JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 20, No. 2 : 343~350 Feb. 2016

온칩버스를 이용한 런타임 하드웨어 트로이 목마 검출 SoC 설계

Kanda Guard · 박승용 · 류광기*

Run-Time Hardware Trojans Detection Using On-Chip Bus for System-on-Chip Design

Guard Kanda · Seungyong Park · Kwangki Ryoo*

Department of Information and Communication Engineering, Hanbat National University, Daejeon 34158, Korea

요 약

본 논문에서는 감염된 IP로부터 악성 공격을 감지하고 예방하기 위한 안전하고 효율적인 온칩버스를 기술한다. 대부분의 상호-연결 시스템(온칩버스)은 모든 데이터와 제어 신호가 밀접하게 연결되어있기 때문에 하드웨어 말웨어 공격에 취약하다. 본 논문에서 제안하는 보안 버스는 개선된 아비터, 어드레스 디코딩, 마스터와 슬레이브 인터페이스로 구성되며, AHB (Advanced High-performance Bus)와 APB(Advance Peripheral Bus)를 이용하여 설계되었다. 또한, 보안 버스는 매 전송마다 아비터가 마스터의 점유율을 확인하고 감염된 마스터와 슬레이브를 관리하는 알고리즘으로 구현하였다. 제안하는 하드웨어는 Xilinx ISE 14.7을 사용하여 설계하였으며, Virtex4 XC4VLX80 FPGA 디바이스가 장착된 HBE-SoC-IPD 테스트 보드를 사용하여 검증하였다. TSMC 0.13um CMOS 표준 셀 라이브러리로 합성한 결과 약 39K개의 게이트로 구현되었으며 최대 동작주파수는 313MHz이다.

ABSTRACT

A secure and effective on-chip bus for detecting and preventing malicious attacks by infected IPs is presented in this paper. Most system inter-connects (on-chip bus) are vulnerable to hardware Trojan (Malware) attack because all data and control signals are routed. A proposed secure bus with modifications in arbitration, address decoding, and wrapping for bus master and slaves is designed using the Advanced High-Performance and Advance Peripheral Bus (AHB and APB Bus). It is implemented with the concept that arbiter checks share of masters and manage infected masters and slaves in every transaction. The proposed hardware is designed with the Xilinx 14.7 ISE and verified using the HBE-SoC-IPD test board equipped with Virtex4 XC4VLX80 FPGA device. The design has a total gate count of 39K at an operating frequency of 313MHz using the 0.13μm TSMC process.

키워드 : AHB 버스 ,하드웨어 트로이 목마, IP, 악성코드, SoC

Key word : AHB Bus, Hardware Trojan Horse , IP, Malware, SoC

Received 31 December 2015, **Revised** 27 January 2016, **Accepted** 03 February 2016

*** Corresponding Author** Kwangki Ryoo (E-mail:kkryoo@gmail.com, Tel: +82-42-821-1710)
Department of Information and Communication Engineering, Hanbat National University, Daejeon 34158, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2016.20.2.343 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. INTRODUCTION

The semiconductor industry has grown over the years and with predictions by Moore's Law, they are becoming even more complex with each passing generation of technology. The issue of hardware security was raised by DARPA (Defense Advanced Research Projects Agency), an agency of the U.S. Defense Department which deals with emerging technologies and national security[1]. An increase in complexity of IC processing is an equal increase in security vulnerability. A Trojan attack on an IC can be planted in the various stages of the design life-cycle. Due to the complexity and reduced size of semiconductors, the design process has become independent and also possible to incorporate third party IPs into designs. The aforementioned factors have increased the security risk of the semiconductor process and have widened the avenue for hardware Trojan attacks. Despite the enormous amounts of effort that has been devoted to software security, relatively little attention has been directed to hardware security in general, and much less to the issue of how to detect and respond to run-time Trojan attacks. Hardware Trojan are usually designed in a way that makes it highly difficult to detect. The Trojans are triggered by specific conditions or signals that rearly occur in an IC which makes it difficult to detect the presence of Hardware malware using random and functional stimuli. In some cases, such as when an attack shuts down the system functionality, the presence of an attack may be relatively easy to detect. Other attacks, such as those involving a Trojan that works in the background to exfiltrate data to an off-chip location may be more difficult to detect. The run-time handling of Trojan attacks can be partitioned into the two major tasks of 1) detection, and 2) response . In this research, the main system interconnect which is the location of interest for Trojan attacks is redesigned with additional circuitry to the conventional bus arbiter, address decoder, slave to master multiplexor and master to slave multiplexor for Trojan attack detection and prevention.

