A Functional Enhancement Methodology to JTAG Controller in Complex SOC

Guo Jian-min

Department of Automation, Xiamen University
Xiamen, 361005, China
e-mail: guojiuge@yahoo.cn

Luo De-lin*

Department of Automation, Xiamen University
Xiamen, 361005, China
e-mail:luodelin602@163.com

***Abstract*—Based on one complex SOC chip, one functional enhancement methodology to standard IEEE P1149.1 JTAG controller is proposed in this paper. With the enhanced features, all test functions including stuck-at scan, at-speed scan, memory BIST and high-speed physical layer tests can be controlled by the JTAG controller besides traditional boundary scan tests, and further on-chip debug features are also integrated in this enhanced JTAG controller. Therefore, the chip costs can be reduced, and the software development and debug can be facilitated with the enhanced JTAG controller.**

***Index Terms*—SOC, JTAG, boudnary scan test, DFT, on-chip debug**

I. INTRODUCTION

With the rapid development of microelectronics technology, the integrated circuits become more complex with more functions and higher performances, but it also becomes more difficult to test the complex chips, so DFT (Design For Testability) circuits must be added to the chips during the design stage to reduce test costs.

Joint Test Action Group (JTAG) proposes the boundary scan test architecture in 1986. This architecture defines the hardware and software for boundary scan tests. It's a milestone event for DFT technology, and this architecture is approved as IEEE P1149.1 protocol by IEEE organization.

It is permitted to extend the JTAG circuits to get more functions besides boundary scan tests [1], and one most popular enhancement is on-chip debug by adding some JTAG instructions and special logics [2].

Based on one communication and multimedia SOC chip, by analyzing the chip architecture and DFT schemes, one functional enhancement methodology to standard IEEE P1149.1 JTAG controller is proposed in this paper. Besides traditional boundary scan tests, this enhanced JTAG controller can also control internal scan tests, memory BIST tests and high-speed physical-layer interface tests, and especially provide the on-chip real-time debug feature to facilitate software development.

II. IEEE P1149.1 JTAG PROTOCOL ANALYSIS

In the JTAG architecture, some boundary scan cells are added between I/O pins and core logic, and these scan cells are linked a boundary scan chain, that is, boundary scan register. The test access port (TAP) controller can control boundary scan chain, so chip pin states can be read or set serially to test the chip and the connections. The boundary scan architecture is composed of several ports as follows [3]:

(1) Test access port (TAP). TAP includes TDI port (test data in), TDO (test data output), TMS (test mode select), TCK (test clock) and TRST (test reset). TDI is used to receive test data and test instruction. TDO is used to output test data. TMS controls test the whole test stages such as register selection, data load and data shift. TCK is the test clock, and test instruction, test data and control signals are synchronized by rising-edge TCK, while output signals are synchronized by falling-edge TCK. TRST is a n optional low-active test reset signal.

(2) TAP controller. TAP controller includes one 16-state finite state machine mainly, and the states are switched between sixteen states according to TCK and TMS. When TAP controller is kept in different state, different control signals will be active to control the boundary scan test stages.

(3) Instruction register and decoder. The JTAG instructions are related with different test modes. The instruction can control how the scan register work and which date register is selected between TDI and TDO path.

(4) Data register. There can be many kinds of data registers, and all of them should be scanned ones. For the data register, boundary-scan register and bypass register are obligatory, while other data registers are optional like device identity register. Besides the registers defined by JTAG standard, user-defined scan registers are also permitted for some especial cases.

The boundary scan architecture is a standardized test mechanism, and it can be extended for other functions besides traditional boundary-scan tests. With user-defined JTAG instructions and data registers, the boundary scan architecture can manage all kinds of test logics inside the chip and mange them flexibly, including internal scan tests, memory built-in self-test and logic built-in self-test, etc. With the extended JTAG interface, all tests can be performed with low-cost testers. This "load slow, run fast" test methodology is a low-cost and high-quality way. The DFT architecture in IBM S/390 is similar to it [4].

Besides traditional boundary scan tests and extended test managements, the JTAG controller can also be expanded further for on-chip debug with new debug-special instructions

*Corresponding author.

