A Dynamically Reconfigurable Field Programmable Gate Array Hardware Foundation for Security Applications

Samuel J. Stone, Roy Porter, Yong C. Kim
Department of Electrical Engineering
Air Force Institute of Technology
Dayton, OH 45433
samuel.stone@us.af.mil
{yong.kim,roy.porter}@afit.edu

Jason V. Paul
Munitions Directorate
Air Force Research Labs
Eglin AFB, FL
jason.paul@us.af.mil

Abstract

As Field Programmable Gate Arrays (FPGAs) become more widely used, security concerns have been raised regarding FPGA use for cryptographic, sensitive, or proprietary data [7]. Storing or implementing proprietary code and designs on FPGAs could result in the compromise of sensitive information if the FPGA device was physically relinquished or remotely accessible to adversaries seeking to obtain the information. A Hardware Description Language (HDL) FPGA architecture supporting dynamic reconfiguration through granular reconfiguration control is presented for use in security applications. Testing validates the reconfiguration results and compares power usage, timing, and area estimates from a conventional and Dynamically Reconfigurable FPGA (DRFPGA) model.

1. Introduction

Sensitive or proprietary designs on FPGA platforms are vulnerable to exploitation and/or theft due to inherent insecurities in the FPGA design. Although multiple defensive measures have been implemented (and overcome) for FPGA designs, the possibility exists to create a secure design through the implementation of DRFPGA circuits. Using a DRFPGA removes the static attributes from their design; thus, substantially increasing the difficulty of successful reverse-engineering attacks.

Current FPGA architectures do not provide adequate protection against adversarial tampering or reverse-engineering attacks targeting proprietary technology. There

exists ample documentation on the vulnerabilities affecting FPGAs to allow adversaries with relatively modest experience and/or resources to successfully obtain FPGA implemented designs or even alter programs to operate in a manner inconsistent with the intended application [4, 2, 7]. While defensive technologies and dynamic configuration capabilities exist on modern designs, these are fundamentally static in nature and do not address all methods used by malicious adversaries.

This research analyzed current FPGA architectures and proposes a new application of dynamic reconfiguration for an anti-tampering capable architecture fulfilling the need for a method of securing proprietary technology. What differentiates this research effort from previous work is the idea of a custom hardware platform specifically catering to security applications. By creating a model representing current FPGA architectures and a model for the new proposed architecture, tests can be executed to evaluate the feasibility of a custom DRFPGA platform.

2. Research Scope

This research aimed to propose a DRFPGA providing defense against exploitation of FPGA vulnerabilities. Critical FPGA vulnerabilities pose a significant risk to FPGA security. The risk of exploitation is significantly diminished through the implementation of dynamic reconfiguration.

3. Current FPGA Dynamic Reconfiguration

Dynamic reconfiguration on FPGA platforms has been the subject of multiple research efforts in the past decade [5, 1, 3]. However, existing research is primarily targeting optimization of circuit designs through reconfiguration. Additionally, the reconfigurable circuits rely on existing commercial hardware which does not provide the granularity

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

needed to secure FPGA designs using dynamic reconfiguration.

4. DRFPGA Design

A VHDL model was constructed in support of this research. The DRFPGA architecture is similar in structure and capability to existing FPGA platforms such as the Xilinx[®] and Altera[®] FPGAs.

4.1 DRFPGA Functional Design

The VHDL model DRFPGA created for this research is a 4 x 4 array of Logic Blocks (LBs), each capable of implementing relatively simple boolean functions. The 16 block array size was chosen as it allows the implementation of 16-bit functions without unnecessary testing and evaluation overhead associated with a larger circuit. The array has also been constructed to allow multiple 4x4 arrays to be connected together.

The LBs are similar in function to the Logic Elements (LEs) found in Altera[®] FPGAs or the Configurable Logic Blocks (CLBs) found in the Xilinx[®] line of FPGAs. They can be closer compared to the CLBs as they contain eight 4-input, 1-output Look Up Tables (LUTs) per LB. The operational foundation was first developed by Paul et al. as an FPGA supporting retiming [6].

