International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-2 Issue-15 June-2014

Tessent BSCAN Insertion on 28nm SOC

M. A. Wavhal[1], S. U. Bhandari[2]

Abstract

The Testing plays vital role to ensure the correctness of chip functionality. Boundary scan is a structured design-for-test technique which makes digital I/O pins testable by means of inserting boundary scan cells between core logic and pins. It enhances chips accessibility and testability. This project has implemented the Boundary scan on 28 nm SOC having approximated 5 million gate counts. Total pin count of SOC is 107.The project work is done on Linux platform. Tcl scripting language is used for setting parameters for design and placing runs on servers. File manipulations are done in Vi editor. BSCAN is done using Tessent BSCAN tool from Mentor Graphics. The simulations are carried out on NC Verilog simulator from Cadence. The outcome of project is, Gate level Netlist with BSCAN architecture inserted. The BSCAN cells inserted have length of 146. To cover entire SOC, the scan insertion and MBIST can be performed.

Keywords

Boundary Scan (BSCAN), SoC, Mentor Graphics, Tcl (Tool command language).

1. Introduction

Design for Test ("Design for Testability" or "DFT") is a name for design techniques that add certain testability features to a chip. The purpose of manufacturing tests is to validate that the product hardware contains no defects that could, otherwise, adversely affect the products correct functioning. Testing has two major aspects: control and observation. To test any system it is necessary to put the system into a known state, supply known input data (test data) and observe the system to see system to see if it performs as designed and manufactured. If control or observation cannot be carried out, there is no way to know empirically if the system performs as it should [1].

Boundary scan is a structured design-for-test technique, standardized as IEEE 1149.1 standard. In Boundary scan, a scan shift-register stage is placed adjacent to every input or output pin of the chip i.e.at the component boundaries. These shift registers are nothing but the boundary scan cells.

The cells are connected around the periphery of the IC, which forms the boundary scan path. Data can flow directly through the boundary-scan cell when normal operation of the component is required [2]. During testing, the cells at output pins can be used to drive signal values onto the external network, while those at the input pins can capture the signals received.

2. Background & Relevance

High-volume product lines require a high level of confidence in the components before final system assembly. Most system manufacturers find it too expensive and difficult to completely test an assembled system on the manufacturing floor. It is also virtually impossible to measure test coverage of the functional system tests used in manufacturing. Even if systems could be tested exhaustively, it is a major challenge to identify and replace faulty boards or components. Therefore, as chips and boards get increasingly complex, adequate testing of the chips and boards is mandatory. While it is easier to test the components stand-alone before system assembly, the cost of a defective component that enters the manufacturing process can be high and will increase in proportion to how late in the process the component is identified and replaced [3].

If the limited tests in manufacturing do not identify the defective component, the system may be shipped to a customer, in which case the cost of the defect can be very high indeed when the component finally exhibits the failure mechanism. The manufacturing process thus places the bulk of the test burden at the component level and relies on high quality component tests for overall success. Clearly, the components (both chips and boards) need to be

This work was supported in part by LSI India research and development Pvt. Ltd (An Avago Technology Company) under the guidance of Mr. Priyesh Kumar.

Manuscript received June 18, 2014.

M. A. Wavhal, Department of Electronics & telecommunication, Pimpri Chinchwad College of engineering, Pune, India.

S. U. Bhandari, Department of Electronics & telecommunication, Pimpri Chinchwad College of engineering, Pune, India.

highly testable in order for the system to be manufacturable in volume. Manufacturing testing enables development teams to screen devices for manufacturing defects.
Boundary scan technique covers for stuck-at, interconnects, open and shorts type of faults for digital I/O pins of the chip. It offers the controllability by facilitating the internal input nodes handling with primary inputs and offers observability by facilitating the internal output nodes handling with primary outputs [4].