978-1-4244-3521-0/09/$25.00 ©2009 IEEE

and logics. When on-chip debug is supported by extended JTAG interface, the target chip can be connected with the host computer with a JTAG cable, then all resources in the target chip can be accessed by the host computer, and the program running course can also be controlled by the host without any disturbances. The on-chip debug is used in the processors mainly. The processor core has to be modified to support on-chip debug besides JTAG extension itself. Generally the processor on-chip debug functions includes internal resources accesses by remote hosts through JTAG interface, instruction breakpoints, data breakpoints, single-step runs and program traces, etc.

III. THE JTAG EXTENSION METHODOLOGY IN COMPLEX SOC

In this part, the chip DFT scheme and on-chip real-time debug requirements will be analyzed firstly, then based on the chip architecture and mentioned-before JTAG protocol analysis, a JTAG extension methodology will be presented. With the extensions, the powerful JTAG controller will be capable of boundary scan tests, test managements and on-chip debug to lower the chip costs and facilitate the software developments.

A. Requirements for On-chip Real-time Debug

The software development and debug become more difficult with increased chip complexity, and an effective method to simplify the debug is highly desired. Usually there are some architecture-related debug features provided by the processors. For example, in MIPS32 architecture processor, the debug is supported in several ways: breakpoint instruction (BREAK) is added to generate an exception, some trap instructions are used to self-trapped into especial exceptions when the conditions are meet, and one pair of control registers (WATCH) are provided to trigger exceptions for specified instruction and data requests. Some special supports in the operating system are required in this kind of debug mechanisms, or certain monitor software is necessary to insert breakpoints, set watch-point, process the exceptions and communicate with the users, etc. Further, additional hardware debug tools like logic analyzer are required possibly.

For complex SOC, system bus or processor outputs are not connected with chip pins, so logic analyzer can't be used to debug it. For software breakpoints or self-trap debug, programs are required to be resident in internal RAM memory, so it will be impractical for those systems based on ROM memory fully. For consumer electronic systems, those debug-special communication interfaces like RS-232 or Ethernet will add the costs and volume. While on-chip debug will solve all above-mentioned problems. One processor or SOC chip integrated with on-chip debug features will be connected with debug host only with a JTAG cable, and the connection diagram is illustrated in figure 1. In this paper, there are so many complex functional components and peripheral IP, it will be very difficult to debug the software with software stubs or chip pins, so on-chip debug feature is highly desired.

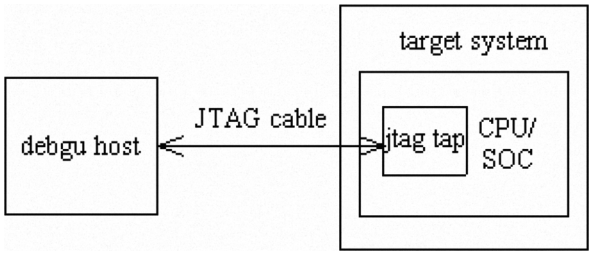

Figure 1. On-chip debug connection diagram

B. Analysis of DFT Scheme

The communication and multimedia SOC chip in this paper is powerful and complex at the same time. It is composed of application processor, digital signal processor, communication and multimedia algorithm components, USB, Ethernet MAC, DDR controller, NAND flash controller, I2C controller, etc. There are more than tem million gates in this chip, and the target frequency is 400-500MHZ with 90nm technology.

It's not easy to test such a complex chip. The planned test scheme in this chip includes internal stuck-at scan test, internal at-speed scan test, memory built-in self-test and some special tests for high-speed physical layer interfaces [7]. For internal stuck-at and at-speed scan tests, because there are so many registers in this chip, they are divided into about eighty scan chains, and it's necessary to manage these scan chains with different test modes reasonably. There are also about one hundred SRAM blocks, so it's necessary to consider how to start BIST tests and output test results. Besides scan tests and BIST tests, some special tests are required to test all kinds of high-speed IO physical layer interfaces.

In order to perform above-mentioned tests, many chip pins are required. Although many of them can share functional pins with functional logics, there still are some test pins that can't be shared. In the other hand, this chip is one IO-limited chip, namely the chip size is decided by chip pin numbers but not chip area. Therefore, if many test pins are added for the tests specially, the chip size and the chip costs will be increased evidently.

