ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

Vol. 4, Issue 2, February 2016

# Embedded Security Considering Hardware Trojan

Bharath K B, Roopa K Swamy

Student, Dept. of E.C.E, SJBIT, Bangalore, India

Assistant Professor, Dept. of E.C.E, SJBIT, Bangalore, India

**ABSTRACT:** The security in today's life is too difficult, since many hacking algorithms have been generated so that there is no security for the data. To provide security for the data, many intelligent have been developed different algorithms to safe the data. One of the techniques is the hardware Trojan. This paper attempts to provide a security for the data using the Hardware Trojan mechanism. The proposed algorithm used here is to provide the data only for the valid users. Questasim tool is used to implement and simulation of the system and Precision RTL tool used for simulation purpose. The algorithm used here provides the data in the secured way.

**KEYWORDS**: Hardware Trojans; Embedded security; password generation; Questasim, Precision RTL.

## I. Introduction

In general, hardware Trojans try to bypass or destroy the three major security concerns of any system by: leaking confidential information and secret keys covertly to the adversary (Confidentiality attack); changing the value of a certain register (Integrity attack); disabling, deranging or destroying the entire hardware or components of it (Availability attack). Traditional Hardware testing strategies cannot effectively detect Trojans because the probability of triggering a Hardware Trojan during functional testing is extremely low. Plus, the small Trojan size with respect to chip overall size reduces the Trojan impact on side channels such as static and dynamic power.

A **hardware Trojan** (HT) is a malicious modification of the circuitry of an integrated circuit. A hardware Trojan is completely characterized by its physical representation and its behavior. The payload of an HT is the entire activity that the Trojan executes when it is triggered. In general, malicious Trojans try to bypass or disable the security fence of a system: It can leak confidential information by radio emission. HTs also could disable, derange or destroy the entire chip or components of it.

The main purpose of this algorithm is to secure the data from the hackers. In this case, the user id is inserted so that the valid user passwords will be stored in the Look Up Table (LUT). When user enters his user id, the password is automatically generated with respect to some function. The password is then compared with the passwords which is stored in the LUT. If the password matches, then that user id is allowed to take the data.

## II. Related work

The Figure 1 shows the block diagram of Hardware Trojan which is used for the authentication purpose. Nowadays, security is the major issues in any field. This figure illustrates that only a valid user is allowed to enter into the operation. Initially, all the passwords are stored in the LUTs (Look Up Table). When user enters his/her ID then it automatically generates the passwords according to some function F(x), where 'x' is the User ID.

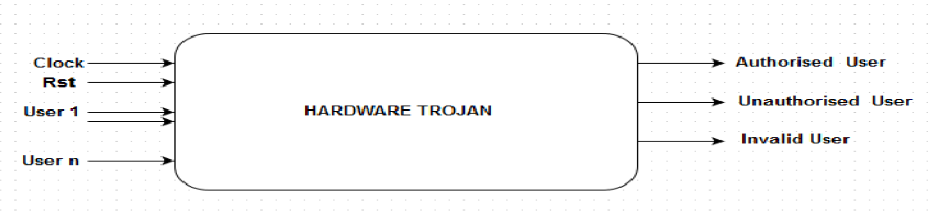


Figure 1: Block Diagram of Hardware Trojan

DOI: 10.15680/IJIRCCE.2016. 0402163

## III. PROPOSED ALGORITHM

Once the password is generated, It is compared with the original password which has been stored in the LUTs. One of the three possible cases will happen, as shown in the figure 2.

- If the password matches, then he is a valid user and he can proceed with the further things.
- If the password does not match, then he is an invalid user.
- If the password matches, and the number of user exceeds more than the requirement, then it enters to the unauthorized state.

