Specific Design and Optimization of JTAG IP Core

Xiaobo Zhang Yanfeng Jiang* Jiaxin Ju
Department of Microelectronics, College of Information Engineering
North China University of Technology,
Beijing, 100144, China
*Corresponding Author: yfjiang@ncut.edu.cn

Abstract—**A JTAG IP core based on IEEE1149.1 standard has been reported here, including its design and implementation. It has been described using synthesized Verilog HDL language. Simulation demonstration has also been made and the result has been synthesized. It has been demonstrated that the IP core design is feasibility. Moreover, based on the characteristic of DFT using JTAG standard, some improvements on the JTAG structure are proposed to get an optimized result.**

Keywords-IEEE 1149.1; IP core; ATE

I. Introduction

IP core refers to that pre-designed model with complex functions, which can be used in multiply applications. There are three kinds of cores, including soft core, hard core and fixed core. Soft core refers to the HDL code at the RTL level and gate level. Fixed core means the netlist after synthesized, in which users are permitted to define key parameters according to different technology. Hard core means that all designs have been finished and technology been fixed, in which behavior level after package can be offered by factory.

There are two fundamental ways to apply testability to a design: ad hoc (or unstructured) and structured. Ad hoc testability is simply intended to solve a particular test problem (i.e., adding a test point to observe a signal line during board test). While this may be useful for a particular test environment, in this case factory board test, it may not be useful for any other test environment, such as IC or system test. Structured testability, on the other hand, is designed to be useful at several levels of test. For instance, an IC with built-in self test (BIST) and a standard test interface can be used for IC test, board test, and system test. This type of testability is considered structured because it is standardized and can be used in several test environments.

Before the formation of the Joint Test Action Group (JTAG) and the IEEE 1149.1 standard, the Test Automation Department of TI's Defense Systems and Electronics Group (DSEG), had considered boundary scan as a method to improve the test, integration, and maintenance of systems being designed for the Department of Defense (DoD). From this study, an architecture was developed, along with a library of specialized test cells particularly well-suited for boundary-scan and BIST applications. The test architecture, referred to as System Controllability, Observability, and Partitioning Environment (SCOPE□), provides boundary scan and BIST capability to each input and output pin of the host IC. The architecture is supported by a library of modular bit slice called SCOPE cells that offer a range of boundary test capability. Some of the cells are targeted for simple boundary-scan applications. Other cells support the design of more sophisticated boundary test circuits such as pseudorandom and binary pattern generators and parallel signature analysis registers. During test, the SCOPE cells receive control from the test bus interface to execute a boundary scan or BIST controllability and observability test operation. One novel feature of the SCOPE architecture is its ability to activate the boundary test circuits while the host IC is in a normal operational mode. This capability provides boundary test features to support system integration, emulation, and at-speed testing.[1-5]

Now, almost all the chips have integrated the JTAG interface. As we know, it is characterized as easy operation. So, a demand has rised that a JTAG IP core should be designed and applicable for common user. In this paper, an optimized JTAG IP core has designed and the implemented method will introduced in Part II and Part III.

II. DESIGN And OPTIMAZATION of 1149.1

The IEEE 1149.1 test bus and boundary-scan architecture allow an IC, and similarly a board or system, to be controlled via a standard four-wire interface. Each IEEE 1149.1-compliant IC incorporates a feature known as boundary scan that allows each functional pin of the IC to be controlled and observed via the four-wire interface. Test, debug, or initialization patterns can be loaded serially into the appropriate IC(s) via the IEEE 1149.1 test bus. This allows IC, board, or system functions to be observed or controlled without actual physical access.

The IEEE 1149.1 test bus comprises two main elements: a test access port (TAP), which interfaces internal IC logic with the external world via a four-wire (optionally, five-wire) bus; and a boundary-scan architecture, which defines standard boundary cells to drive and receive data at the IC pins. IEEE 1149.1 also defines both mandatory and optional opcodes and test features.

The test bus signals are: Test Clock (TCK), Test Mode Select (TMS), Test Data In (TDI), Test Data Out (TDO), and the optional Test Logic Reset (TRST). The IEEE 1149.1 specification also specifies that the ICs can be connected in

978-1-4244-2587-7/09/$25.00 ©2009 IEEE

either a ring or star configuration. A simplified block diagram of the architecture is shown in Figure 1. The design process for the function in Fig.1 is shown below. Some optimization methods have been included during the IP design[1-2].