3. Boundary scan details

This project covers complete boundary scan on SOC which will be manufacture using 28 nm Technology. The Gate count is around 5 million. To perform boundary scan the Tessent tool undergoes a particular flow. The flow is as follows:

ETChecker-clocks
This step extracts the information of the RTL design along with clock architecture. It provides the details of clock network graphically as well as specifies properties that includes the parameters viz. functional frequency, label etc. (Values are to be entered by DFT engineer)[5]. The table after each step describes generated files and directories:

Table 1: ETChecker-clocks Outcome

Sr. No	File/directory	Description
1	etCheckInfo	Default output Directory where logs and reports are generated after run.
2	my design. etpClockTree	My design Contains no clock source information due to only Bscan mode
3	my design. EtpClockDomainI-nfo	My design Contains noclock domain base information due to only Bscan mode

ETChecker-Rules
It checks for design rules to ensure that your circuit is free of violations and meeting embedded test requirements described in RTL Description Requirements. This run requires several iterations to identify and correct the design aspects that do not meet the requirements for implementing Mentor Graphics Embedded Test [5].

Table 2: ETChecker-rules Outcome

Sr.No	File/directory	Description
1	etcHand-off	Directory consisting of etCheckerInfo file which is to be handed off for next step run
2	etChekerInfo	Total pin count: 75 Module Name: My design Contains list of: 1) Pin names with direction 2) Black boxes 3) Constant pins 4)Instantiated Modules
3	RootModule. etpConstraints File	Contains specified constraint such as blackbox isolation, pin JTAG options, Pad cells. Also inferred constraints define asserted value on pins.

Run-ETPlan-gen
The ETPlanner uses etCheckerInfo file and other optional files such as .LVICTech File, .ET-Defaults File, .CADSetup File and DEF or PDEF Files to generate my design.etplan file. This file describes the embedded test requirement plan for chip. The generated plan provides a correct by construction embedded test plan for the design based on the user-defined requirements [6, 8].

Table 3: Run-ETPlan-gen Outcome

Sr. No.	File/directory	Description
1	OutDir	The Directory contains following listed files
2	etplanner.log-checkplan	It indicates CPU time, page faults, max data size and error summary
3	My design. etplan-diffs	In my design, the Sim-Model Dir and Hierarchical DirPath are different.

Make-checkplan
In this mode, ETPlanner uses the .etCheckerInfo file and the .etplan file to check the embedded test plan against embedded test compatibility rules and generate a report (ET Summary) that provides information related to test time and power requirement for the embedded test plan. This report

also provides all warning associated with the .etplan file content [6, 8].

Table 4: Make-Checkplan Outcome

Sr. No	File/directory	Description
1	My design. ETSummary	The file shows that approximately 49 flip-flops are used for
2	My design .etplan	For next step run the link and Hierarchical data path is specified where design Netlist is stored.
3	My design.etplan. README	Contains complete syntax of .etplan file for reference
4	Makefile	It has several targets that allow user to archive the generated workspace.
5	etplanner.log-genplan	It indicates CPU time, page faults, max data size and error summary

Gen-LVWS

In this mode, ETPlanner uses the .etplan file (generated with mode GenPlan) and the .etCheckerInfo file to generate Mentor Graphics embedded test environment that contains the directory structure recommended for the automation of the embedded test generation and verification, make targets, configuration files, etc. The .etCheckerInfo and .etplan files are mandatory for this mode [6, 8].

Table 5: Gen-LVWS Outcome

Sr. No.	File/directory	Observations
1	MY-DESIGN-LVWS	It contains the directories generated for next steps as well as Make- file and Readme file
2	etplanner.log-genLVWS	It indicates CPU time, page faults, max data size and error summary
3	Makefile	It has several targets that allow user to archive the generated workspace
4	MY-DESIGN. etassemble	It contains parameters which needs to be set as NumberUserDRbits:1 NumberUserbits:20 numberBist-Ports:17 for my design
5	MY-DESIGN. Embedded Test	For my design it uses Synthesis Time Units: ns Synthesis Tool: DCTCL Design- Compiler Version: POST-X-2005.09

Embedded-test

This step is performed using make Embedded-test target. It makes use of my design.etassemble file generated by ETPlanner. It generates the TAP RTL for your design. Also the scripts to convert RTL to Gate level Netlist [7,8].