The remainder of this paper is organised as follows, related research in section II, proposed bus architecture in section III, experiment and discussion in section IV and finally conclusion in section V

Ⅱ. RELATED RESEARCH

The previous work relevant to the current paper lies in two areas : works directly addressing hardware Trojans and those generally addressing issues of reliability, self-repair and regeneration of circuits and finally embedded reconfigurable hardware.

2.1. Hardware Trojan

Hardware Trojan is a piece of malicious hardware which is embedded in a system without the knowledge of the end-user or third party IP user as stated in [2]. This malicious hardware can change the functionality of a system or can leak confidential information or even grant entry into the system from an external location without the permission or knowledge of the user. From figure 1, hardware Trojan typically consists of two main features, stated in [2] that is

1. It should have a malicious intent and
2. It should evade detection under conventional

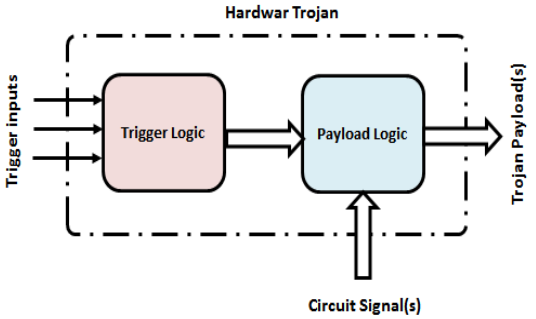

Fig. 1 General Structure of Hardware Trojans

post-manufacturing test/validation process, hence the general structure with two main parts. The triggering logic part and the payload logic part in [3].

A Trojan takes inputs from some internal nets of the main circuit to the Trojan payload and re-stitches some other nets of the main circuit through the payload to modify design functionality. The Trigger Logic is the part of the Trojan that determines the conditions under which the payload Logic sends erroneous data into the main circuit. These parts of the Trojan are the main aspects that cause the Trojan to be fully functional. The payload logic cannot function without the trigger logic and the trigger logic's output is useless if there is nothing to trigger. There are two forms of Trojan, The Sequential Trojan and the Combinational Trojan. Combinational Trojan in figure 2, does not contain any state elements such as flip-flops or latches. It depends on the simultaneous occurrence of a set of rare node conditions for its triggering, while the sequential Trojan shown in figure 3, undergoes a series of state transitions that occur on a system clock edge before triggering a malfunction.