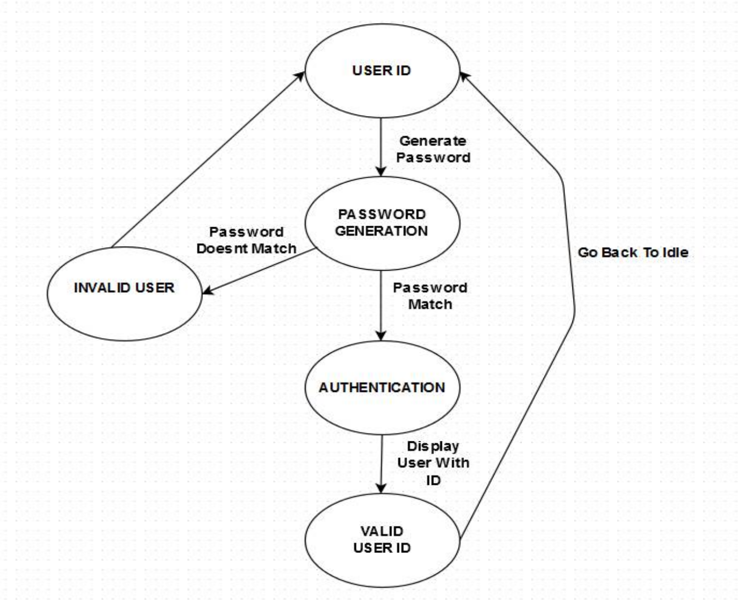


Figure 2: State Diagram of Implemented hardware Trojan.

## IV. PSEUDO CODE

Step 1: At first User id is inserted by the user.
Step 2: According to some function f(x), the password for valid user id is stored in the LUT.
Step 3: After user id is inserted, system generates the password.
Step 4: After the password is generated, it is compared to the password stored in LUT.
Step 5: If the password matches, then he is the valid user and the service is provided to user and go to step 1 else If the password does not match, then he is not a valid user.
Step 6: If more number of user id is entered, more than the valid number of users, then authentication is not done and goes to step 1.
Step 7: End.

## V. SIMULATION RESULTS

The simulation studies involve the Hardware Trojan for an embedded security system as shown in Figure 3. The proposed Hardware Trojan is implemented with Questasim and simulated using Precision RTL tools which are offered by Mentor Graphics Corporation. Questasim is an Event based Simulator. Questasim simulator tool is used to verify the code and to check the response of the Implemented Hardware Trojan, and complete coverage is done with the given test cases. Figure 3 shows the simulation window of the implemented hardware Trojan where the function of the implemented HTs can be verified. By doing this simulation, it will be useful to detect the errors and correcting the errors. From figure 3 to figure 7 deals with the simulation results of Hardware Trojan. From figure 8 to figure 10 deals with the synthesis results.