Table 6: Embedded-test Outcome

Sr. No.	File/directory	Description
1	MY-DESIGN. pinorder	My design has 107 pins.
2	MY-DESIGN. Tap	It displays test port pins (tck, trst, tms, tdi, tdo) assignments, signal connections and pad info etc.
3	etassemble. log	For my design it indicates: Total time: 124.47 units CPU time, page faults, Max data size: in Mbytes and error summary
4	MY-DESIGN-LVISION-JTAP.config	For my design the opcode is set to 40 bits.
5	Mydesign.bsdl	In port list it givesdirection as in, inout and linkage, Pin mapping shows pin number and name, bscan definition shows Boundary length of my design is 146.
6	MY-DESIGN.et	It is the Netlist generated by tool.
7	MY-DESIGN-LVISION-JTAP.	It shows TAP controller Synopsis synthesis script.
8	MY-DESIGN-etassemble-modal.sdc	It has procedures for functional mode, pin asserts, JTAP timing and embedded test mode.
9	MY-DESIGN-LVISION-JTAP.vb	It gives TAP controller RTLdescription
10	MY-Design. Startup	The clock period of TCK pin is set to 100 ns.
11	SDC-remapping-procs. tcl	It contains remapping of Tessent STA constraints which may have got altered by synthesis or layout tool.

12	MY-DESIGN-LV-BGROUP.synopsys-tcl	It contains bscang i.e. bscan group synopsis tcl synthesis script

Designe
This step involves generating test benches for verification and performing simulation. The testbenches generated by ETVerify at this stage verify all types and instances of embedded test controllers in the given physical region. It performs the rule check and generates Test Connection Map file [7, 8].

Table 7: Design Outcome

Sr. No.	File/directory	Description
1	MY-DESIGN.tcm	The test connection map file showing assignments for 107 pins
2	designe.log-MY-DESIGN	Showing Max data size of 11193 Mbytes and 1 warning.

Edit-synthesis-script
It modifies or updates the scripts written in step 6, to make scripts compatible with test mode specified. For my project it keeps only BSCAN related [7].

Table 8: Edit-synthesis-script Outcome

Sr. No.	File	Description
1	MY-Design-LV-BGROUP.synopsys-tcl	It is bscang SYNOPSYS TCL synthesis script
2	My-design-LVISION-JTAP.synopsys-	
	It is TAP controller Synopsys tcl synthesis script tcl	

Make-synth
This optional make target synthesizes all embedded test controllers which were generated by ETAssemble [7].

Table 9: Make-synth Outcome

Sr. No.	File	Description
1	My-design-LVISION-JTAP.v	It is JTAG TAP inserted Netlist
2	My-design-LVJEN-1.v	It is Netlist of My-design-LVJEN-1 module
3	My-design-LVJBID-S.v	It is Netlist of My-design-LVJBID-S.v module
4	My-design-LVJEN-1-NM.v	It is Netlist of My-design-LVJEN-1-NM.v module
5	My-Design-LVJBID-NM.v	It is Netlist of My-design-LVJBID-NM.v module
6	My-design-LVJBID-D-NM.v	It is Netlist of My-design-LVJBID-D-NM.v module
7	My-design-LVJBID-S-DO.v	It is Netlist of My-design-LVJBID-S-DO.v module
8	My-design-LV-BGROUP-DEF.v	It is Netlist of My-design-LV-BGROUP-DEF.v module
9	command.log	This file define the behavior of many parts of the Synopsys Synthesis Tools

Concatenate-Netlist
It merges top level Netlist with Netlist containing TAP. Then it generates a single top level Netlist with DFT inserted in it [8].