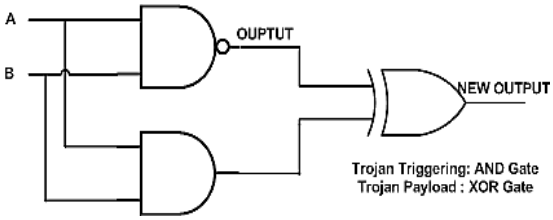

Fig. 2 Combinational Trojans

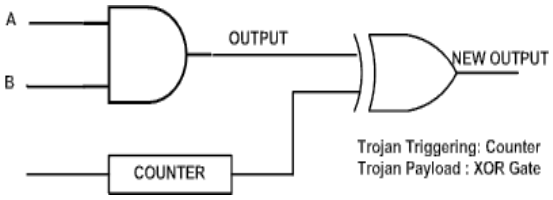

Fig. 3 Sequential Trojans

2.2. Trojan Detection Methodologies

Owing to design complexity and reduction in hardware size of ICs, it can now be designed independently, meaning third-party IPs can be incorporated into designs and designs can also be outsourced and used together with test services and EDA (Electronic Design Automation) tools for a design process. In [4], attacks can happen in any of the untrusted phases or stages in the IC life cycle. The main stages of the IC life cycle are the design and fabrication phases or stages. Each of the stages involved in the life cycle of the IC can be a potential stage for an attack. That is, the IC can be tempered with through the addition, deletion and/or alteration of the circuit structure and also through modification of the steps of the manufacturing process. Objectives of such attacks shown in [5, 6] include

1. Tarnishing the image of a company just to gain competitive advantage in the market
2. Disruption of major national infrastructure by causing malfunction in electronics used in mission-critical systems or leak sensitive information from the chip to illegally access a secure system.

A number of Trojan Detection methodology have been developed over the past years. These methods are either destructive or non-destructive. The destructive is more complex, difficult and time consuming due to processes such as reverse engineering where the whole chip will have to be disintegrated for it to be examined. With the non-destructive one, run-time detection takes place during test-time stage. The Test-time detection include side channel signal analysis and Trojan action analysis.

Agrawal et al worked on the first side channel information used to detect the contribution of Trojan to circuit power consumption in [7]. Trojans typically change a design's parametric characteristics, for example, by degrading the performance, changing power characteristics or introducing reliability problems in the chip. This influences power and/or delay characteristics of wires and gates in the affected circuit and hence side channel signals including timing and power, can be used for Trojan detection. The power signature of Trojan-free (golden circuit) ICs are obtained through the random application of patterns and power measurement. Limited number of ICs are reverse engineered to ensure they are Trojan free. Upon receipt of the reference signature, the same random patterns were performed to the IC under

authentication. The results are then compared.Trojan inserted ICs are seen to consume more power than the Trojan free ICs due to the additional gates. Figure 4 shows an example.

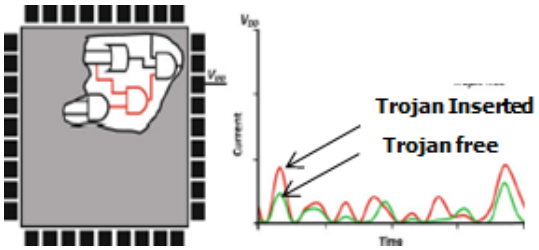

Fig. 4 Trojan Detection Using Power Analysis

There are also Techniques that analyze the impact of Trojans on design performance. Just as additional gates or wiring introduces extra capacitances, any rising or falling on Trojan-inserted paths creates extra time for transition. A Trojan detection scheme using path delay fingerprint which was proposed by Y. Jin and Y. Makris [8] is similar to that carried out by agrawal et al [7]. A circuit has several paths with each representing one part of the entire circuit characteristic. The method measured the delay of several nominated paths on several chips to bring process variations into account. Then after, the chips are reverse engineered to ensure they are genuine, then their measurements are used as a signature. The procedure is repeated on another chip and the results compared to the signature. Any difference indicate the likelihood of a Trojan insertion.

Ⅲ. PROPOSED SECURE BUS ARCHITECTURE

Run-time monitoring of hardware Trojan activities are performed after the IC has been deployed and have several unique advantages. For instance, a Trojan-inserted hardware will perform its intended function for its entire lifetime if the Trojan remains inactive and therefore, a valid way to test this will be to deploy the Trojan infected IC and monitor its behaviour. One of the most common IPs used in most hardware design is an on-chip bus. The system bus, an AMBA AHB bus for this research, is mostly the target of interest for most Trojans [9]. Proposing a secure AHB bus, will help in the run time detection of Trojans.

