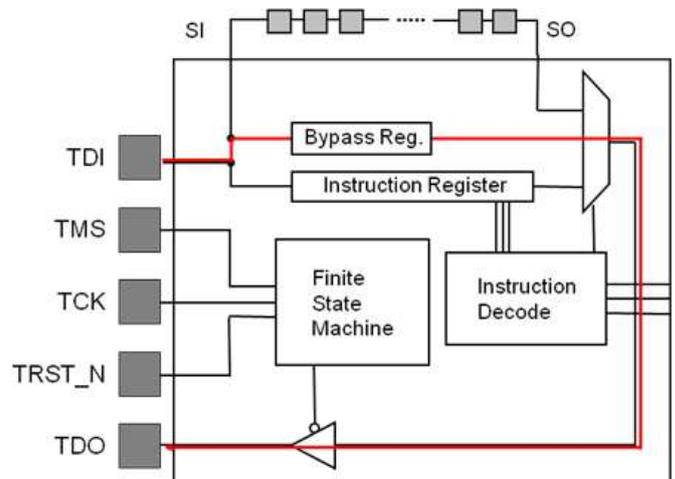
Figure 2

Fig. 2 shows internal jig block diagram, which consist of XJLINK 2.0 adaptor with two I/O PCBs. The interconnection is such that I/O PCB 1 JTAG chain is

connected serially to I/O PCB 2. TDI of chain 2 is connected of chain of TDO of 1 while TCK and TMS are connected parallel to both chains.

XJTAG tool i.e. XJLINK2.0 is connected to I/O PCB using ribbon cable 20 pin. Maximum of 4 TAP ports can be declared in XJDeveloper pin mapping which can control 4 TAP ports at a time.

1 TAP port is used to connect chain of CPLD in the I/O board itself. Other 3 TAP ports are useful in connecting to DUT having 3 different JTAG chains.



The TRST pin is an optional active-low reset to the test logic - usually asynchronous, but sometimes synchronous, depending on the chip. If the pin is not available, the test logic can be reset by switching to the reset state synchronously, using TCK and TMS. Note that resetting test logic doesn't necessarily imply resetting anything else. There are generally some processor-specific JTAG operations which can reset all or part of the chip being debugged. Since only one data line is available, the protocol is serial. The clock input is at the TCK pin. One bit of data is transferred in from TDI, and out to TDO per TCK rising clock edge. Different instructions can be loaded.

Instructions for typical ICs might read the chip ID, sample input pins, drive (or float) output pins, manipulate chip functions, or bypass (pipe TDI to TDO to logically shorten chains of multiple chips).As with any clocked signal, data presented to TDI must be valid for some chip-specific Setup time before and Hold time after the relevant (here, rising) clock edge. TDO data is valid for some chip-specific time after the falling edge of TCK.

The maximum operating frequency of TCK varies depending on all chips in the chain (the lowest speed must be used), but it is typically 10-100 MHz (100-10 ns per bit). Also TCK frequencies depend on board layout and JTAG adapter capabilities and state. One chip might have a 40 MHz JTAG clock, but only if it is using a 200 MHz clock for non-JTAG operations; and it might need to use a much

slower clock when it is in a low power mode. Accordingly, some JTAG adapters have *adaptive clocking* using an RTCK (Return TCK) signal. Faster TCK frequencies are most useful when JTAG is used to transfer lots of data, such as when storing a program executable into memory. Clocking changes on TMS steps through a standardized JTAG state machine. The JTAG state machine can reset, access an instruction register, or access data selected by the instruction register.

JTAG platforms often add signals to the handful defined by the IEEE 1149.1 specification. A System Reset (SRST) signal is quite common, letting debuggers reset the whole system, not just the parts with JTAG support. Sometimes there are event signals used to trigger activity by the host or by the device being monitored through JTAG; or, perhaps, additional control lines.

Even though few consumer products provide an explicit JTAG port connector, the connections are often available on the printed circuit board as a remnant from development prototyping and/or production. When exploited, these connections often provide the most viable means for reverse engineering.

