International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 5, Number 4 (2012), pp. 533-543

http://www.irphouse.com

New Logic Module for secured FPGA based system

Mr. Binu K Mathew and Dr. K.P Zachariah

Research Scholar, Anna University of Technology, Coimbatore, Tamil Nadu, India
kbinumathew@gmail.com
Professor, SAINTGITS College of Engineering,
Kottukulam Hills, Pathamuttom, Kottayam, Kerala, India.

Abstract

Field Programmable Gate Arrays (FPGAs) are used as a primary element in various domains of human life like aero-space, automotive, military etc which require them to operate in different types of environments. Security of FPGAs is of major concern as the IP developed after of research can be stolen within hours. Now a days, extensive research is going on, considering various security aspects of FPGAs as the primary concern. The security aspects of FPGAs must be considered so as to make sure that the FPGA system is well protected from all types of possible attacks. This paper explains the cloning of FPGA bit-stream and proposes a method to defend the copying of FPGA bit-streams.

Index terms **- FPGAs, secure devices, volatile devices, SRAM**

INTRODUCTION

Field Programmable Gate Arrays are devices that consist of logic modules and a programmable interconnection network, which are user-programmable, expected to perform user defined logic functions. Along with basic array of logic modules, latest FPGAs have on-chip ADC, dedicated DSP block etc. FPGAs play a vital role in the day to day life of human being like national infra structure, transportation, military and medical areas. Enhanced features of FPGAs point to some security aspects of FPGAs. A design is an intellectual property of a designer, who has invested lot of resources, which should be protected from cloning and unauthorized usage [1], [2]. The IP core developed by a designer should not be cloned by someone else who is not intended to do so. Researchers has proposed several possible attacks against FPGAs, including modifying the hardware, extracting other information through physical side channels, changing the functionality through design tools and cloning of IP core

developed by a manufacturer. [1], [2]. Out of the several threat models proposed so far, the simplest FPGA threat model is the copying of the bit-stream. Several cryptographic algorithms are proposed to reduce the effort of bit-stream copying [7], [8]. Cryptographic algorithms not only improve the degree of security provided to the FPGA based system, this also increases the hardware complexity. This paper proposes a novel Logic Module architecture, which overcomes the threat caused due to cloning of the FPGA bit streams. The rest of the paper is organized as follows- Section 2 discuss the general architecture of a FPGA and different types of FPGAs. Bit-stream coping or cloning of FPGA based system is discussed in Section 3 and design of a new logic module that can be used to improve security of FPGA based system is proposed in Section 4. Results are discussed in Section 5 and conclusions in Section 6.

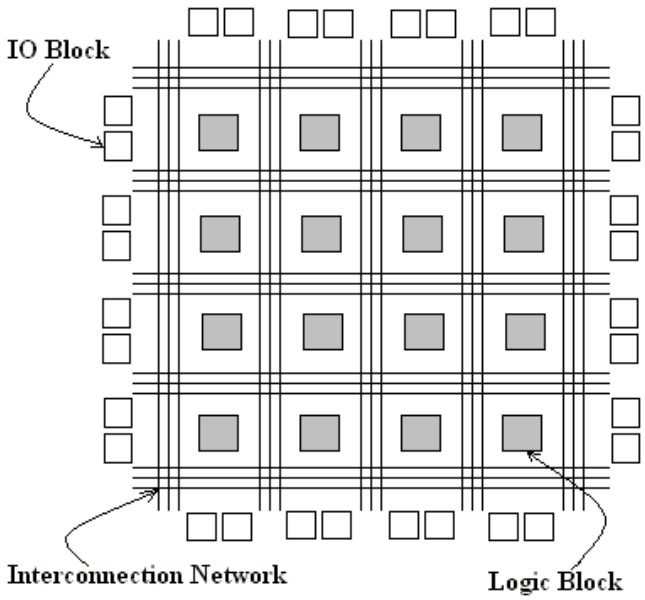

Fig.1. General FPGA Architecture

FIELD PROGRAMMABLE GATE ARRAY

A FPGA is a gate array which can be programmed by the designer in filed after manufacturing, so called as "field-programmable gate array". A hardware description language (HDL) like VHDL or Verilog HDL is used to design the system under consideration and compiled using the software provided by the FPGA vendor. This software converts the design file written using HDL is converted to FPGA compatible bit streams and downloaded from personal computer using a cable. FPGAs can be used to implement any logical function that an Application Specific Integrated Circuit could perform [10]. The ability to change the functionality after shipping, partial re-configuration of the design and the low non-recurring engineering (NRE) costs compared to an ASIC design are the advantages of a FPGA based system for many applications.

