ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 5, July 2012

New techniques to enhance FPGA based system security

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

Abstract: **- Field Programmable Gate Arrays (FPGAs) are used as a primary element for various applications like aero-space, automotive, military etc which require them to operate in different types of environments. Security of FPGAs is of concern as these devices are used in various critical applications. Extensive research is going on, considering security aspects of FPGAs as the primary interest. The security aspects of FPGAs must be considered so as to make sure that the FPGA system is well protected from all types of attacks. This paper proposes some threat models and defense models against possible attacks for FPGA based systems.**

Index terms **- LUT, FPGAs, secure devices, non-volatile, SRAM, EPROM, EEPROM, Secure Models, Threat Models**

I. INTRODUCTION

FIELD Programmable Gate Array (FPGA) is a device that consists of logic blocks and an interconnection network, which are user-programmable, expected to perform user defined logic functions.. Along with basic array of logic blocks, 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. Various possible attacks against FPGAs include modifying the hardware, extracting other information through physical side channels, adding unwanted functionality through design tools and coping of intellectual property [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]. Even though cryptographic algorithms improve the degree of security provided to the FPGA based system, this leads to increased complexity of the system. This paper proposes a technique to overcome the threat caused due to cloning of the FPGA based systems. 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 a new technique to enhance security of FPGA based system is proposed in Section 4. Results are discussed in Section 5 and conclusions in Section 6.

Manuscript received June, 2012.

Binu K Mathew, *Research Scholar, Anna University of Technology, (kbinumathew@gmail.com) Coimbatore, Tamil Nadu, India,*

Dr. K.P Zacharia*, Professor, SAINTGITS College of Engineering, Kottukulam Hills, Pathamuttom, Kottayam, Kerala, India.*

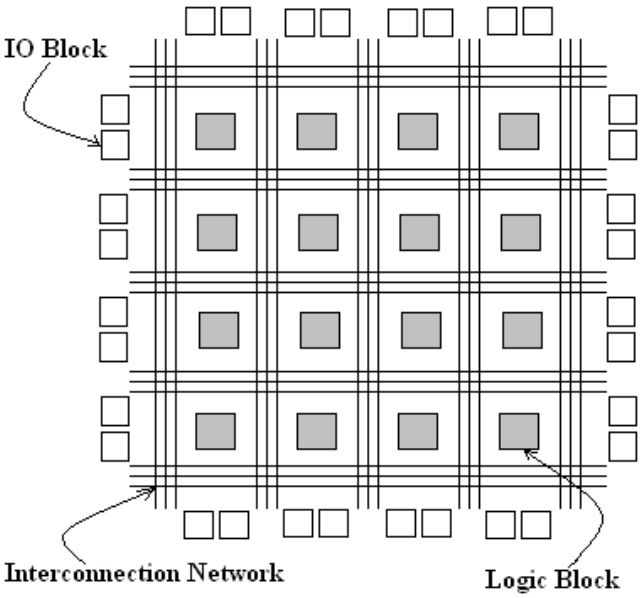

Fig.1. General FPGA Architecture

II. FIELD PROGRAMMABLE GATE ARRAY

A FPGA is an IC designed as customer or designer configurable after manufacturing, so called as "field-programmable". A hardware description language 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 costs compared to an ASIC design are the advantages of a FPGA based system for many applications.

FPGAs, which contain an array of logic blocks 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. FPGA based systems has several advantages compared to conventional systems. The major advantage with a FPGA based is the ease of prototyping. A designer can check the functionality of a system under consideration by simply downloading the bit-stream into the FPGA. FPGAs are widely used in various domains of human life like avionics, medical electronics, communication, signal processing etc. 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.

A. 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.

B. 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, SRAM cells are used to store the configuration bits while in 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]. In EPROM based FPGAs, EPROM cells are used to store the configuration bits, while in EEPROM based FPGAs, EEPROM cells [9], [10]. Fig.2 shows a 3-LUT with memory cells as configuration bits.