3.1. Proposed Secure Arbiter

The bus arbiter is the "traffic cop" that determines which master requesting permission to use the bus actually gets to use the bus. Basically the conventional arbiter uses the bus request, HBUSREQx signals and the bus lock, HLOCKx signals which is used by a master that wants to use the bus for an extended period of time, to inform the arbiter of its intention. The arbiter then generates the HGRANT and HMASTER signals which are the grant and master IDs respectively. A malicious master without the intention to perform a burst transaction can keep the bus locked to itself. Not all, a master with desire to perform a burst transaction can keep the bus longer than the expected time. The proposed architecture shown in figure 5 operates with multiple arbitration schemes. A Trojan infected IP can have its HLOCK signal high. The secure arbiter using a threshold value, a value determined by the system designers either by use of a golden model or by analysis to determines how long a master can keep the bus locked. After the bus is granted to a master, the MLD block begins to monitor the particular master ID using the bus and if the master used or activated its HLOCK signal for more than the threshold. A Malicious Lock Detect signal is generated. This signal serves as an enable signal for the Trojan Mask Reg. This register keeps the ID of the current master that locked the bus when the malicious logic detect signal was asserted. This ID is used to mask out any new bus request coming in every time. If a new request matches the master whose ID has been registered in the Trojan Mask Reg., that particular Master is blocked. The multiple arbitration mentioned earlier has two main schemes.

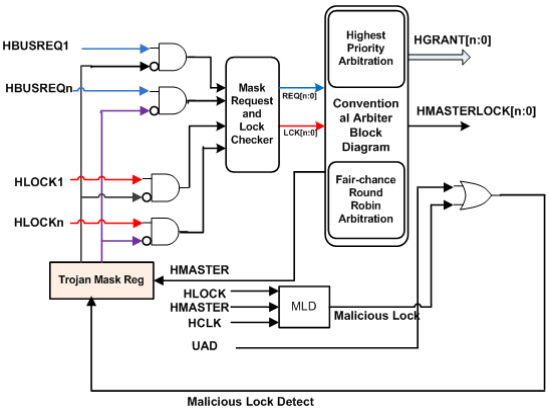

Fig. 5 Proposed Secure Arbiter Block

The fair_chance round robin arbitration allows a master to use the bus again only after all other masters have had their turn to use the bus. That is, if there are 3 masters requesting for the bus usage and master one has the bus, it will not get to use the bus again until master 2 and master 3 have all had their chance to use the bus. This scheme is used together with the highest priority arbitration to curb the issue of master starvation while still keeping order of priority in task execution. If there is an instance where a master which has high priority keeps the bus for longer than the allowed, a threshold, which is also determined by the use of a golden model or an extensive examination is is provided. A master which occupys the bus without locking the bus to it self is checked again by the MLD block. A master that uses the bus longer than this threshold generates the switch arbitration signal which automatically switches the arbitration scheme to fair chance round robin if the current scheme was the highest priority arbitration. The signal UAD is an Unauthorized Access Detection signal that is generated by the address Decoder. The UAD or Malicious Lock signal is used to generate the enable signal(Malicious Lock Detect) to register the Master ID.

3.2. Proposed Secure Address Decoder

The proposed secure address decoder basically uses a simple ROM in the Comparator where each master and its corresponding restricted address for a slave is set by the system designer. The ROM uses the master ID HMASTER as the row address. The slave select HSEL signal, after it has been decoded into column address, is used to read the particular slaves restricted address. The memory for this research is a 5X144 ROM with each location occupying about 36 bits of data.

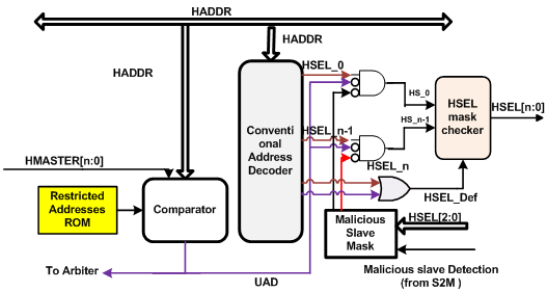

Fig. 6 Proposed Secure Address Decoder

The first 32 bits of the data is for the start address and the final 4 bits of the location is for the augment to the stop address hence the individual addresses are not set if there are so many restricted addresses. An address is decoded and if the transaction is to a restricted address region, UAD is used to mask out the selected slave but activates the default slave to handle the transaction. Malicious slave Mask register keeps the IDs of slaves that have been marked as Trojan Infected. Malicious slave detect signal enables this register and keeps the ID of the slave (HSEL). The ID is checked and used to mask out slaves selected for a transaction. Default slave is again marked to carryout the transaction.