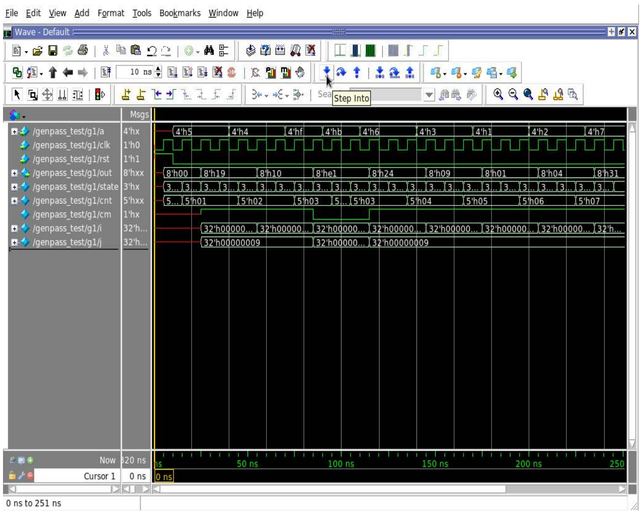


Figure 3: Simulation of HT

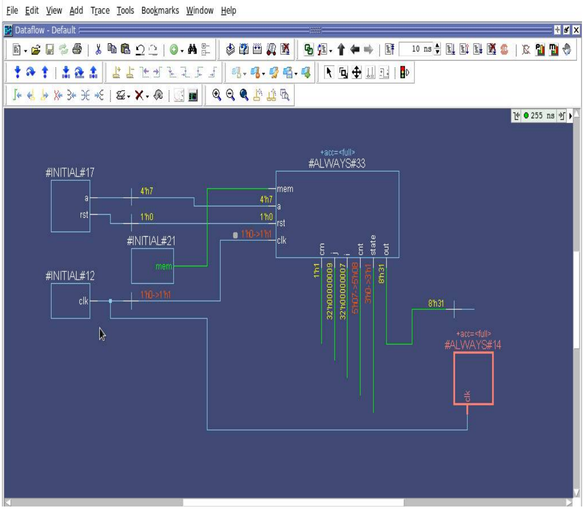


Figure 4: Dataflow Debugging

The Dataflow window allows you to explore the "physical" connectivity of your design; to trace events that propagate through the design; and to identify the cause of unexpected outputs. The window displays processes; signals, nets, and registers; and interconnect which is as shown in Fig 4. Figure 5 represents the schematic view of Hardware Trojan where it specifies the way to check the physical connections. This tool will also generate a State Diagram according to the design what it is implemented. The expected state diagram is as shown in fig 6. Finally fig 7 represents the coverage report of the design. Code Coverage refers that how the test bench is effective considering all the possible cases. A program with high code coverage implies that, code is fully tested and software bugs are less. There are different types of code coverage such as statement coverage, branch coverage, condition coverage and toggle coverage. As shown in fig 7, all coverage is covered resulting the 100% code coverage.

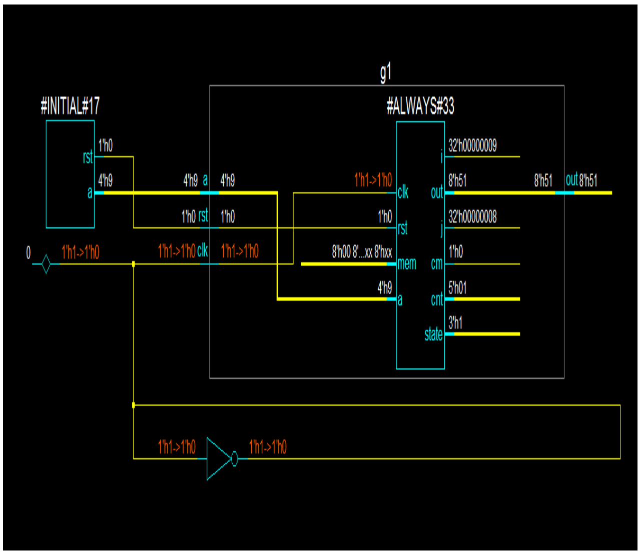


Figure 5: Schematic View

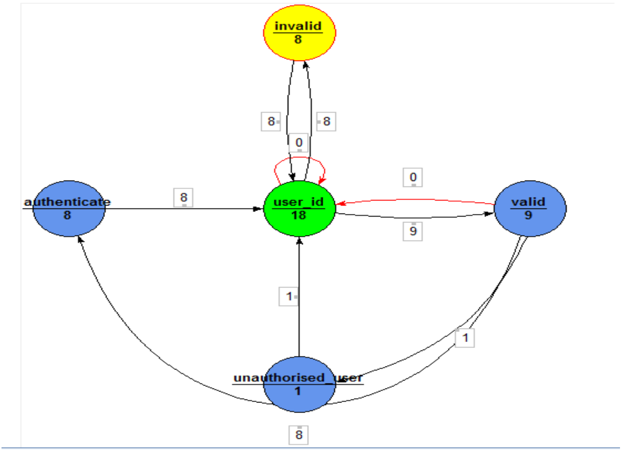
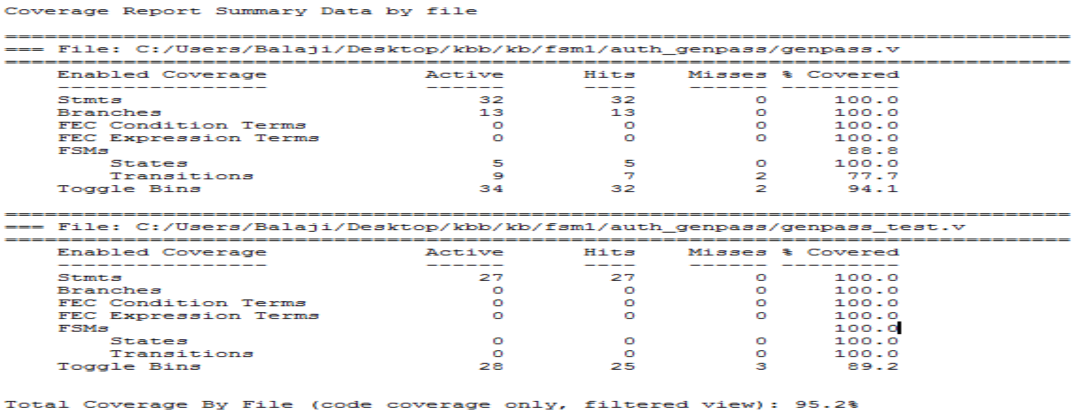