To implement a function Z=A'B'C'+ABC'+ABC, the configuration bits should be set as "10000011". FPGAs with SRAM cells are volatile – as SRAM cells cannot retain the values stored when power is interrupted. FPGAs with anti-fuses are one time programmable, fuses once blown cannot be changed to the previous open state. i.e once programmed, configuration bit of FPGAs with anti-fuses cannot be changed. 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 EPROM based FPGAs are not.

III. BIT STREAM COPYING IN FPGAS

Today FPGAs are used in various critical applications like Defence, Aero-space etc. A FPGA based system, developed after a lot of research and development activities should be made secure in all respects. These valuable designs are vulnerable to all possible attacks. Attackers search for all possibilities to crack or tamper these intellectual properties.

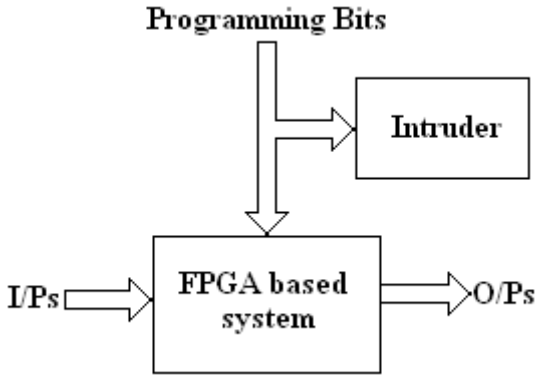

Fig.2 An intruder trying to copy FPGA bit-streams

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 most common attacks against FPGA based systems i.e. copying of FPGA bit streams. Copying of FPGA bit streams may lead to the evolution of new systems which may be a modification of the already existing system resulting in loss and reputation of the company which actually developed the system.

Field Programmable Gate Arrays are generic devices; i.e. a bit-stream made for one device can be used in any other of the same family and size [1], [2]. A competent attacker can copy bit-streams of the FPGA with the aid of a logic analyzer, and use them to make systems which can compete with the original one. The illegal act of copying the bit streams is a major concern for the actual system developer as this result in loss of profit. Several work were done to combat the problem of bit-stream coping. Earlier 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].

FPGA bit streams are copied when the FPGA is being programmed. FPGA bit streams are encrypted to avoid copying of FPGA bit streams. 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].

IV. CONTROL WORD BASED DESIGN

In the previous section, we have discussed a common attack a FPGA based system may encounter, cloning of bit-streams. Several cryptographic algorithms are currently available which increase the hardware complexity. In this section, author proposes a technique that enhances the security of FPGA based system with reduced hardware complexity. The idea is to define the FPGA based system fully or partially as a look-up table. 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.3 shows a 3 input look-up table and 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. 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 reprogrammable compared to the conventional FPGA based systems.

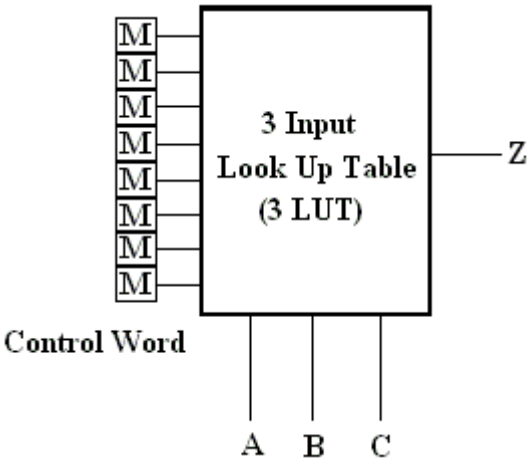

Fig.3 Configuration bits of a 3-input look up table

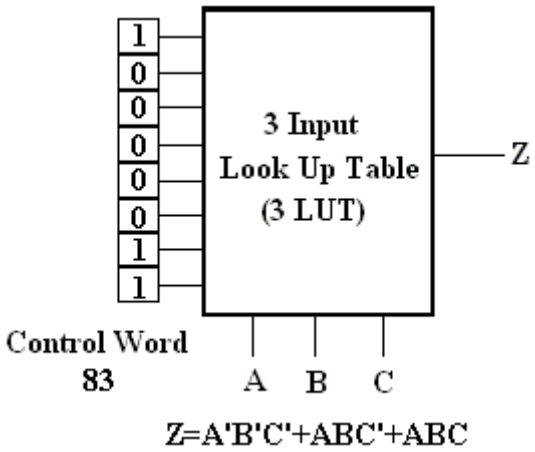

Fig.4 A 3-LUT configured to realize a function Z=A'B'C'+ABC'+ABC

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.