3.3. Proposed Secure Master to Slave Multiplexor

The proposed secure master to slave multiplexor is used to monitor control signals HBURST and HSIZE. simple registers are used to keep the allowable transaction size and number of burst. Master control signals routed to the Master to slave mux is check to find out if any of the signals have been altered by a hardware Trojan prior to getting to the mux. A miss-match will cause the module to replace the erroneous value with the predefined one in the 5 X 3, 5 for the number of slaves

and 3 for the HBURST size ROM.

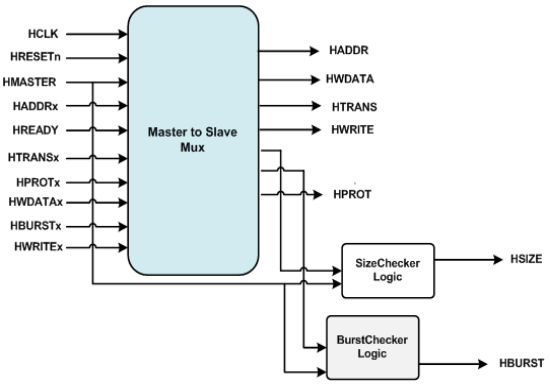

Fig. 7 Proposed Secure Master to Slave Mux

3.4. Proposed Secure Slave to Master Multiplexor

Proposed secure slave to master multiplexor monitors the response signals HRESP and HREADY. These signals are used to delay a transaction by some additional cycles. Trojans may use these signals to cause a denial of service attack during run time. Threshold values for how long a slave can keep its HREADY and HRESP de-asserted are also determined from the golden module. The Mal Resp Detect logic tracks the slave as soon as the HREADY is deactivated and if it gets beyond the threshold value set, it will flag the Malicious Slave Detect Signal. This signal is used as an enable signal for the Malcious Slave Mask register of the secure address decoder. The same monitoring is performed for the HRESP signal.

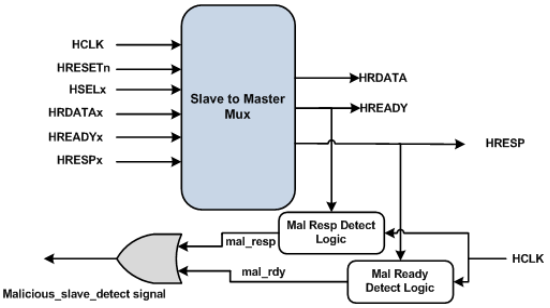

Fig. 8 Proposed Secure Slave to Master Mux

Ⅳ. EXPERIMENTS AND DISCUSSIONS

AMBA AHB was designed and incoporated with the secure features. The test was performed on HBE-SoC-IPD test board equipped with Virtex4 XC4VLX80 FPGA device in figure 9. Dummy threshold values were set for testing the security features and some signals asserted for longer periods of time than that allowed by the threshold.