Figure 6: Generated State Machine



```
Coverage Report Summary Data by file

================================================================================
=== File: C:/Users/Balaji/Desktop/kbb/kb/fsm1/auth_genpass/genpass.v
================================================================================
    Enabled Coverage              Active      Hits    Misses % Covered
    ----------------              ------      ----    ------ ---------
    Stmts                             32        32         0     100.0
    Branches                          13        13         0     100.0
    FEC Condition Terms                0         0         0     100.0
    FEC Expression Terms               0         0         0     100.0
    FSMs                                                          88.8
        States                         5         5         0     100.0
        Transitions                    9         7         2      77.7
    Toggle Bins                       34        32         2      94.1

================================================================================
=== File: C:/Users/Balaji/Desktop/kbb/kb/fsm1/auth_genpass/genpass_test.v
================================================================================
    Enabled Coverage              Active      Hits    Misses % Covered
    ----------------              ------      ----    ------ ---------
    Stmts                             27        27         0     100.0
    Branches                           0         0         0     100.0
    FEC Condition Terms                0         0         0     100.0
    FEC Expression Terms               0         0         0     100.0
    FSMs                                                         100.0
        States                         0         0         0     100.0
        Transitions                    0         0         0     100.0
    Toggle Bins                       28        25         3      89.2

Total Coverage By File (code coverage only, filtered view): 95.2%
```

Figure 7: Coverage report.

Figure 8 represents the RTL schematic window of the designed hardware Trojan where RTL refers as Register Transfer Logic which is the basic level to develop any design. Figure 9 represents the Technology schematic is represented in terms of LUTs (Look Up Table) which are designed according to the specific fpga kit that is chosen for the development of the design. Technology schematic will be constructed in such a way that it is same as architecture of the chosen FPGA kit. Figure 10 represents the area report of the hardware Trojan.

