Indian Journal of Science and Technology, *Vol 8(17), DOI: 10.17485/ijst/2015/v8i17/62664, August 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Verilog Design of Programmable JTAG Controller for Digital VLSI IC’s

Ramesh Bhakthavatchalu, Saranya K. Kannan* and M. Nirmala Devi

Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amritanagar, Coimbatore – 641112, Tamil Nadu, India; rameshb@am.amrita.edu, saranyakannan20@gmail.com, m_nirmala@cb.amrita.edu

Abstract

The objective of this work is to design and implement a custom reconfigurable JTAG controller in Verilog. It can be directly inserted in to a new digital IC designs with little modifications. It is fully compatible with IEEE 1149.1 standards. Additional programmable private instructions can also be added in to the design. A secure access mechanism is provided in to the controller which helps in protecting the system by preventing the un-authorized users from interfering with the system functions. A locking and opening mechanism and a password key based access control were incorporated as part of the JTAG controller module. The controller was configured to fit into different ISCAS’89 digital VLSI benchmark designs and results are analysed. It is observed that as the design size increases the area and power overhead decreases but the number of boundary scan vectors increases. All the designs were written in Verilog and RTL simulations were performed using Cadence NC-Sim Simulator. Cadence Encounter Test Architect 13.1 was used to check the boundary scan flow and analysis. A line graph to depict the power and area overhead is also shown. Complete performance analysis of the ISCAS’89 designs with and without the JTAG controller was performed. The power and area overhead was found to be negligible as the size of the VLSI designs increases.

Keywords: Access Control, Boundary Scan, JTAG Controller, Locking Mechanism, Programmable, Security, TAP

1. Introduction

JTAG/IEEE 1149.1 is a common platform for device, board and system level testing and debugging[1,2]. JTAG port act as the interaction point between the external world and the devices and it also provides access to the internal components for the purpose of circuit debug and configuration. In JTAG, testing and debugging is carried through one of the main hardware component Test Access Port (TAP) which contains four mandatory pins (TDI, TMS, TCK, TDO) and an optional pin for asynchronous reset (TRST). Some of the IEEE standards use 1149.1 infrastructures for configuring Programmable Logic Devices (PLDs)[3].

A TAP/JTAG controller is a module that controls and co-ordinates the operations of the entire test architecture. Similar to a logic design module a TAP controller can also be designs using a Hardware Description Language (HDL). Almost all the current day VLSI IC’s manufactured have TAP/JTAG controller as part of them. Since the TAP controller operations are standardized by IEEE 1149.1 it is possible to create a programmable TAP controller that can be modified and used on several designs[4].

In earlier times JTAG was designed as a test interface standard without any security concern. But with the increasing capability of hardware attackers, more and more side-channels that can compromise the security of the device have been discovered. Improper use of JTAG port is one of the available side channels. Usually JTAG is disabled after initialization of products. But in some device applications, it is kept enabled for the code or firmware updates[2]. This functionality of the JTAG makes scan-based attacks easier and can be used to upload corrupted firmware and read out internal contents of the device[5]. For example, set top box firmware updates occur through the JTAG port. Thus if the JTAG port is insecure,

**Author for correspondence*

unauthorized users can either reprogram part of the system according to their will or steal the (Intellectual Property) IP information of the system. Generally, security problems can occur due to the discrepancy between the expected operation and practical operations of electronic systems. Most of the digital hardware systems contain test interfaces through which the system can be hacked.

Test interfaces are necessary to make the system testable. So testability is a very important property that allows the user to verify correct functionality of the hardware device. Testability measures make the system more testable, and hence increase test coverage, but module may loss its security. So there is always a conflict between security and testability[6].

The objective of this paper is to design and implement a programmable JTAG controller with access control mechanism. This security scheme consists of a locking mechanism with different levels of protection that prevents the unauthorized users from accessing the private and confidential information of a device. Different users have different access levels. Proposed architecture requires only minimal hardware and meets all the specifications of IEEE 1149.1 standards.

Section 2 of this paper discuss about the need for a programmable JTAG controller, JTAG security and related past work. Section 3 gives details about the implementation of proposed JTAG controller design. Section 4 shows results and analysis. Section 5 concludes the paper with performance highlights of the proposed method.