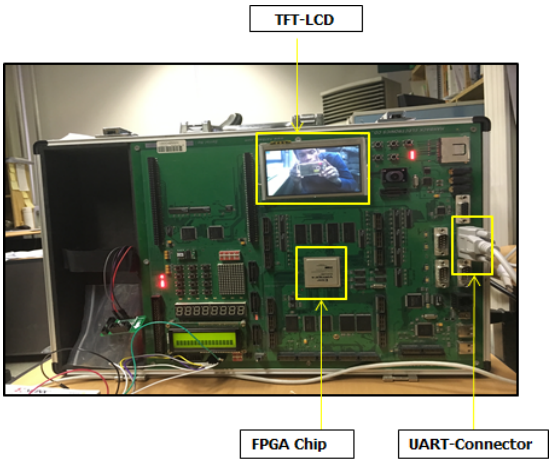

Fig. 9 HBE-SoC-IPD Test Board

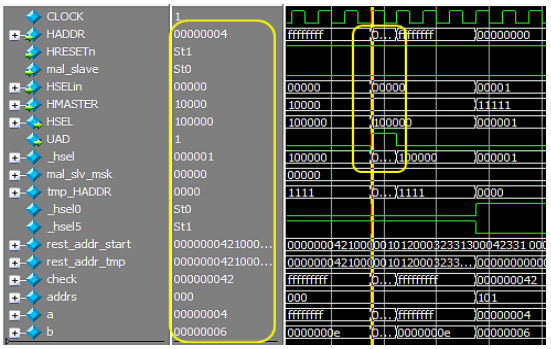

Fig. 10 Simulation of Proposed Secure Address Decoder

The Modelsim SE version 10.1c simulator was also used to perform some simulation as shown in figures 10 and 11. In figure 10 a transaction is performed to slave 1. Slave 1 has the transaction address marked as an unauthorized address. The transaction is then sent to the

default slave and hence _hsel5 is activated which is the default slave. From figure 11, a simulation of the proposed secure slave to master mux is shown. In this simulation, currently selected slave, slave 1 keeps its HREADY signal low for more than 10 clock cycles as specified by the threshold value.

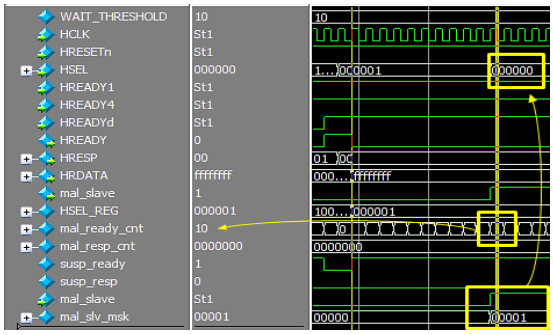

Fig. 11 Simulation of Proposed Secure Slave to Master Mux

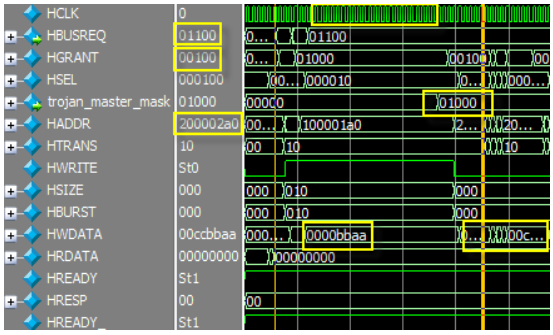

Fig. 12 Timing simulation of overall architecture showing detection of master 2

The Slave is registered in the mal_slv_mak. This register is used to block out a selection of master 1 as shown.The proposed secure bus architectures were designed with the Xilinx 14.7 ISE, using Verilog HDL and synthesized with Synopsys' Design Compiler using TSMC 0.13µm cell library. Table 1 shows the results of the conventional and modified architectures Figure 12 shows an overall simulation where masters 2 and 3 request for the bus with the request 01100. Master 2 is granted the bus because it has higher priority over master 3. Master 2 also keeps the bus locked to itself. it is seen that master 2 keeps the bus locked for 25 clock cycles which is threshold value for this simulation. After this time which can be seen from HCLK, the arbitration mode changes to fair_chance round robin.

Master 2 with HMASTER **01000** is registered as acting malicious in **trojan_master_mask**. The next arbitration grants the bus to master 3 with the **HGRANT** of **00100**. Master 2 is masked out and does not have access to the bus as the grant remains with the bus master 3 who has not locked the bus.