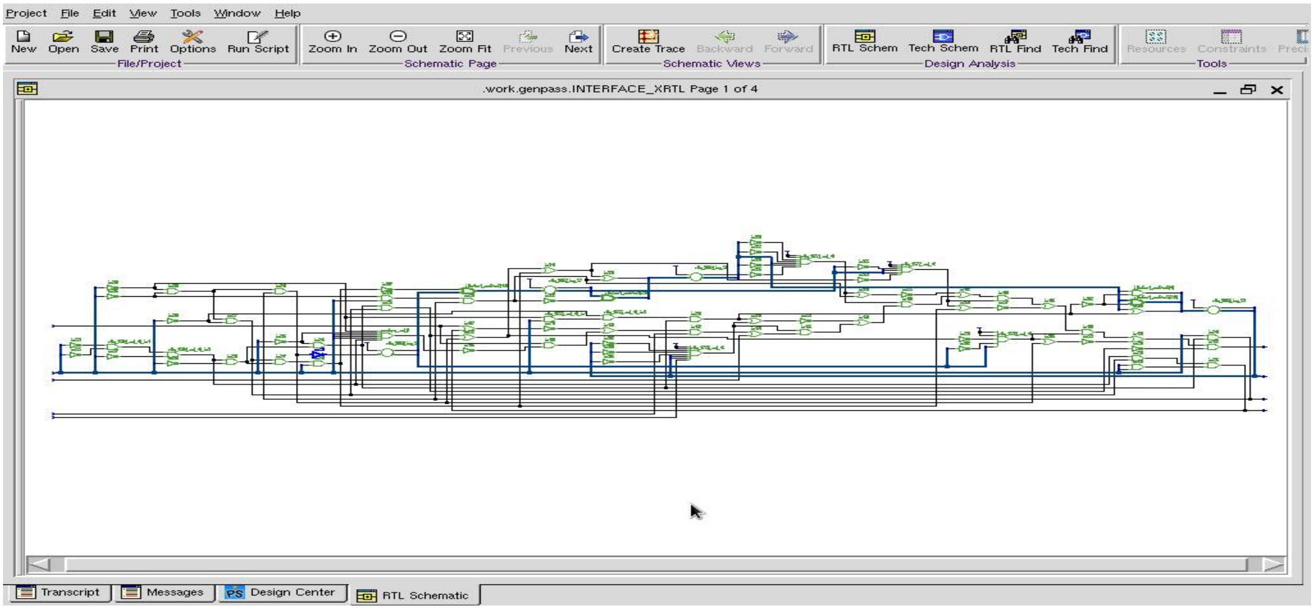


Figure 8: RTL Schematic of Hardware Trojan.