FPGAs, which contain an array of logic modules whose functionality, can be determined through multiple programmable configuration bits. Logic blocks configured to realize a specific function, are connected using a set of programmable interconnections. Systems based on FPGAs posses several advantages compared to conventional systems. Ease of prototyping is one of the major advantages of a FPGA based system. A designer can check the functionality of a system under consideration by simply downloading the bit-stream into the FPGA. Based on the programming technique used, FPGAs can be classified as SRAM based, anti fuse based, EPROM based and EEPROM based. Once programmed, contents of anti fuse based FPGAs are made permanent and so called as One Time Programmable (OTP) while others are reprogrammable. SRAM based FPGAs are volatile while EPROM based and EEPROM based FPGAs are non-volatile.

General FPGA Architecture

The general architecture of a FPGA is shown in Fig.1. The architecture of FPGA can be explained as arrays of logic block, which can be interconnected using a programmable interconnect network along with input output block (IO Blocks). The logic block in an FPGA can be as simple as a transistor or as complex as a micro processor, which is capable of implementing various combinational and sequential logic functions [8]-[10]. The logic block in a commercial FPGA is basically multiplexer, Look-up-table or AND-OR array. The periphery of the FPGA consists of I/O blocks, which process signal to and from the FPGA. The routing network in FPGA consists of wire segments of different lengths, which are interconnected using programmable switches. Wires for interconnection are laid in wiring channels or routing channels that run horizontally and vertically through the chip. If long wire segments are used, only a fraction of logic blocks can be used. If small wire segments are used to implement a logic function, more number of interconnections should be used resulting in an increased delay [10]. Different programming technologies are used to implement the programmable switches.

Types of FPGAs

FPGAs must be configured to implement various combinational as well as sequential functions. Based on the configuration technology used, FPGAs can be classified as
SRAM based
Anti-fuse based
EPROM based
EEPROM based

In SRAM based FPGAs which are volatile in nature, SRAM cells are used to store the configuration bits. These cells lose the bits stored whenever power is interrupted. This is the major drawback of SRAM based FPGAs. In the case of anti-fuse based FPGAs, a low resistance permanent link is formed to connect the configuration lines to either logic ‘0’ or logic’1’ [9], [10]. Once programmed these FPGAs hold the bits for ever and cannot be reprogrammed. So these FPGAs are also called as One Time Programmable (OTP) FPGAs. In the case of EPROM based FPGAs, EPROM cells

are used to hold the configuration bits, while in EEPROM based FPGAs, EEPROM cells [9], [10]. EPROM and EEPROM based FPGAs are non-volatile, as EPROM and EEPROM cells are used to store configuration bits. EEPROM based FPGAs are in-system programmable while FPGAs based on EPROM FPGAs are not. SRAM cells are volatile and also vulnerable to various kinds of attacks. On the other hand, these are easily re-programmable, while anti-fuse based FPGAs are not.

CLONING OF FPGA BIT STREAM

FPGAs are used in various important applications like Medical, Defense, Aero-space etc. A FPGA based system, developed after a lot of research and development activities should be made secure in all respects. FPGA based systems are prone to different types of attacks, like cloning of bit streams, modification of bit stream, unauthorized usage of FPGA based system etc[1]-[3]. This section gives an insight to the most common attacks against FPGA based systems i.e. copying of FPGA bit streams. Copying of FPGA bit streams called cloning may lead to the evolution of new systems.