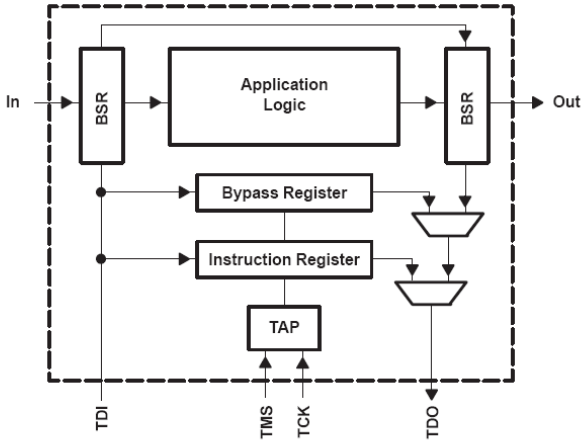

Figure 1. IEEE 1149.1 Scan-Bus and Boundary-Scan Architecture

The TAP controller is a 16-state finite state machine that controls the operation of most of the boundary scan circuit. A clock (TCK) is used to control transitions while the Test Mode Select (TMS) signal is used to control the state of the controller. TRST is used to reset the state of the controller and is active low. A number of control signals are output from this to other portions of the design. Fig.2 shows its state diagram.

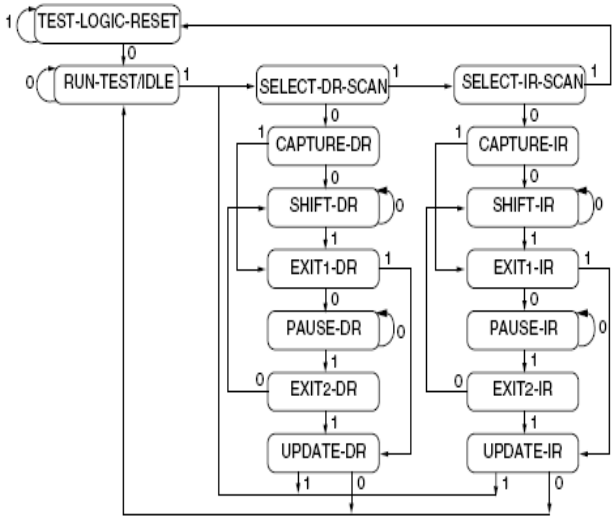

Fig.2 State diagram for TAP controller

The TBIC (Test Bus Interface Circuit) is used to control the connections between AB1, AB2, AT1, and AT2. It also provides some additional functionality such as pulling a bus to a high or low reference voltage and providing a comparator output for digital threshold comparison[3]. The single bit digitizers have been implemented by connecting the analog pins to digital buffers. The table below represents the possible switching patterns for each test (note that each test defines the Mode1 and Mode2 signals).

The TBIC is seperated into three different stages: the registers, the control logic, and the switches, as shown in Fig.3.

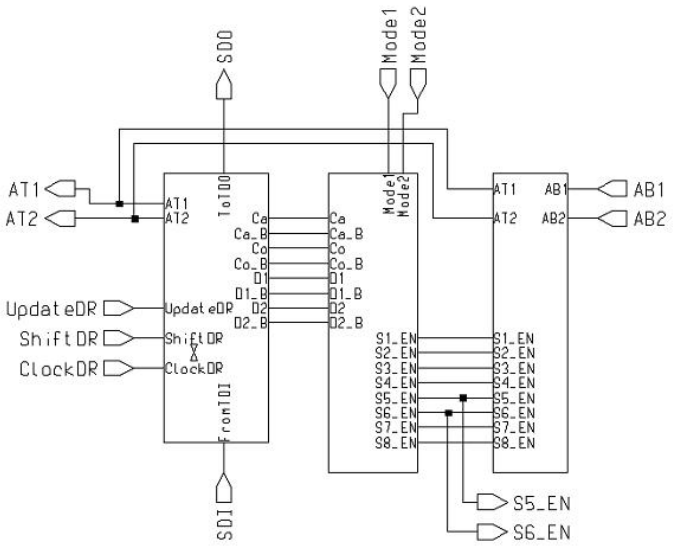

Fig.3 Schematic of TBIC structure in three stages

This is the register stage of the three stages to the TBIC. Bits are shifted into a control register and then loaded into an update register. The outputs of that register leads to the control logic. This is the input to the TBIC and recieves many signals from the TAP.

Fig.4 shows the circuit schematic of the register. In Table 1, the simulation results for TBIC register are listed.