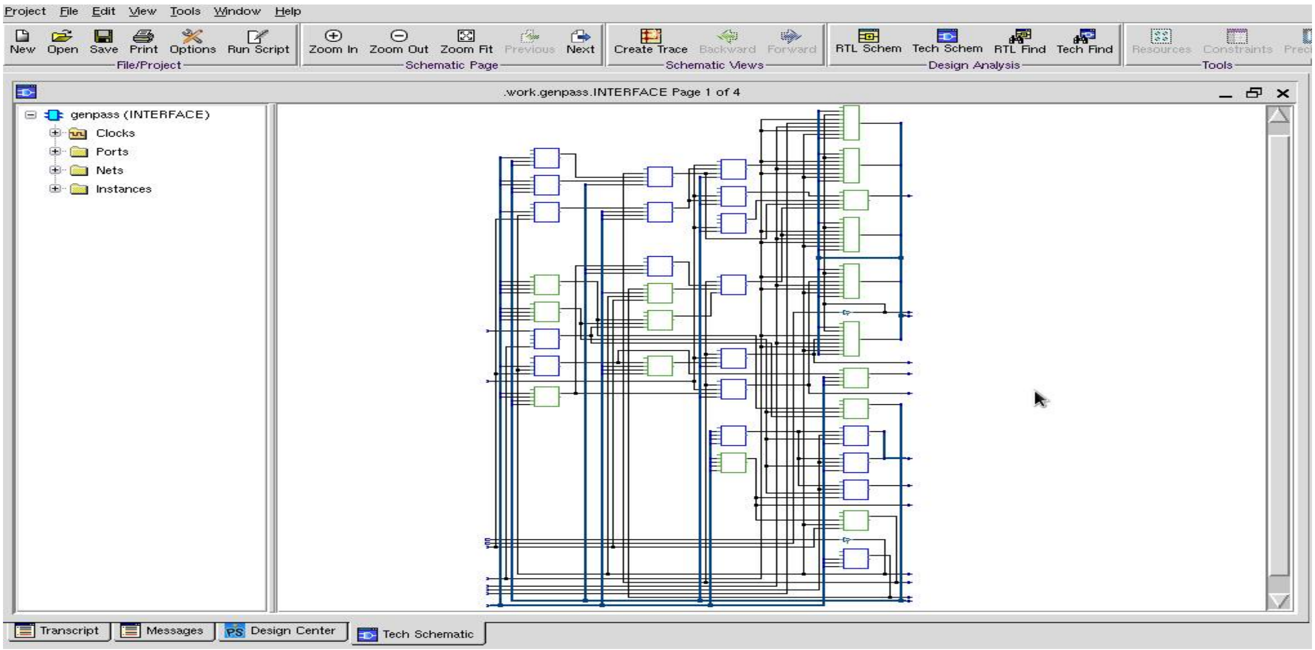
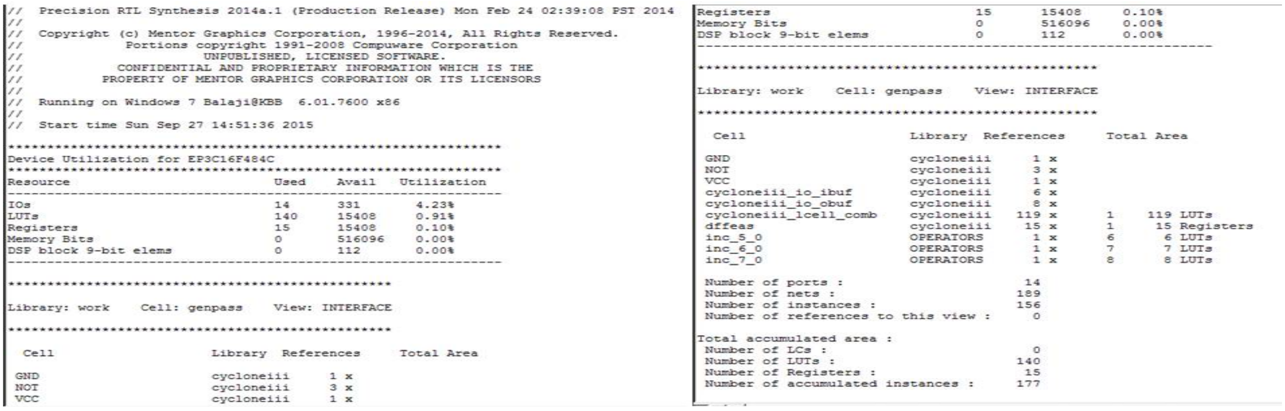