2. JTAG Controller and Security Issues

It is common practice that a separate JTAG controller is designed for every new VLSI design implemented. Since much of the JTAG architecture is uniform to every design implemented it is possible to have a single JTAG controller designed in a Hardware Description Language (HDL) and can be programmed to adapt to various designs. Every product must be tested before it has been introduced into the market. Every VLSI IC manufactured, is to be tested but many times testability enhancement of the design may lead to reduction in system security. SoCs used in various applications contains the IPs which stores confidential information about the devices. SoC designs are usually heterogeneous in nature, with predesigned modules from different vendors embedded in it. Since modules come from different vendors, different testing techniques are used for testing each module. Such testing makes security assurance even more challenging. SoC development presents new security challenges in how to test, configure, and debug the modules within the chip[6]. Data confidentiality and IP protection can be broken through testing[7].

Most of the device debugging operations and uploading of powerful features in the system occurs through the JTAG port. To ensure the security of sensitive information without disturbing the debugging functionality, one have to limit the device access to only authorized users by introducing secure JTAG port[8,9].

In[10], a locking/unlocking mechanism for controlling access to the system is proposed. If the system is locked, then user will not have access to any of the JTAG instruction and if the system gets unlocked, user will have complete access to all JTAG instruction. But it doesn't given any feasible implementation overhead details in terms of area, power and speed.

Debugging ability of the JTAG test structure makes it vulnerable to various kinds of attacks. There are different ways by which an attacker can attack the devices during testing and debugging. It includes controlling the TMS/TCK signals, sniffing and modifying the TDI/TDO signals, and by accessing the secret keys. Different types of JTAG based attacks includes sniff secret data, read-out attack, true vector collection attacks, modify state of authentication part, return false responses to test were discussed in[11]. In[11] a security scheme that employs three standard security primitives which includes a hash function, a stream cipher and a message authentication code was presented.

Security attacks can be passive attacks or active attacks. A passive attack allows learning or making use of information from system without affecting the system resources whereas an active attack may either alter the system resources or affect their operation[12].

Several solutions were being proposed for securing JTAG during debugging and testing. Multilevel Secure JTAG Architecture is proposed in[13] for monitoring and controlling individual scan chain and hence restricts the malicious data being loaded into the JTAG controller. In[14] a Protected JTAG that controls protection level of the device and hence limits the acceptable interaction that takes place through JTAG port during different phase of product development has been discussed. An

Anti-tamper JTAG TAP design using a True Random Number Generator (TRNG) and a Secure Hash (SHA-256) for IC test and on-chip internals is described in[15]. A security enhancement scheme for SoC test access which maintains economy of shared wiring while achieving security benefits of star topology test access wiring is discussed in[16]. A real-life complete software solution for a JTAG security system was proposed in[17]. Multilevel Security for JTAG Architecture using AES encryption/decryption was proposed in[18] but will lead to considerable area, power and speed overhead. In[19] a reconfigurable 2D LFSR was used for generating test patterns for BIST used in SoC type designs. This method is useful for testing SoC with a large numbers of cores within it. This solution does not include TAP controller programmability for testing the multiple cores. A flipped scan chain architecture by inserting an inverter as a security measure to reduce the possible scan based attacks in the ICs was discussed in[20]. Credentials based security system for JTAG test structure is discussed in [21]. In[22] a secure access to reconfigurable scan network is presented and is made possible by extending the TAP with sequence filter.

Different security features has been proposed in the JTAG in various above methods, but most of these security features adds a large area overhead to the device. This makes the feature difficult to implement. This paper suggests an access control mechanism using a lock/open register and authentication scheme. It is shown that this method has less area overhead even for small sized ISCAS'89 designs.

3. Proposed Programmable Secure JTAG Architecture

A programmable controller with security scheme is implemented. A typical JTAG block diagram with a locking mechanism added is shown in the Figure 1. The proposed architecture consist of two PRIVATE instructions: LOCK and OPEN. When the LOCK instructions is active, then TAP controller maps all the instructions except OPEN instruction to a harmless bypass logic until the OPEN instruction with a valid key code is applied. In addition to locking TAP controller, it also provides different levels of access to the system. Once the tap controller gets opened, a security code has to be entered which selects the amount of access that the user can have on the system functions.