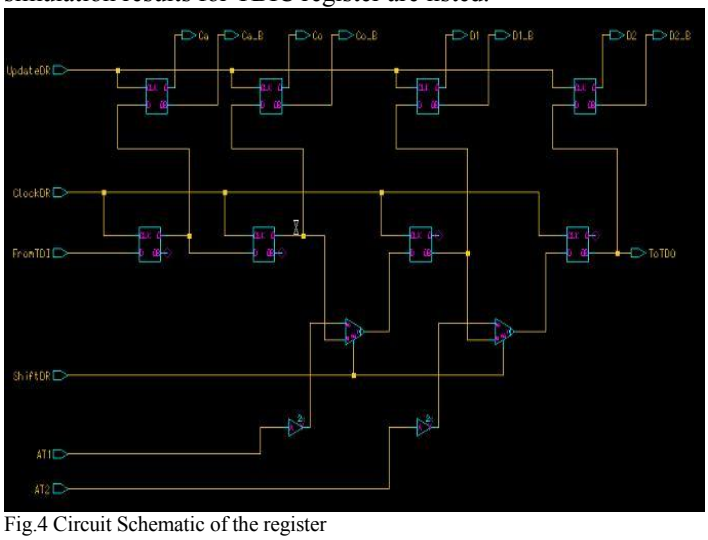

Fig.4 Circuit Schematic of the register

TABLE I. SIMULATION RESULT FOR THE REGISTER

Parameter	Simulation Value
Cell Dimensions (lambda)	830x430
Power Consumption (mW)	5.75
Output Transition Time (ns)	0.88
Rise Time (ns)	0.53
Fall Time (ns)	0.53

Clock DR, from the TAP, signals the loading of the control register's flipflops. Shift DR, also from the TAP, when high enables the shifting function of the control register. Shifting begins at TDI and ends at TDO. When Shift DR is low, capture mode is on, where AT1 and AT2 external signals are loaded into the control register. Thus, ShiftDR is the select signal of a 2X1 multiplexor. When UpdateDR, also from the TAP, is enabled, the values from the control register are loaded into the Update register. The outputs of these flipflops are part of the control logic inputs.

Fig.5 shows the logic functions used to control the switches, which are adopted from the dot4 standard document[4]. These functions use the outputs from the TBIC control registers and the two mode signals from the instruction decoder.

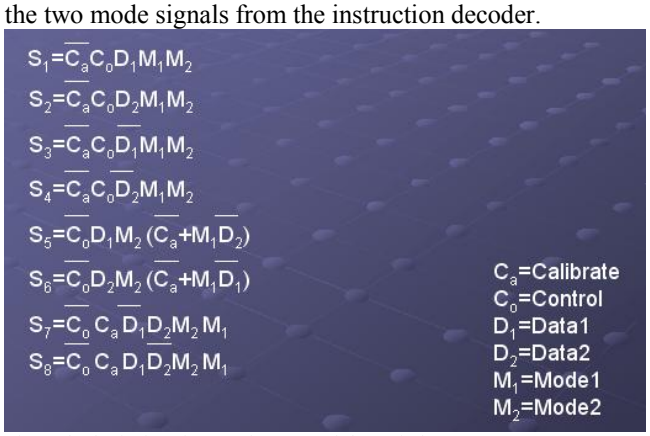

Fig.5 The logic function used to control the switches

2.2.3 TBIC switches

This is the switching network of the TBIC. Each switching block is a transmisson gate shown in Fig6. The enable of each and every block is defined by the control logic outputs. Switches 5 through 8 have external signals AT1 and AT2 as inputs, as shown in Fig.5. However, Switches 1 through 4 may override those signals to either voltage high or ground. The output of the whole network is reduced to two signals which are AB1 and AB2, the Analog bus signals.