C. The Functional Extension of JTAG Controller

Based on the DFT scheme mentioned above, it's necessary to reduce test pins as much as possible to reduce the chip size and chip costs. During the tests, there are three kinds of test signals. The first kind signals are basic control signals including test clock, test reset and test enable, etc. They must be input or output though special chip pins. The second kind signals are the input or output signals for test data like scan-in scan-out, etc. They can be shared with functional chip pins. The third kind of signal includes test mode selection, test start signal and test results, etc. These signals can't be shared with functional chip pins, but if they are input or output through test-special chip pins, then many chip pins will be required, and the chip costs will be increased remarkably with increased chip size and packaging costs. For the third kind test signals, they can also be controlled by extended JTAG controller by adding user-defined JTAG instructions and data registers. With the

extended JTAG controller, this kind of test signals will be input or output through JTAG signals, so no test-special chip pins will be added and the chip cost is reduced.

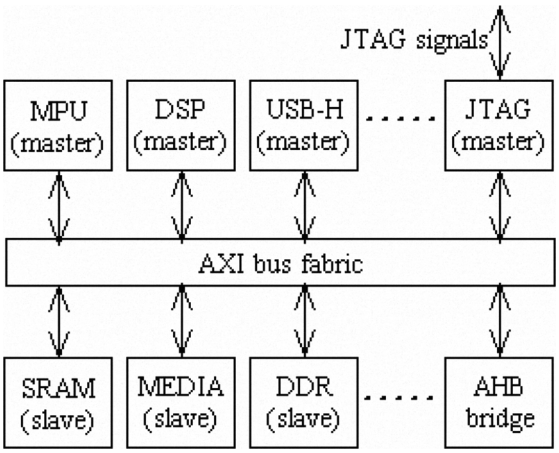

Figure 2. The chip architecture diagram

The SOC chip in this paper is based on AMBA AXI bus fabric. Different internal host components can operate simultaneously. The extended JTAG controller can be connected to the AXI bus fabric to serve as one host component, then the debug host can access all memory space through one JTAG cable, including all addressable registers in all components, internal static memory and external dynamical memory, etc. There fore the programs can be debug in an on-chip real-time way. The new architecture is illustrated in figure 2.

TABLE I. THE EXTENDED JTAG INSTRUCTIONS

Code	Instruction	Function
0x00	EXTEST	Boundary scan external tests
0x01	IDCODE	Select device ID register
0x02	SAMPLE	Sample chip pins
0x03	CONTROL	Read/write control
0x04	ADDRESS	Select read/write adress register
0x05	DATA	Select data register
0x06	CLK_CTL	Select clock control register
0x07	RESET	Select reset register
0x08	TEST_CFG1	Select test management register-1
0x09	TEST_CFG2	Select test management register-2
....	TEST_CFGx	Select test management register-x
0x1f	BYPASS	Select bypass register

TABLE II. THE EXTENDED JTAG DATA REGISTER

Reg. Name	Function	Instructions to access it
BYPASS	Bypass register	BYPASS
ID	Device ID register	IDCODE
BSR	Boundary scan register	EXTEST/SAMPLE, etc
CTRL	Control register	CONTROL
ADDR	Address register	ADDRESS
DATA	Read/write data register	DATA/CONTROL
CLKCTL	Clock control register	CLK_CTRL
RST	Reset register	RESET
TSTCFG1	Test management register-1	TEST_CFG1
TSTCFG2	Test management register-2	TEST_CFG2
TSTCFGx	Test management register-x	TEST_CFGx

In order to implement the test managements and on-chip debug functions mentioned above, the JTAG controller must be enhanced with user-defined JTAG instructions, data registers and corresponding control logics. For JTAG instructions, besides three obligatory instructions (BYPASS, EXTEST, SAMPLE), about twenty user-defined JTAG instructions are added for test managements and on-chip debug, which is listed in table I. For data registers, they are also extended to include address register, read/write data register and test management register, etc. They are listed in table II.