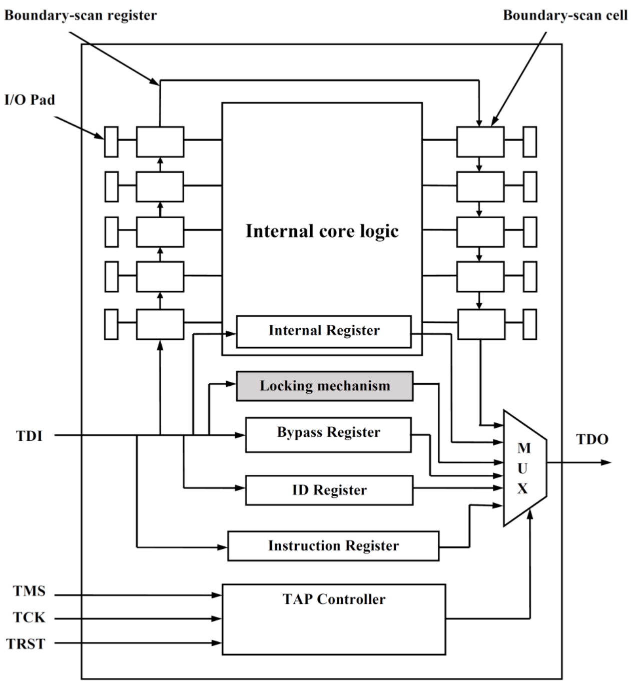

Figure 1. JTAG with locking mechanism.

The internal logic circuit of the security system is shown in Figure 2. It consists of key/lock shift register, key register, lock register, comparator, Private Instruction (PI) register and associated multiplexers. It also includes three level selecting registers (register A, register B, register C) with keys embedded in it. Level select registers will determine the level of access given to the users. How different registers are selected is shown in Table 1.

This paper defines four level of entry for the JTAG port. Protection level includes:

A first level without any protection mechanism and a second level with permission to all operations except

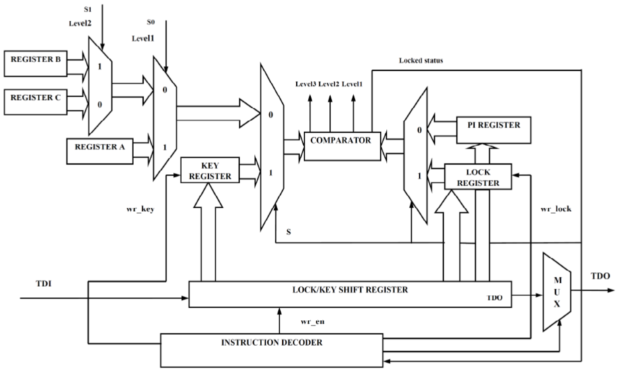

Figure 2. Structure of dual-stage multilevel security system.

Table 1. Register Selection Table

S	S0	S1	Register select
1	0	0	Key register & Lock register
0	1	0	Register A & PI Register
0	0	1	Register B & PI Register
0	0	0	Register C & PI Register

hardware configurable capability and a third level which permits only running the JTAG flow and a fourth level which fully locks the system.

3.1 Steps Involved in Locking the TAP Controller

1. LOCK instruction can be applied during any time when the TAP controller is in the normal active working state.
2. LOCK instruction is entered into the instruction register through TDI. Decoder associated with IR decodes the instruction.
3. Key/lock shift register and lock register are enabled.
4. Lock code is entered into the key/lock shift register through TDI.
5. Lock code is transferred from key/lock shift register to the lock register.
6. Comparator compares the contents of the key register and lock register.
7. The contents are different then locked status fed to the decoder gets activated. Decoder logic maps all instructions except OPEN to bypass instruction.
8. Locked status can be released only by executing OPEN instruction with valid key code.