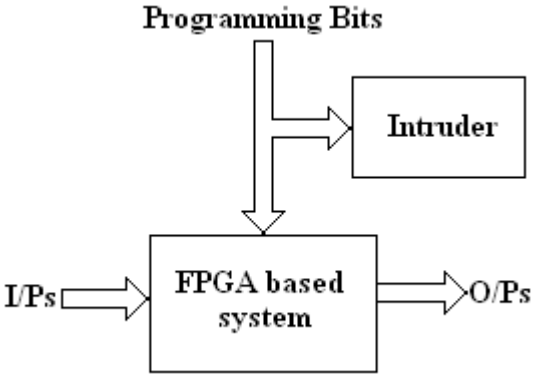

Fig.2 An intruder trying to clone FPGA bit-streams

Field Programmable Gate Arrays are generic devices; i.e. a bit-stream made for one device can be used in any other device of the same family and size [1], [2]. An attacker who knows to program a FPGA can copy bit-streams of the FPGA with the aid of a logic analyzer, and use them to make systems whose performance matches with the original one. The illegal act of copying the bit streams is a major concern for the original system developer as this result in loss of profit. Several work were done to combat the problem of bit-stream coping. Previous work done at University of California, proposes that moats and bridges can be used to isolate the IP core [4], [5].Copying bit-streams results in loss of revenue but also may act as a threat to nation which is using the FPGA based systems. For applications like military and aero-space, were security is a major concern, copying of FPGA bit-streams may become a reason for threat to the nation which is using that system. Several techniques are proposed to overcome the problem of bit-stream copying. A novel technique for comprehensive IP

protection and Digital Rights Management is proposed in [7]. Various aspects for FPGA based system security is studied and explained in [6] and [7]. Security architecture and a set of static and run time primitives to separate cores is explained in detail in [5].

Fig. 2 shows how FPGA bit streams are copied when the FPGA is being programmed. The chance of cloning can be reduced by encryption and decryption. Several encryption algorithms are used in industry to avoid the problem of cloning. On chip decryption must be provided so that, the encrypted bit streams are converted back to its original form. With the help of an encryption key, bit streams are encrypted and sent to the FPGA and encrypted bit streams are converted back to its original form with the help of a decryption key [6]. For systems where security is not a major concern, bit stream encoding can also be used which introduce some degree of security. Various cryptographic algorithms are proposed by several authors, which include DES, Triple DES, AES etc which provides security at the expense of increased hardware complexity [1]. The technique of encryption and decryption can reduce the chance of cloning. As the bit stream is encrypted, before loading into the FPGA, these bit streams should be decrypted using the specified decryption key. Usage of encrypted bit stream or bit stream decrypted with a wrong decryption key will not perform the intended functionality. Even though encryption and decryption reduces the chance of cloning of FPGA bit streams, this increases the cost and complexity of the FPGA based system. This is a potential drawback of encryption and decryption.

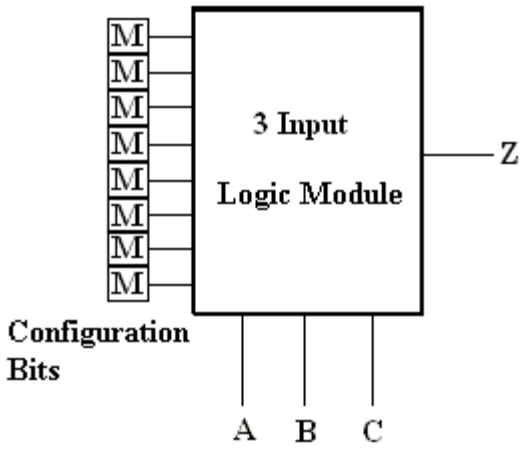

Fig.3 Schematic diagram of a 3-input logic module

PROGRAMMABLE LOGIC MODULE BASED FPGA

As discussed in Section II, there are different types of FPGAs. Different types of FPGAs differ in the way configuration bits are applied to the multiplexer inputs. Basically a logic module is a multiplexer with N select lines and 2^N input lines. To implement a N variable function, a multiplexer with 2^N inputs and N select lines

should be selected as a logic module. Fig. 3 shows the schematic diagram of a 3 input logic module. This logic module can realize a function based on the bits loaded into the configuration memory. In SRAM based FPGAs, configuration bits are stored in the SRAM cells which are volatile in nature. SRAM cells cannot retain its contents at power down. In comparison with the SRAM counterparts, EPROM based FPGAs store the configuration bits into the EPROM cells and are non-volatile in nature. In anti fuse based FPGAs, the input to the logic module is hardwired and cannot be altered, once the design is finalized. As SRAM cells are volatile while other techniques like EPROM, EEPROM and anti fuse based FPGAs are non volatile, SRAM based FPGAs must be loaded with bit stream every time, at power up. So the problem of bit stream cloning is usually found in SRAM based FPGAs and possibility of bit stream cloning is less in other types of FPGAs.