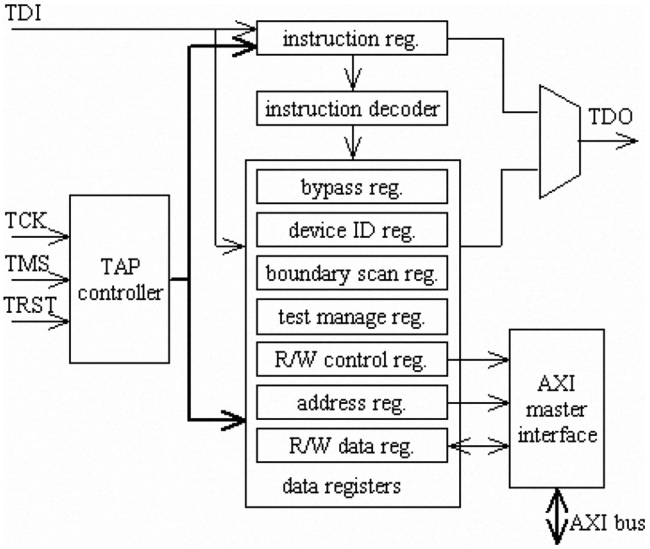

Figure 3. The diagram of extended JTAG controller

In order to connect the extended JTAG controller to AXI bus fabric, one AXI master interface must be added in JTAG controller module, but it's very complex to design a AXI master interface compatible with AXI protocol fully. Considering that it's not necessary to require high bus performance during boundary scan, test management and on-chip debug, the AXI master interface of extended JTAG controller can be simplified to support AXI connection and normal accesses only, and those advanced AXI bus features like out-of-order operations and burst transfers are ignored, so it's will be much simpler to design the AXI master interface.

The diagram of extended JTAG controller is illustrated in figure 3. It is composed of six parts: TAP state machine, instruction register, instruction decoder, data registers, simplified AXI master interface and 2-to-1 multiplexer. By driving TCK, TMS and TDI signals, the tester and debug host machine can control the state transfers inside the JTAG TAP state machine, send JTAG instructions to JTAG controller and read or write the data registers. For test management, the whole control course is rather simple. Firstly the test management instructions are sent to the JTAG controller from the tester, then they are decoded to generate control signals and select test management data registers, further the tester will write control data to test management data register to control the chip tests. The course for on-chip debug is much more complex, and it is exemplified with a read request to certain chip state register

978-1-4244-3521-0/09/$25.00 ©2009 IEEE

from the debug host. Firstly ADDRESS instruction is sent to the JTAG controller from the debug host to select the address register, then the target address is written into the ADDRESS register further, further JTAG read/write control instructions are written into the instruction register to select read/write data register. Now the AXI master interface of JTAG controller starts a read request to the target address through the AXI bus fabric. The data is fetched and stored in the read/write data register, and then the debug host will read it back.

D. The Verification of Extended JTAG Controller

Although the logic counts of the extended JTAG controller is small, the design is rather complex because of so much control logics and a complete function verification is required.

The verification platform is built with VERA language mainly. The JTAG signal transitions for internal TAP state transfers are emulated with a serial of VERA tasks. The code coverage and function coverage are employed to evaluate the verification completeness and ensure the design correctness. Besides, FGPA verification is further employed, and all kinds of real operations from the debug host are tested in the FPGA board.

IV. CONSULSIONS

Based on one complex SOC chip, by analyzing the chip architecture, DFT scheme and on-chip debug requirement, one functional enhancement methodology to standard IEEE P1149.1 JTAG controller is proposed in this paper. With enhanced features, the extended JTAG controller will be multifunctional to include traditional boundary scan tests, test managements and on-chip real-time debug. Therefore, the chip costs can be reduced, and the software development and debug are facilitated.

REFERENCES

[1] Hu Xue-liang, Zhang Chun, Wang Zhi-hua, "A overview of JTAG Technology: Development and Application", Microelectronics, Vol.35, No.6, 2005, pp. 624-630

[2] Wu Hao, Liu Peng, "Implementation of debug interface of DSP processor based on JTAG architecture", Computer Engineering, Vol.31, No.1, 2005, pp. 228-230

[3] IEEE Std P114911 —2001 IEEE standard test access port and boundary2scan architecture [S], 2001

[4] Mary P. Kusko, et al, "Microprocessor Test and Test Tool Methodology for the 500MHz IBM S/390 G5 Chip", In Proceedings of International Test Conference, 1998, pp. 18-23.

[5] EJTAG specification, Revision 2161[M], Mt View: MIPS Technologies, Inc., 2001

[6] MIPS R4000 microprocessor user's manual[M], 2nd ed. California: MIPS Technologies , Inc. , 1994

[7] Hu Yu, Han Yinhe, and Li Xiaowei, "Design-for-Testability and test technologies for System-on-a-Chip", Computer Research And Development, vol.42, no.1, 2005, pp. 153-16.