Table. 1 Result comparison between conventional and proposed bus architectures

	Architecture Comparison	
	Conventional	Proposed
Size	Master 5, Slave 5	Master 5 , Slave 5
Process	130nm	130nm
Frequency	300MHz	313MHz
Gate Count	32K	39K

DC synthesis result shows added security features increased gate count from 32K to 39K, which accounts for approximately 22% increase in the area. However, there is a slight improvement in the operating frequency from 300MHz to 313MHz accounting for approximately 4.5% gain in operating frequency .This gain is as a result of shortening the critical path by inserting register slices to break combinational path to sequential.

Ⅴ. CONCLUSION

The paper proposes a secured bus architecture for run-time Trojan detection. IPs embedded with Trojan circuits are masked out of transactions and are not allowed to operate through the secured bus. Transactions made to restricted addresses of a slave are also routed to the default slave instead. Further studies will continue on the design to improve its robustness, and using other signals of an on-chip bus for precise and efficient Trojan detection

ACKNOWLEDGMENTS

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the Global IT Talent support program (IITP-2015-R0134-15-1019) supervised by the IITP (Institute for Information and Communication Technology Promotion).

REFERENCES

[1] DARPA BAA 07 - 24 - solicitations-microsystems technology office [Internet]. Available : http:// www.darpa.mil/mto/solicitations/baa07-24/index.html

[2] M Tehranipoor, H Salmani X. Zhang, *Integrated Circuit Authentication,* Springer publishers, 2014.

[3] Bhunia et al.: "Hardware Trojan Attacks: Threat Analysis and Countermeasures," *in Proceedings of the IEEE,* vol 102, no.8, pp 1229-1247, Aug. 2014.

[4] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *in the IEEE Design & Test of Computers,* vol.27, no.1, pp. 10 – 25, Feb. 2010.

[5] S. Adee, "The hunt for the kill switch," *Spectr. IEEE 45 (5)* May (2008) 34-39

[6] Y. Alkabani and F. Koushanfar, "Consistency-based characterization for IC Trojan detection," *in Proceedings of International Conference on Computer-Aided Design,* pp. 123 – 127, Nov. 2009.

[7] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi and B. Sunar, "Trojan Detection using IC Fingerprinting," *in Proceedings of the Symposium on Security and Privacy,* pp. 296 – 310, May 2007.

[8] Y. Jin and Y. Makris, "Hardware Trojan Detection using Path Delay Fingerprint," *in Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST 2008),* pp. 51 – 57, Jun. 2008.

[9] L.W. Kim and J. D. Villasenor, "A System -On-Chip Bus Architecture for Thwarting Integrated Circuit Trojan Horses" *IEEE transaction on VLSI,* Oct. 2011.

Kanda Guard

2012 Kwame Nkrumah University of Science and Technology, BSc Computer Science
2013 ~ Hanbat National University, Grad. School of Information and Communication Engineering, Masters
※관심분야 : SoC Platform Design and Multimedia CODEC Design, Hardware Software Co-simulation

박승용(Seungyong Park)

2010년 한밭대학교 공과대학 정보통신공학과 공학사
2012년 한밭대학교 정보통신전문대학원 정보통신공학과 공학석사
2012년 ~ 현재 한밭대학교 정보통신전문대학원 정보통신공학과 박사과정
※ 관심분야 : SoC 플랫폼 설계 및 검증, 멀티미디어 코덱 설계

류광기(Kwangki Ryoo)

1986년 한양대학교 공과대학 전자공학과 공학사
1988년 한양대학교 대학원 전자공학과 공학석사
2000년 한양대학교 대학원 전자공학과 공학박사
1991년 ~ 1994년 육군사관학교 교수부 전자공학과 전임강사
2000년 ~ 2002년 ETRI 시스템IC설계팀 선임연구원
2010년 ~ 2011년 Univ of Texas at Dallas 방문교수
2003년 ~ 현재 한밭대학교 공과대학 정보통신공학과 교수
※ 관심분야 : 공학교육, SoC 플랫폼 설계 및 검증, 멀티미디어 코덱 설계