Communications model: - In JTAG, devices expose one or more *test access ports* (TAPs). Example three TAPs, which might be individual chips or might be modules inside one chip. A daisy chain of TAPs is called a *scan chain*, or (loosely) a target. Scan chains can be arbitrarily long, but in practice twenty TAPs is unusually long. To use JTAG, a host is connected to the target's JTAG signals (TMS, TCK, TDI, TDO, etc.) through some kind of *JTAG adapter*, which may need to handle issues like level shifting and galvanic isolation. The adapter connects to the host using some interface such as USB, PCI, Ethernet, and so forth.

Primitives: - The host communicates with the TAPs by manipulating TMS and TDI in conjunction with TCK, and reading results through TDO (which is the only standard host-side input). TMS/TDI/TCK output transitions create the basic JTAG communication primitive on which higher layer

JTAG IEEE Std 1149.1 (boundary scan) instructions

Instruction register sizes tend to be small, perhaps four or seven bits wide. Except for BYPASS and EXTEST, all instruction opcodes are defined by the TAP implementor, as are their associated data registers; undefined instruction codes should not be used. Two key instructions are: 1) The BYPASS instruction, an opcode of all ones regardless of the TAP's instruction register size, must be supported by all TAPs. The instruction selects a single bit data register (also called BYPASS). The instruction allows this device to be bypassed (do nothing) while other devices in the scan path are exercised.

2) The IEEE 1149.1 (JTAG) standard describes a number of instructions to support boundary scan applications. Some of these instructions are "mandatory", but TAPs used for debug instead of boundary scan testing sometimes provide minimal or no support for these instructions. Those "mandatory" instructions operate on the Boundary Scan Register (BSR) defined in the BSDL file, and include:

- EXTEST for external testing, such as using pins to probe board-level behaviors
- PRELOAD loading pin output values before EXTEST (sometimes combined with SAMPLE)
- SAMPLE reading pin values into the boundary scan register

2. XJTAG

2.1 XJTAG SOFTWARE

○ Open Source

UrJTAG project supports many JTAG tools, processors, and boards.

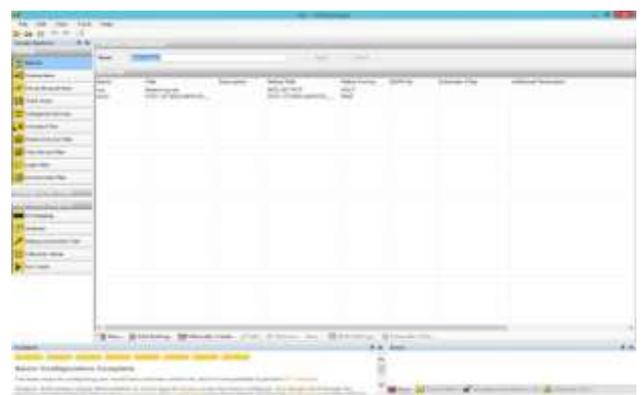
OpenOCD project supports various inexpensive JTAG adapters including USB ones based on FT2232 chips, and is mostly used with ARM projects. It provides GDB and telnet interfaces, both from Linux and from MS-Windows.

○ Freeware

Atmel provides AVR Studio on MS-Windows, for AVR8 microcontrollers, and a cross-platform AVR32studio product to support AVR32 systems.

Xilinx provides lower end FPGA development tools at no cost

HappyJTAG2 is embedded JTAG solution with FT2232 chip for AVR8 microcontrollers and Atmel AVR Studio on MS-Windows.