In the previous section, we have discussed cloning of FPGA bit-stream, a common attack a FPGA based system may encounter. Several cryptographic algorithms are currently available which increase the hardware complexity. In this section, author proposes a new logic module which enhances the security of FPGA based system with reduced hardware complexity. In the case of a conventional logic module, the intended functionality is achieved by setting the contents of a SRAM cell connected to the inputs of logic module to either '0' or '1'. The contents of SRAM cells which define the functionality and interconnection pattern forms the FPGA bit streams. In the case of a system which is not secured enough, these bit streams can be cloned to duplicate the contents, resulting in a FPGA based system with less effort. The principle of operation of proposed logic module is to keep the logic module transparent, in the sense; functionality of the logic module depends on the control word entered by the user. A logic module with three inputs can realize several functionalities. For the proposed logic module, the functionality can be altered by changing the control word with a new one. The idea behind this new logic module is to keep the functionality of FPGA based system fully or partially as transparent so that no one knows what the intended functionality is. The customer has to load a control word so as to perform the specified operation. As the system is defined as a look-up table, the functionality is open to the user, i.e the look-up table behaves according to the control word given by the user. Even though this technique may not ensure security against brute force attack, these systems have better security compared to FPGA based systems whose bit streams are not encrypted.

Fig.4 shows a look-up table based design for a 3 variable function. The functionality of the FPGA based system changes in accordance with the control word loaded in to the configuration memory. The complexity of the control word depends up on the number of variables in the function that should be realized. The control word is 8 bit wide if number of variable in the FPGA based system is 3 and 16 bit wide if number of variables is 4. In general the length of control word is 2^N, if number of input variables in the design is 'N'. The degree of security offered by these systems depends on the length of the control word with less hardware overhead compared to FPGA based systems with encrypted bit-streams. Traditional look-up table is a block of RAM memory and this paper proposes multiplexer as a look-up table. Combinational circuits with limited number of inputs can be very easily implemented

using the proposed logic module which offers better security compared to systems designed using conventional logic module. Along with reduced hardware overhead, another advantage of look-up table based design is the ability to reconfigure the system at run time. The system can be reconfigured at run time by changing the control word given to the system. This means that the system not only provides secured design, but also it is reconfigurable at run time, which most of the secured systems does not have. By using a look-up table, the design is open, in the sense, based on the control word given, functionality changes. Control Word based FPGA systems are easily reconfigurable compared to the conventional FPGA based systems. In the case of FPGAs with conventional logic module, the user has to modify the program using software like ModelSim and he should generate a bit stream using the software provided by the software vendor. For systems designed using the Programmable Logic Module, the system behaves in a different way when the control word is changed.

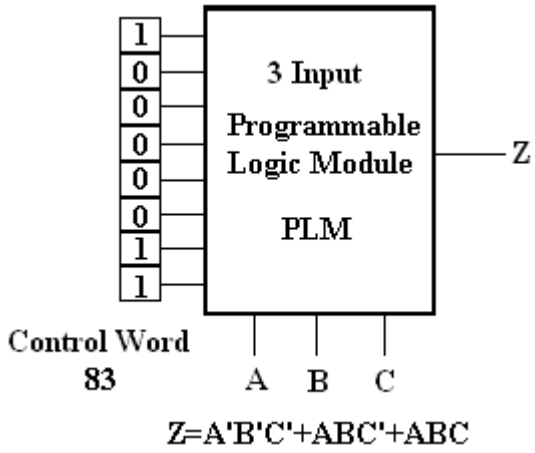

Fig.4 A 3-LUT configured to realize a function

Z=A’B’C’+ABC’+ABC

For a combinational circuit with three variables, there are several possible control words and the device performs the intended function if and only if the proper control word is applied. If a wrong control word is applied, the functionality of the combinational circuit will be different from the function with actual control word. For a multiplexer based look-up table with N input variables, number of input lines is 2^N i.e total number of bits in the control word is 2^N. The possible values for the control word ranges from 00 to 2^N-1. If N=3, the control word is of 8 bits and there are 256 possible combinations for the control words. When number of input variables is more, the elements in the set of values for the control word becomes considerable and it becomes infeasible to find the correct control word from large set of values. The attacker or intruder may not know whether he is using the FPGA based system with its intended functionality or not. Even though degree of security provided by the

FPGA based system with look-up table is low when number of input variables is less, the level of security is in par with the FPGA systems with bit-stream encryption. In systems with bit stream encryption, the encrypted bit stream is send from the remote computer and loaded into the FPGA after decrypting using appropriate decryption key. A designer who doesn't want to reveal design details of the heart of his system can use control word based system. The system performs the intended function when the correct control word is applied; else system behaves in accordance with the applied control word. The proposed technique can be applied either fully or partially to any function so as to make the system secure.