3.2 Steps Involved in Opening the TAP Controller

1. OPEN instruction can be applied only when the TAP controller is in the locked state.
2. OPEN instruction is entered into the instruction register through TDI. Decoder associated with IR decodes the instruction.
3. Key/lock shift register and key register are enabled.
4. Key code is entered into the key/lock shift register through TDI.
5. Key code is transferred from key/lock shift register to the key register.
6. Comparator compares the contents of the key register and lock register.
7. If the contents are same, then a locked status is fed into the decoder get deactivated.
8. Security code is entered into the key/lock shift register through TDI.
9. Security code is transferred from key/lock shift register to PI register.
10. Contents of PI register are compared with the contents of level select registers (register A, register B, register C).
11. Contents of which level select register matches with that of PI register, corresponding level will be high.
12. Based on the level enable signals, corresponding logic only enabled for the user. User will have that level of access to the circuit logic.
13. Test instruction entered through the TDI can be executed.

It should be noted that the LOCK register should contains a non-zero value at the time of reset. Each operations in the JTAG is controlled by a 16 state finite state machine called TAP controller. Incorporating security features defined two additional states to the TAP controller. State machine of the TAP controller is modified as shown in Figure 3. Two additional states included are

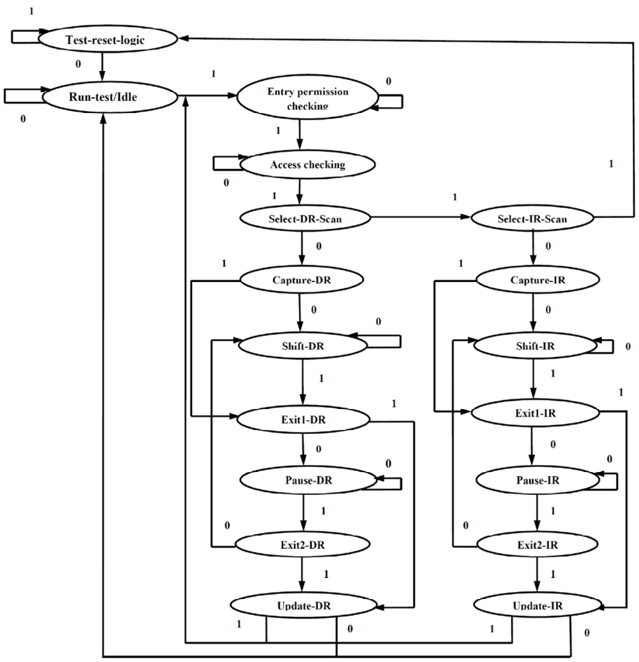

Figure 3. Modified state machine of TAP controller with security features.

entry permission checking state and access checking state. Entry permission checking state checks whether the TAP (JTAG) controller is opened or not by monitoring value on the TMS. Access checking state checks the level of user access to the JTAG and internal logic circuits.

4. Results and Analysis

ISCAS_89 consists of sequential benchmark circuits. Boundary scan/JTAG is applied to many benchmark circuits of ISCAS_89 designs. A TAP controller is written in Verilog HDL and simulated using Model sim RTL simulator. Boundary scan is inserted in the benchmark designs using RTL complier 13.10 (RC) and verified using Encounter test Architect 13.1.100 (ET).

Table 2 shows no. of PI, PO, BC cells, area, power, number of gates, boundary scan vectors after boundary scan insertion to the ’89 designs. Table 3 indicates the results after the addition of the PRIVATE instruction modules. It shows a small increase in area, power, number of gates and boundary scan vectors. Table 4 indicates the increase in area and power after adding the proposed access scheme to ISCAS_89 designs. Security enhancement adds not even 1% to the total area of the circuits. A line graph to show the amount of percentage overhead when adding the security module is shown in Figure 4. Though this method does not employ any encryption or decryption techniques but it has two stage controls to access in to the JTAG structure and has negligible area and power overhead comparing to the size of the VLSI designs.