CREATING YOUR XJTAG PROJECT

Identifying the netlist

The netlist is a file output from the schematic editor, it describes the components that are on the board and how those components are connected. The netlist for the XJDemo board is contained in the ODB++ project in the XJDemo ODB++ zip file, which was copied to the project folder previously. When using ODB++ format you can either select the top level folder or a zip file that contains the entire folder structure

Identifying the BOM

The BOM file provides extra information about the components on the board. This file is not required to create a test system with XJDeveloper but the extra information that it provides can be very useful in allowing XJDeveloper to automatically categorise devices in the circuit

Identifying the power/ground nets

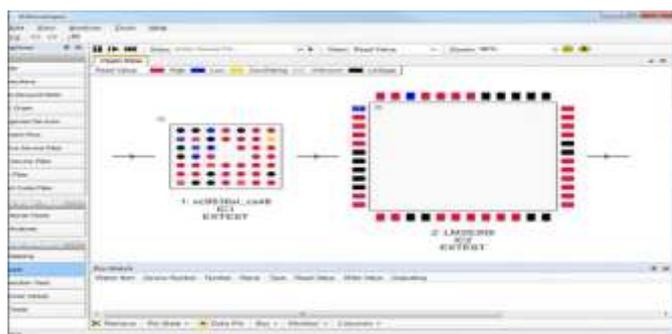
There are three reasons why XJDeveloper needs to know which nets are power and ground

Identifying the JTAG chain

Only some of the devices in any circuit are enabled with JTAG boundary scan; the next stage increasing the test system is to identify those devices in the circuit and associate the BSDL files that describe the JTAG implementation in each device

Reviewing test coverage and running the Connection Test

- It is possible to start using XJDeveloper to test your circuit almost as soon as the description of the JTAG chain is complete. At this stage the Connection test can be used to test any net that only contains pins that are on the devices in the JTAG chain.
- You can review the test coverage that your test system will achieve at any stage of your setup



ADVANTAGES

- Need for physical test points on board is eliminated => simpler board layouts
- Cheap test fixtures
- Reduced time on in-circuit test systems
- Increased use of standard interfaces
- Faster time-to-market
- Reduce your time & cost of board development and manufacturing test
- Create your own reusable test library and preserve your test IP
- Regain visibility of BGA interconnects hidden under the component
- Debug and test complex PCBs using our 'out-of-the-box' boundary scan solution

CONCLUSIONS

A simple approach is proposed in this paper to reduce the area, power, time and test coverage.

XJTAG is very simple at the hardware level as it is controlled by "XJDEVELOPER software"

XJTAG's real expertise lies with upper-level algorithms hidden inside commercial and open source tools, which provide JTAG tools the ability to control Flashes, processor machine instruction stepping, memory controller registers, and other facilities needed for the board initial startup (tuning SDRAM timings, GPIO multiplexing, initial console UART, etc.)

Universal Test Jig can fully exploit the features of XJTAG for functional testing as well as open and shorts using just programs and adapter cables.

ACKNOWLEDGMENT

The authors would like to thank UTL Technologies Ltd, and the Visvesvaraya Technological University for the support extended. The authors would also like to thank the colleagues, friends and family for their support.

REFERENCES

1. Wikipedia - Internet of things - <https://en.wikipedia.org/wiki/JTAG>
2. http://www.ee.ic.ac.uk/pcheung/teaching/ee3_DS_D/ti_jtag_seminar.pdf
3. <http://www.inaccessnetworks.com/projects/ianjtag/jtag-intro/jtag-intro.html>

4. http://toolbox.xilinx.com/docsan/3_1i/data/common/jtg/dppa/appa.htm
5. Computer Design, January 1994, "Testing Dilemmas and Corporate Alliances Fuel Boundary Scan's Acceptance"
6. Test and Measurement World, October 1992, "Concurrent Engineering is Common Sense"
7. Computer Design, November 1992, "Design and Test Engineers Alter Rules to Facilitate Test"
8. EDN, December 3, 1992. "No 'Accounting' for Boundary Scan Test"
9. Neal Stollon (2011). On-Chip Instrumentation. Springer.
10. Randy Johnson, Steward Christie (Intel Corporation, 2009), JTAG 101—IEEE 1149.x and Software Debug
11. Oshana, Rob (29 October 2002). "Introduction to JTAG". Embedded Systems Design. Retrieved 5 April 2007.