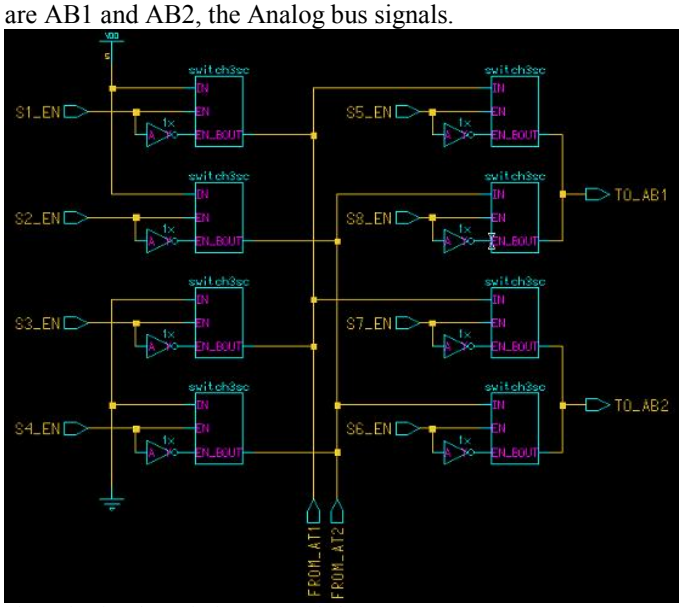

Fig.6 Circuit Schematic for TBIC switches

The miscellaneous parts are those pieces not contained in a larger module. They consist of the parts described in the following sections. Its block is shown in Fig.7.

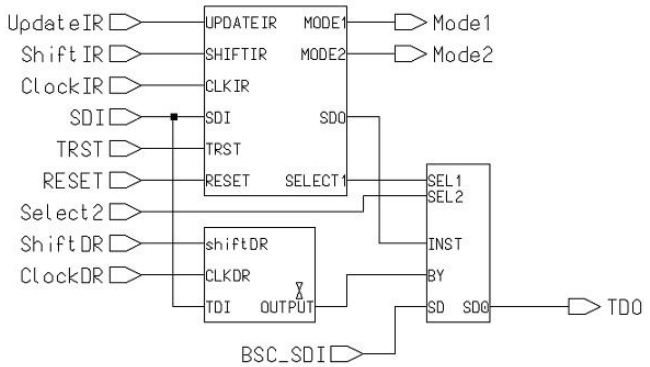

Fig.7 Block Diagram of Miscellaneous parts

This register allows instructions to be shifted in serially and then loaded onto parallel outputs. The instructions are 2-bits to handle the 4 mandatory instructions for 1149.4: Bypass, Sample/Preload, Extest, and Probe. Below is a short description of the functionality each of these functions performs.

--BYPASS – This allows a device in a chain of devices to be bypassed via a 1-bit register. This might be done to speed up testing of other components.

--SAMPLE/PRELOAD – This allows a device's core logic to be tested by sampling the inputs and outputs of a device during normal operation. It is also used to prepare for EXTEST by preloading values used for external testing.

--EXTEST – This allows for interconnect testing among 1149.1 devices. A pattern is preloaded on some pins known to be connected to another set of pins which are then read and their values compared.

--PROBE – Allows analog pins to be connected to the ATAP using internal buses so that external complex impedances can be tested by applying a known external source.

The bypass register is a single bit shift register that allows a device in a boundary scan chain to be bypassed. This speeds things up when a particular device does not need to be included in the chain.

This circuit allows the selection of the proper serial register. In our design this includes the bypass register, instruction register, or the boundary scan register.

Table 2 shows the simulation results of bypass register and output MUX. Based on the simulation result, the rise time and falll time of the two circuits coincide with the demand of the standard[5].

TABLE II. SIMULATION RESULTS OF BYPASS REGISTER AND OUTPUT MUX

Bypass Register		Output MUX	
Rise Time (ns)	Fall Time (ns)	Rise Time (ns)	Fall Time (ns)
0.42	0.4238	0.549	0.399

III. Result and Discussion

In this design, the simulation is carred out based on ModelSim SE6.1b software. Its result are shown in Fig.8.

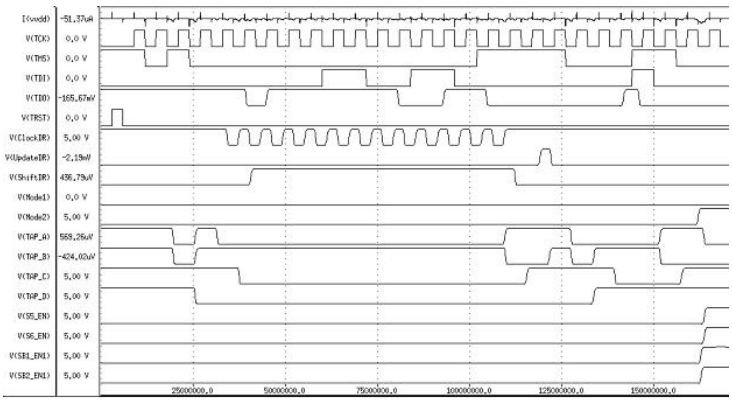

Fig.8 Simulation Waveform of TAP Controller

The concrete simulation process is in the following sequence:

The instruction register is written by code 1011. Reset the measuring logic circuit by tck=0;trst=1. After 1 clock cycle, trst=0,tms=0, the state is in Measuring/Duty. After another clock cycle, tms=1,IDCODE instruction is carried out.