Table 2. Details of Some ISCAS_89 Designs with Standard Boundary scan

Design	#PI	#PO	#BC cells		Clock	Area (nm^2)	Power (nw)	No of gates	No of FFs	Boundary scan vectors	
			BC_IN	BC_OUT						1149	EXTEST
s344	16	12	9	11	1	250345	8420675.494	246	70	2906	560
s382	10	7	3	6	1	144663	4893473.316	220	60	2747	264
s510	26	8	19	7	1	307979	7020957.158	299	62	2986	720
s820	25	20	18	19	1	413806	10387524.264	377	85	3152	1159
s838	41	2	34	1	1	394706	7422539.019	336	92	3106	1335
s13207	70	153	62	152	1	2130888	88743211.413	2417	984	5777	10304
s38417	35	107	28	106	1	1364363	64702150.167	5651	1828	4597	5139
s35932	42	321	35	320	1	3490889	194823657.285	6962	2427	7905	14168
s38584	45	306	38	304	1	3363049	165495627.671	7489	1944	7707	13759

Table 3. ISCAS_89 Designs with Private Instruction Added and Without Security Scheme

Design	#PI	#PO	#BC cells		Clock	Area (nm^2)	Power (nw)	No of gates	No of FFs	Boundary scan vectors	
			BC_IN	BC_OUT						1149	EXTEST
s344	16	12	9	11	1	250690	8421901.834	354	104	3574	560
s382	10	7	3	6	1	145009	4936538.032	328	94	3415	264
s510	26	8	19	7	1	308324	7041457.062	407	96	3654	720
s820	25	20	18	19	1	414143	10433161.236	483	119	3820	1159
s838	41	2	34	1	1	395051	7510102.973	449	126	3774	1335
s13207	70	153	62	152	1	2131231	89686268.603	2522	1018	6445	10304
s38417	35	107	28	106	1	1364713	65164059.783	5779	1862	5265	5139
s35932	42	321	35	320	1	3491234	209467521.251	7070	2461	8573	14168
s38584	45	306	38	304	1	3363409	167737710.851	7620	1978	8375	13759

Table 4. Performance Overhead After Incorporating JTAG Security Scheme

Design	Without security		With security		Increased %	
	Area (nm^2)	Power (nw)	Area (nm^2)	Power (nw)	Area	Power
s344	250690	8421901.834	252239	8489392.103	0.617894611	0.801366132
s382	145009	4936538.032	146558	5004028.301	1.068209559	1.367157886
s510	308324	7041457.062	309873	7108947.331	0.502393586	0.95847022
s820	414143	10433161.236	415692	10500651.51	0.374025397	0.646882258
s838	395051	7510102.973	396600	7577593.242	0.392101273	0.898659702
s13207	2131231	89686268.603	2132780	89753758.87	0.072681	0.075251507
s38417	1364713	65164059.783	1366262	65231550.05	0.113503718	0.103569773
s35932	3491234	209467521.251	3492783	209535011.5	0.044368266	0.032219921
s38584	3363409	167737710.851	3364958	167805201.1	0.046054464	0.040235597

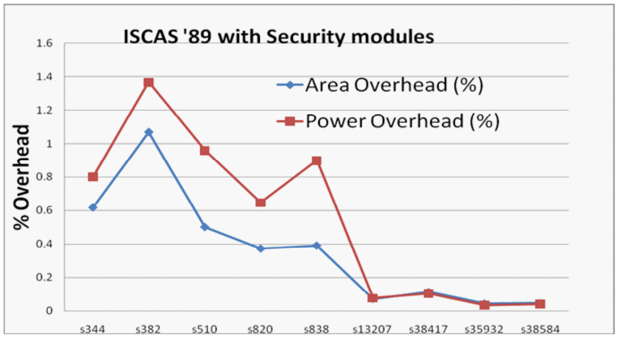

Figure 4. Graph showing the % overhead of the security module.

5. Conclusion

A programmable JTAG controller in Verilog is designed and implemented. This design can be easily employed in many different VLSI designs of varying sizes. A security scheme is also employed. The first stage has a lock and open mechanism where the key to open the TAP controller can be dynamically set by the user locking the system. A second stage three level privilege based access is also implemented. It adds different levels of security to different users. This security scheme adds only small hardware overhead to the designs in comparison to many of the other methods which acquire a large overhead on area, power and speed of the designs. This JTAG controller is equipped to add new PRIVATE instructions and this method is fully conformable with IEEE std. 1149.1.