MOTIVATIONAL EXAMPLE

Let us consider a FPGA based system which realizes a one bit full-adder. A one bit full-adder has two outputs, SUM and CARRY. Table 1 shows the truth table of a one bit full adder. Two Programmable Logic Modules are used to realize the functionality of a one bit full adder. Fig. 5 shows the implementation of two Programmable Logic Modules to implement the functionality of a full adder. PLM 1 realizes the SUM function and PLM 2 realizes the CARRY function. The SUM and CARRY functions are implemented using the proposed Programmable Logic Module. The PLM acts as a combinational function based on the control word loaded in to the PLM. To perform the functionality of a full adder, the control words are 96 and E8 for PLM which realize the SUM and CARRY equations respectively. The PLM1 implements the SUM function when 96 is applied and PLM2 realize the CARRY function when PLM2 is loaded with E8 as the control word. If an attacker or intruder loads a wrong control word into the control word register, a function different from the actual function is realized. For example, if a wrong control word like 23 is loaded, the PLM realizes a function which is different from the intended one.

TABLE I-Truth Table of a 1 bit Full Adder

A	**B**	**Cin**	**SUM**	**CARRY**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

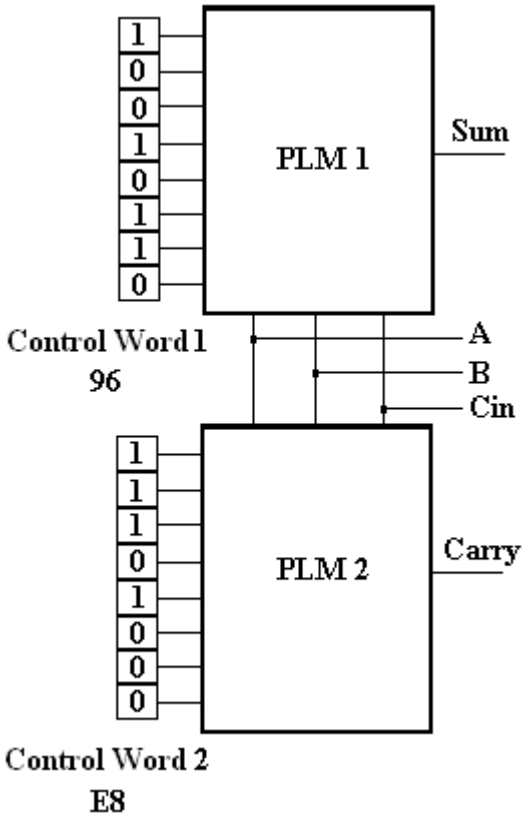

Fig.5 Implementation of one bit full adder using PLM

It may not be easy to implement a function using PLM approach, if number of input variables is more. In that case the function can be split into different child functions and each child function can be implemented using the PLM. For example, if the number of input variables is six, the intended function can be split into two child functions. One PLM can be used to realize one child function and second PLM can realize the second child function. This approach can be extended to functions with N number of inputs. As number of inputs increases, the complexity of the circuit also increases. Fig.6 explains this concept of implementing a function with six variables. Let A, B, C, D, E and F be the input variables.. The function is divided into two child functions, F1 and F2 where F1 is a function of A, B and C and F2 is a function of D, E and F. PLM 1 implements F1 and PLM 2 implements F2 so that the six variable parent function is implemented.

Experimental RESULTS

The proposed Programmable Logic Module (PLM) is implemented in VHDL and the code is synthesized using Xilinx ISE 8.1i. PLM is loaded with different control words and various inputs were applied to check the functionality. A one bit full adder has been implemented on two PLMs, PLM1 is used to implement the SUM function and PLM2 implements the CARRY function. PLM1 and PLM2 realize the SUM and

CARRY function if and only if correct control word is loaded into the PLM. Fig.6 and Fig. 7 shows simulation results for a PLM loaded with control word (CW) 96 and E8 respectively. If a wrong control word is entered, the PLM behaves in a different way. An intruder trying to copy the bit stream will not succeed as there is no bit stream to copy for systems which is fully based on the proposed PLM or the copied bit stream is of no use, as the design is specified only partially.