Table 10: Concatenate-Netlist Outcome

Sr. No.	File	Description
1	Mydesign. v-postLV	This Netlist is generated by Design Grabber using ETChecker Netlist.
2	designg.log	This is the log file showing summary about design

Config-etsignoff
This make target runs ETVerify to create the .etSignoff configuration file. This file defines testbench configuration for design. It contains a list of test steps which is required to perform early verification of all embedded test features on design [8].

Table 11: Config-etsignoff Outcome

Sr. No.	File	Description
1	My-design. etSignOff	It includes run tests such as TestLogicReset, InstReg, IDReg, BypassReg, TAPIntDR, BscanReg, Input, Sample, HighZ, Clamp and Output.
2	READM E-etv	It gives the wrapper and some parameters for logicbistverify.
3	My-design.jtag-phy	It gives the ODD and EVEN polarity for different pins like GPIO port, uart etc.
4	My-design. Loadboard - info-tpl	It defines the wrapper for JTAP, CE and 1149.1 pins.
5	My-design-collar.sim	No simulation targets for embedded LVDBs needed to be generated
6	etv.log-config-etSignOff	This file shows the there is 1 TAP controller in top module with single frequency group of period 100.0 units

Lvdb-prelayout

It generates a pre-layout Logic Vision circuit database. The pre-layout LVDB is stored in the ETSignOff Directory. This ensures that within only the ETSignoff directory, you will be able to sign off the final post-layout version of your chip [8].

Table 12: Lvdb-prelayout Outcome

Sr. No.	File	Description
1	etv.log-CreateLVDB	No error messages are generated

Make-testbench

This make target runs ETVerify to create test benches for the verification of all embedded structures. For this it makes use of .signoff configuration file and the data from pre layout lvdb [8].

Table 13: Make-testbench Outcome

Sr. No.	File	Description
1	Design.vif	It stores the pattern and waveforms
2	tapbistv.v	It is the Netlist of tapbist
3	etv.log testbench	No errors are present so next step run can be done.

PostLV-MBIST-Edits

This step will update the Netlist to make some TAP connections and providing controllability for power control signals (LS (Light Sleep), DS (Deep Sleep), SD (Shut Down)) of the memories.

Table 14: PostLV-MBIST-Edits Outcome

Sr. No.	File/directory	Description
1	logs	It has PostLV-MBIST-Edits .log file
2	Netlists	It has Netlist My design.postLV
3	My-design-updates4mbist.v	This file make Netlist ready for LV
4	POSTLV-MBIST-EDITS.log	This log file gets generated after this step run

PostLV-BCAD-Edits

This step will update the Netlist to provide controllability to the pad input pins like PU, PD, DR0, DR1 etc. These changes are necessary for BSCAN simulations to work fine.

Table 15: PostLV-BCAD-Edits Outcome

Sr. No.	File/directory	Description
1	Command.log	Showing procedures group-variable, read-Verilog, read-vhdl etc.
2	POSTLV-BCAD-EDITS.log	This log file gets generated after this step run

4. Simulation Results

After completing entire boundary scan flow, the simulation of design is carried with cadence ncverilog Simulator. Once simulation is completed the log file shows whether it is successful or done with violations. For my design initially there were 226 compare failures. These failures were removed by tracing schematic backwards from the point of failure occurrence. After locating root for failure, the MUXes with required logic are inserted at appropriate level [9]. Fig.1 depicts the schematic tracing window of cadence simulator and fig.2 depicts snapshot console window with successful simulation.