6. References

1. IEEE standard Test Access Port and Boundary scan Architecture. IEEE std 1149.1–2001. Institute of electrical and Electronics Engineers; 2001 Jul 23. p. 27–36. ISBN: 0738129445.
2. Das A, Rolt JD, Ghosh S, Seys S, Dupuis S, Natale GD, Flottes M-L, Rouzeyre B, Verbauwhede I. Secure JTAG Implementation using Schnorr Protocol. Journal of Electronic Testing: Theory and Applications (JETTA). New York: Springer Science and Business Media; 2013 Apr; 29(2):193–209.
3. IEEE standard for In System Configuration of programmable Devices. IEEE std 1532–2002. Institute of Electrical and Electronics engineers; 2002 Dec 1. p. 3–7. ISBN: 0738135070.
4. Parker K. The Boundary-Scan Handbook. 2nd ed. Kluwer Academic Publishers; 2002. p. 9–40.
5. Rolt JD, Das A, Natale GD, Flottes Marie-Lise, Rouzeyre B, Verbauwhede I. Test Versus Security: Past and Present. IEEE Transactions on Emerging Topics in Computing. 2014 Feb; 2(1):50–62.
6. Tehranipoor M, Wang C. Introduction to Hardware Security and Trust. Chapter 17. Springer Publishing Company; 2011. p. 385–409.
7. Hely D, Rosenfeld K, Karri R. Security challenges During VLSI test. IEEE 9th New Circuits and System Conference; 2011 Jun 26–29. p. 486–9.
8. Jacobson NG. Intest security circuit for boundary scan architecture. 2002 Dec 24. United States Patent 6499124.
9. Pierce L, Tragoudas S. Enhanced Secure Architecture for Joint Action Test Group Systems. IEEE Transactions

on Very Large Scale Integration Systems. 2013 Jul; 21(7):1342–5.

10. Novak F, Biasizzo A. Security Extension for IEEE std 1149.1. Journal of Electronic Testing: Theory and Applications. 2006 Jun; 22(3):301–3.
11. Rosenfeld K, Karri R. Attacks and defenses for JTAG. IEEE Design and Test of computers. 2013 Mar 7; (99):1.
12. Stallings W. Cryptography and Network Security. Chapter 1. Principles and Practices. 5th ed. 2007. p. 15–6.
13. Pierce L, Tragoudas S. Multilevel Secure JTAG Architecture. IEEE 17th International On-Line Symposium; 2011 Jul 13–15. P. 208–9.
14. Buskey RF, Frosik BB. Protected JTAG. Proccedings of International Conference on Parallel processing Workshops; 2006. p. 405–14.
15. Clark CJ. Anti-tamper JTAG TAP designenables DRM to JTAG registers and P1687 on-chip instruments. IEEE Symposium on hardware Oriented Security and Trust(HOST); 2010 Jun. p. 19–24.
16. Rosenfeld K, Karri R. Security-Aware SoC Test Access Mechanism. IEEE 29th VLSI Test Symposium (VTS); 2011 May. p. 100–4.
17. Yoo Sang-Gunn, Park Keun-Young, Kim J. Software Architecture of JTAG Security System. WSEAS Transactions on Systems. 2012 Aug; 11(8):398–408. E-ISSN: 2224–2678.
18. Kumar PA, Kumar PS, Patwa A. JTAG Architecture with Multi Level Security. IOSR Journal of Computer Engineering. 2012 May-Jun; 1(1):54–9. ISSN:2278–0661.
19. Shabaz Md, Patel A, Iyer S, Ravi S, Kittur HM. Design of Reconfigurable 2-D Linear Feedback Shift Register for Built-In-Self-Testing of Multiple System-on-Chip Cores. Indian Journal of Science and Technology. 2015 Jan; 8(S2):207–11.
20. Ramya C, Saravanan S. Rectifying Various Scan-based Attacks on Secure IC's. Indian Journal of Science and Technology. 2015 Jul; 8(13)61856:1–6.
21. Park K, Yoo SG, Kim T, Kim J. JTAG Security System Based on Credentials. Journal of Electronic Testing. 2010 Oct; 26(5):549–57.
22. Baranowski R, Kochte MA, Wunderlich Hans-Joachim. Securing access to reconfigurable scan networks. 22nd Asian Test Symposium (ATS). 2013 Nov 18–21. p. 295–300.