After a clock cycle delay, instruction register is written by BYPASS instruction code(0101). After another clock cycle, tms=1 and bypass function operates.

Instruction operations are carried out in sequence as INCODE, SAMPLE/PRELOAD, BYPASS , INCODE , SAMPLE/PRELOAD , EXTEST,DEBUG.

After a clock cycle delay, write 0100 and 0010 in sequence into instruction register. For both of them are not allocated to specific operation instruction, the decodes during simulation have been selected as BYPASS.

After a clock cycle delay, SAMPLE/PRELOAD,MBIST are carried out in sequence.

In thus sequence, all modules can be demonstrated and simulated in their operation mode.

The synthesized result of JTAG is shown in Fig.9, based on Symplify Pro 7.3 environment.

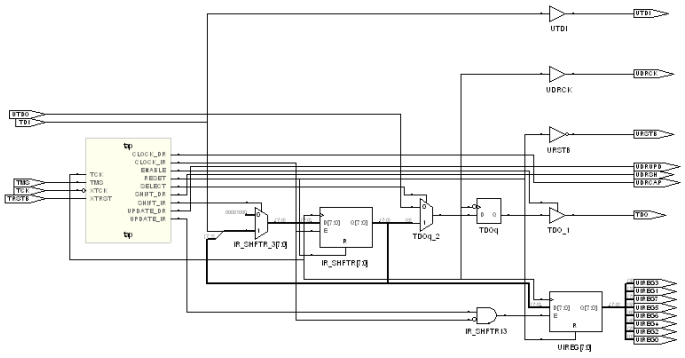

Fig.9 Synthesized result of JTAG

After the device came back we had to test it to verify that it performed as expected. In order to do this, the same vectors used to simulate the overall design after layouts were manually mapped into a new format which suitable for Vanguard ATE system. After applying the test vectors, the response vectors were then output to a file. After some more manual manipulation a simple plot of the testing process was done using Matlab, which is shown in Fig.10. A comparison to the simulation results then was used to show that the device was operating properly.

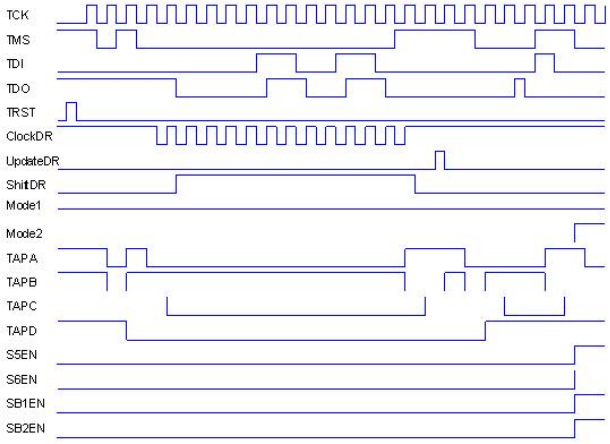

Fig.10 Actual test result of the IP core

IV. Conclusion

The goal of JTAG IP core design and implementation lies in the fact that a common IP soft core will bring more convenience and flexibility in IC design and test. Based on above analysis, a soft core compatible with IEEE 1149.1 standard has been designed and implemented. Some design optimizations on TAP controller and mIscellaneous parts have been made to achieve a better results.The simulation result and practical testing results have been given and the results demonstrate that the design method of IP core is correct.

ACKNOWLEDGEMENT

This work is supported by Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality. (PHR(IHLB)) , Beijing Novel Research Star(2005B01) funded by Ministry of Beijing Science and Technology and Funding No. 20042D0500202 from Beijing Government.

REFERENCE

[1] IEEE Standard 1149.1-1990/ANSI, ``Standard Test Access Port and Boundary-Scan Architecture".

[2] H. Kajitani, H. Sato, H Saito, and S. Oresjo, ``Practical Test Generation with IEEE 1149.1 Boundary-Scan", ATE and Instrumentation Conference 1992, pp. 10-12.

[3] J. Brown, ``Design of a Parallel-to-Serial Bus Converter", Electro 1991 Proceedings, pp. 534-539.

[4] J. Brown & J. Marshall, ``A Second Generation Boundary-Scan Tester", Wescon 1991 Proceedings, pp. 578-582.

[5] National Semiconductor, ``PSC100, Parallel to Serial Converter", 1991.