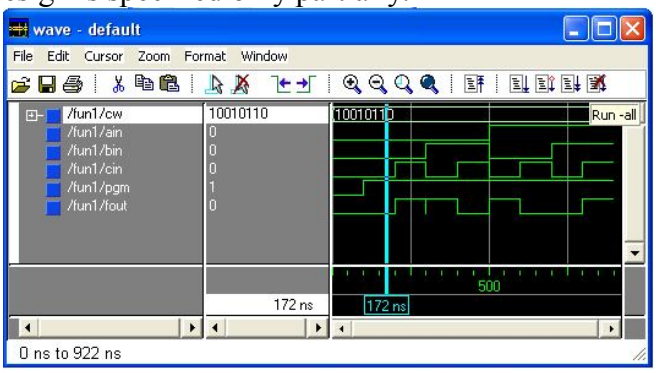

Fig.6. PLM loaded with control word 96

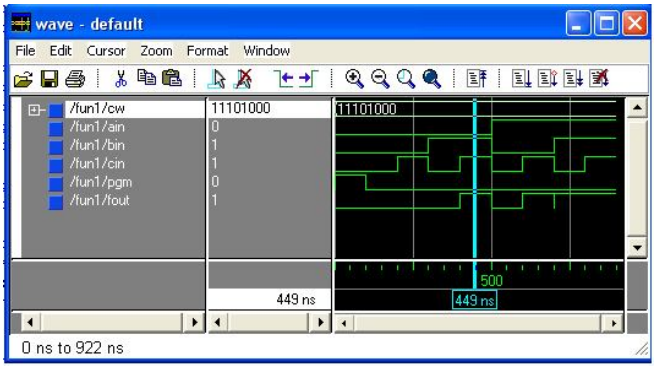

Fig.7. PLM loaded with control word E8

Conclusion

In this paper a new technique is proposed that can be used to provide security for circuits implemented of an FPGA. Even though there are several methods existing to enhance security of FPGA based systems, these methods increase the hardware overhead. Also the conventional FPGA based systems are not run time re-configurable. This new technique not only provides security, but also makes the system run time reconfigurable, which is not possible with the encryption based systems. Bit stream encryption is mainly employed to avoid copying of bit streams when they are loaded into the FPGAs. Copying of encrypted bit streams does not make any sense as these bit streams should be decrypted using appropriate decryption key. In the case of control word based FPGA systems, bit streams are not encrypted and an intruder who copies the bit stream may download this copied bit streams into

his FPGA to perform some function. As the design is based on look-up table, the intruder should load the system with proper control word before proceeding further.

The advantages of this new technique can be summarized as

Provides security with less hardware overhead compared to systems with bit stream encryption.

Reconfigurable at run time by loading the system with a new control word.

Control words can be incorporated for the entire system or to certain blocks of the system based on the degree of security needed for the system.

Bit streams cannot be copied by an intruder as the user inputs the bit stream at run time.

References

[1] Saar Drimer, Volatile FPGA design security – a survey, Computer Lab , University of Cambridge, http://www.cl.cam.ac.uk/~sd410/papers/bsauth.pdf

[2] S. Drimer. Authentication of FPGA bit-streams: why and how. In Applied Reconfigurable Computing, volume 4419 of LNCS, pages 73–84, March 2007. http://www.cl.cam.ac.uk/~sd410/papers/bsauth.pdf

[3] T. Huffmire, S. Prasad, T. Sherwood and R. Kastner, Threats and Challenges in reconfigurable hardware security, www.dtic.mil/cgi-bin/Ge**tTR**Doc?AD=ADA511928

[4] Ted Huffmire, S. Prasad, Tim Sherwood and Ryan Kastner, Designing Secure Systems on reconfigurable Hardware, www.dl.acm.org/citation.cfm?id=1367053

[5] Ted Huffmire, S. Prasad, Tim Sherwood and Ryan Kastner, Managing Security in FPGA-based embedded systems, www.portal.acm.org/citation.cfm?id=1477181

[6] R. J. Anderson, M. Bond, J. Clulow, and S. P. Skorobogatov. Cryptographic processors –a survey. Technical Report 641, University of Cambridge, Computer Laboratory, August 2005. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-641.pdf

[7] J. X Zheng, M. Potkonjak "Securing netlist level FPGA design through exploiting process variation and degradation" www.dl.acm.org/citation.cfm?id=2145716

[8] Jonathan Rose, "Architecture of Field Programmable Gate Arrays", www.ieeexplore.ieee.org/iel1/5/5980/00231340.pdf?arnumber=231340

[9] I. Kuon, R. Tessier and J. Rose, "FPGA Architecture: Survey and Challenges", Foundations and Trends in Electronics Design Automation, Vol. 2, pages 135-253, 2007

[10] S. Brown, and J. Rose, Architecture of FPGAs and CPLDs-A tutorial