Project File Edit View Tools Window Help
New Open Save Print Options Run Script
File/Project
Zoom In Zoom Out Zoom Fit Previous Next
Schematic Page
Create Trace Backward Forward
Schematic Views
RTL Schem Tech Schem RTL Find Tech Find
Design Analysis
Resources Constraints Prec
Tools
.work.genpass.INTERFACE Page 1 of 4
genpass (INTERFACE)
Clocks
Ports
Nets
Instances
Transcript
Messages
Design Center
Tech Schematic

Figure 9: Technology Schematic.



```
//  Precision RTL Synthesis 2014a.1 (Production Release) Mon Feb 24 02:39:08 PST 2014
//
//  Copyright (c) Mentor Graphics Corporation, 1996-2014, All Rights Reserved.
//             Portions copyright 1991-2008 Compuware Corporation
//                  UNPUBLISHED, LICENSED SOFTWARE.
//            CONFIDENTIAL AND PROPRIETARY INFORMATION WHICH IS THE
//          PROPERTY OF MENTOR GRAPHICS CORPORATION OR ITS LICENSORS
//
//  Running on Windows 7 Balaji@KBB  6.01.7600 x86
//
//  Start time Sun Sep 27 14:51:36 2015

*************************************************************
Device Utilization for EP3C16F484C
*************************************************************
Resource                          Used    Avail   Utilization
-------------------------------------------------------------
IOs                               14      331       4.23%
LUTs                              140     15408     0.91%
Registers                         15      15408     0.10%
Memory Bits                       0       516096    0.00%
DSP block 9-bit elems             0       112       0.00%
-------------------------------------------------------------

***********************************************

Library: work    Cell: genpass    View: INTERFACE

***********************************************

  Cell                     Library  References     Total Area

 GND                       cycloneiii     1 x
 NOT                       cycloneiii     3 x
 VCC                       cycloneiii     1 x
```

```
Registers                         15      15408     0.10%
Memory Bits                       0       516096    0.00%
DSP block 9-bit elems             0       112       0.00%
-------------------------------------------------------------

***********************************************

Library: work    Cell: genpass    View: INTERFACE

***********************************************

  Cell                     Library  References     Total Area

 GND                       cycloneiii     1 x
 NOT                       cycloneiii     3 x
 VCC                       cycloneiii     1 x
 cycloneiii_io_ibuf        cycloneiii     6 x
 cycloneiii_io_obuf        cycloneiii     8 x
 cycloneiii_lcell_comb     cycloneiii   119 x       1    119 LUTs
 dffeas                    cycloneiii    15 x       1     15 Registers
 inc_5_0                   OPERATORS      1 x       6      6 LUTs
 inc_6_0                   OPERATORS      1 x       7      7 LUTs
 inc_7_0                   OPERATORS      1 x       8      8 LUTs

 Number of ports :                       14
 Number of nets :                       189
 Number of instances :                  156
 Number of references to this view :      0

Total accumulated area :
 Number of LCs :                          0
 Number of LUTs :                       140
 Number of Registers :                   15
 Number of accumulated instances :      177
```

Figure 10: Area report of designed HT.

## VI. CONCLUSION AND FUTURE WORK

The simulation results showed that the proposed algorithm looks more secure so that unwanted users will not be given any service so that the data will be safe. This can be implemented for high security using the cryptographic algorithms so that only valid users can use the system and the service are given to them.

## REFERENCES

1. HuafengLiu and HongweiLuo, Liwei Wang "Design of Hardware Trojan Horse Based on Counter" IEEE, ICQR2MSE, P No. 1007 - 1009, 2011.
2. R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware trojan:threats and emerging solutions," In Proceedings of IEEE Int'l WorkshopHardware-Oriented Security and Trust (HOST 09), pp. 166-171, 2009.
3. Chunhua He, Guangzhou, Bo Hou, Liwei Wang, Yunfei En "A novel hardware Trojan detection method based on side-channel analysis and PCA algorithm" IEEE , Reliability, Maintainability and Safety (ICRMS), P No. 1043 – 1046, 2014

4. Yuan Cao; Chip-Hong Chang; Shoushun Chen, “A Cluster-Based Distributed Active Current Sensing Circuit for HardwareTrojan Detection”, Information Forensics and Security, IEEE Transactions on Year: 2014, Volume: 9, Issue: 12, Pages: 2220 - 2231
5. M. Tehranipoor and F. Koushanfar, “A survey of hardware Trojan taxonomy and detection,” IEEE Design and Test of Computers, pp. 10-25, 2010.
6. Errui Zhou; Zhixun Zhao; Shaoqing Li; Lin Ni “Nonlinear analysis for hardware Trojan detection”, Signal Processing, Communications and Computing (ICSPCC), 2015 IEEE International Conference on Year: 2015, P No. 1 – 4
7. He Li; Qiang Liu, “Hardware Trojan detection acceleration based on word-level statistical properties management”, Field-Programmable Technology (FPT), 2014 International Conference on Year: 2014, P No. 153 – 160

8 Kumar, K.S.; Chanamala, R.; Sahoo, S.R.; Mahapatra, K.K. “An improved AES Hardware Trojan benchmark to validate Trojan detection schemes in an ASIC design flow”, VLSI Design and Test (VDAT), 2015 19th International Symposium onYear: 2015, P No. 1 - 6

## BIOGRAPHY

**Bharath K B** is an M.Tech student in the VLSI Design and Embedded System Department, S J B Institute of Technology, Bangalore, India.

**Roopa K Swamy**, specialized in VLSI Design and Embedded System, is an Assistant Professor, Department of Electronics and Communication Engineering, S J B Institute of Technology, Bangalore, India.