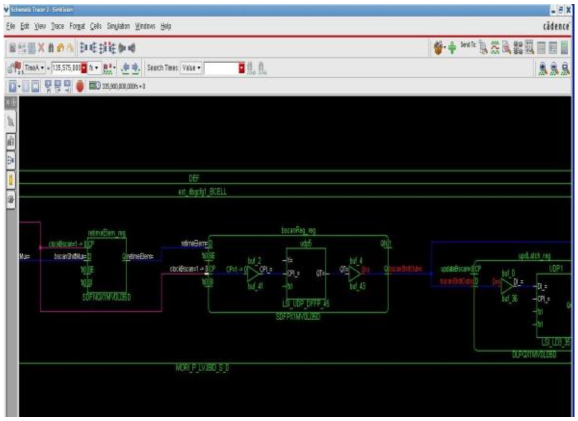

Figure 1: Schematic tracing window

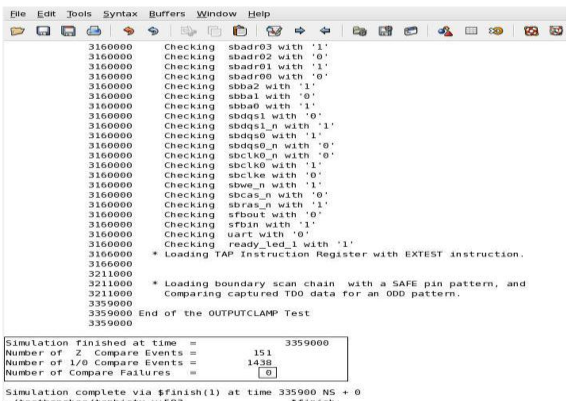

Figure 2: Simulation result console window

5. Conclusion & Future work

Design for Test (DFT) is important as functional verification cannot fully represent to detect manufacturing defects. My project has implemented the Boundary scan chain on SoC to be manufacture in 28 nm technology. For digital I/O pins, the outcome obtained is presented in table below.

Table 16: BSCAN Outcome

Sr.No	Parameters	Value
1	Total pin count	107
2	Boundary scan flip-flops	49
3	Length of Boundary scan chain	146
4	BSCAN pin count	75
5	Instruction Register opcode	40
6	Clock period	100 ns
7	No. Of Z compare events	151
8	No. Of I/O compare events	1438
9	Simulation time	335900 ns

After covering digital I/O pins, the scan insertion can be done on chip in order to cover entire combinational and sequential logic. This can be achieved by chaining sequential logic together as combinational logic couldn't be observed completely. The sequential cells adjecent to combinational logic can capture the data for testing. Once internal logic and I/O pins become testable, the memories can be made testable by means of MBIST i.e. Memory Built In Self-Test. Tessent tool can be used to insert complete testability to SoC.

References

[1] IEEE standard for reduced pin and enhanced functionality Test Access Port and Boundary scan architecture IEEE computer society, 2009.
[2] Mentor Graphics Embedded Test Hardware Reference Software version 9.1, August 2010.
[3] R G Ben Bennetts "Boundary scan tutorial" version 2.1, September 2002.
[4] Colin M. Maunder, Rodham E. Tulloss The test access port and boundary scan architecture IEEE computer society press tutorial.
[5] Mentor Graphics ETchecker usage guide and Reference Software version 9.1, August 2010.
[6] Mentor Graphics ETPlanner tool Reference Software version 9.1, August 2010.
[7] Mentor Graphics ETAssemble tool Reference Software version 9.1, August 2010.
[8] Mentor Graphics Tessent SoC Usage Guide Software version 9.3, February 2011.
[9] Cadence "Design Browser" Product version 11.1, March 2012.

Mayuri Wavhal received her B.E degree from Pune University in 2010. She is currently pursuing ME in VLSI and Embedded systems from P.C.C.O.E Pune University. She is currently working in LSI India Pvt. Ltd as Intern in DFT (Design For Testability) area.

Sheetal Bhandari is an assistant professor of electronics and telecommunication engineering at Pimpri Chinchwad College of Engineering, University of Pune, India. She received her B.E. and M.E. degree from University of Pune in 1998 and 2006 respectively. She has completed her PhD in the area of Reconfigurable Computing in 2013. She is been teaching for about 8 years and has entrepreneurial stint of 4 years. Her academic focus is on Microelectronics and VLSI Design. Her research interests include Partial Reconfiguration and HW-SW Co-Design.