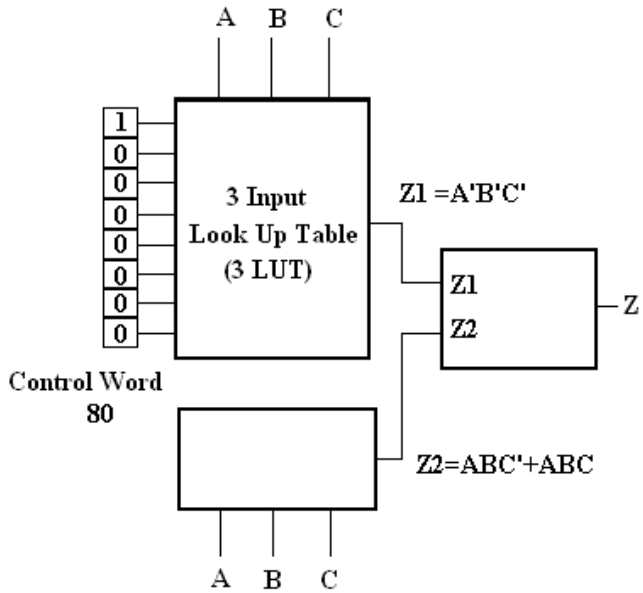

Fig.5 Partial implementation of a function using control word

Let us consider a FPGA based system which realize the function Z=A'B'C'+ABC'+ABC. 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.

Fig.4 shows full implementation of a combinational function Z=A'B'C'+ABC'+ABC using control word. The function is implemented using a look-up table and control word. The look-up table behaves as a combinational function based on the control word applied. To perform the function given above, the control word is 83. The look-up table realizes the function Z=A'B'C'+ABC'+ABC when 83 is applied 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 into the control word register (CWR), the look-up table realizes the function A'BC'+ABC'+ABC.

It may not be feasible to implement a function using look-up table approach, if number of input variables is more. In that case the function can be split into different small functions with less input variables and function under consideration can be implemented using the look-up table approach fully or partially. Fig.5 explains this concept of implementing a function partially using the look-up tables. The function Z=A'B'C'+ABC'+ABC is divided in to two parts, Z1=A'B'C and Z2 = ABC'+ABC and Z1 is implemented using look-up table and Z2 is implemented using the conventional method.

V. Experimental Results

A Programmable Logic Element (PLE) is implemented in VHDL and the code is synthesized using Xilinx ISE 8.1i. PLE is loaded with different control words and various inputs were applied to check the functionality. Control words applied to the PLE are 48, 98 and 24 to implement different functions like ABC'+A'BC, ABC+AB'C+AB'C' and AB'C+A'BC' were implemented. Fig.6 shows simulation results for a PLE loaded with control word (CW) 48. The PLE was loaded with various control words and functionality for various standard functions was verified. It is found that the proposed hardware can be reconfigured at run time so that various functions can be implemented on the fly, which is not possible in the case of FPGAs with conventional Logic Element.

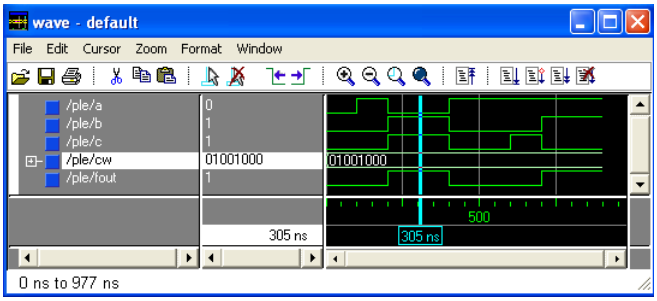

Fig.6. PLE loaded with control word 48

VI. 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.

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/GetTRDoc?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