4.2 DRFPGA Programming Design

In a conventional FPGA, programming is primarily performed using the serial Joint Test Action Group (JTAG) serial interface. Although serial in nature, existing FPGA designs have the added ability to bypass portions of the serial chain; thereby, reducing configuration time by bypassing LBs that will not be configured. Modern FPGAs also have the ability to reconfigure parts of the FPGA while the device is in operation (dynamic reconfiguration) but require the reconfigured portion not be in use. These limiting factors do not provide adequate functionality for a security focused dynamic reconfiguration process. To deny adversaries a static target, the actively protected FPGA must be in a constant state of reconfiguration.

The primary difference between the existing hardware platforms and the proposed DRFPGA is the addition of a method of programming individual LBs without using the serial JTAG interface. While current technologies may permit the targeting of small collections of LBs for reconfiguration, the proposed DRFPGA can target any individual LB, and even sub-sections of the LB, for reconfiguration.

The proposed DRFPGA has three programming methods: the conventional serial method, an LB targeting method, and a LUT targeting method. The serial method is

functionally equivalent to the existing FPGA programming hardware in that it provides serial programming functionality with the capability of targeting sub-sets of the DRFPGA LBs for reconfiguration. The smallest target sub-set for reconfiguration is a column of four LBs. The LB and LUT programming methods provide LB or LUT level granularity respectively.

4.3 DRFPGA LB Programming Network

The proposed LB Programming Network (LBPN) alleviates the overhead associated with reconfiguration by providing a means of targeting any single LB for reconfiguration. Programming is performed by transmitting a serial bitstream with an LB address header through the LBPN.

4.4 DRFPGA LUT Programming Network

The LUT Programming Network (LPN) is similar in make-up to the LBPN. In generic terms, they are both composed of a series of routers and serial data lines serving to provide a bitstream of data to a targeted entity. The LPN differs in that it delivers an opcode to autonomous LUTs as opposed to a configuration bitstream. By incorporating basic functions at the LUT level, the data transmission is reduced from 626 bits for the LBPN bitstream to four bits for the LUT opcode.

4.5 DRFPGA LUTs

Each LUT within the DRFPGA design contains functions capable of reconfiguring the LUT contents. By imbedding basic functions at the LUT level, the proposed design alleviates much of the overhead associated with dynamic reconfiguration. In addition, since the entire LB is not being reconfigured, the design retains state information stored within the reconfigured LB.

5. DRFPGA Test and Evaluation

Testing and simulation was performed using the Mentor Graphics *Modelsim*[®] software package. Synthesis was performed to validate the feasibility of the design and measure performance using the Cadence[®] Encounter tool set. The test and evaluation results verified the accurate function of the active decoy reconfiguration and validated the accurate reconfiguration results obtained using the proposed algorithm.

5.1 DRFPGA Test Circuits

Three test circuits were constructed to evaluate the network and the reconfiguration algorithms. Relatively sim-

ple test circuits were chosen for the tests because the bitstreams for the initial circuits (before reconfiguration) were constructed manually. Each circuit was tested for operation using a full-factorial test pattern set before and after the reconfiguration to verify correct operation. The chosen circuits were an 8-Bit Adder, 8-Bit Counter, and 8-Bit Comparator.

5.2. DRFPGA LBPB Tests

The LBPB allows reconfiguration of targeted LBs. The comparator circuit was designated as the test circuit for the LBPB. The initial circuit comprised of two LBs was replaced with a comparator comprised of one LB and a decoy circuit occupying the second LB previously used by the initial circuit. The purpose of implementing the altered configuration is two-fold: the design is now more optimized using less area and power, and the output and input pins have been moved to prevent detection and monitoring.

The full factorial tests for the comparator circuit verify that all test cases performed correctly.

5.3. DRFPGA LPN Tests

Two tests of the LPN were performed: one for the 8-bit adder and another for the 8-bit counter. The LPN differs from the LBPB in that single LUTs within the LB are reconfigured as opposed to the entire LB. A custom algorithm has been developed to demonstrate the use of LUT level functions. The algorithm permits the restructuring of internal LUT contents without affecting the circuit's operation. This is due to the fact that the reconfiguration algorithm is composed of two phases. The initial phase creates an incorrect bitstream which is remedied through completion of the second phase. The circuit operability is not affected because the reconfiguration occurs at a faster clock than the circuit operation.

Both the 8-Bit adder and counter circuit were reconfigured successfully. The counter circuit is more complicated with regards to reconfiguration because of its sequential operation. The LUT reconfiguration algorithm preserves the circuit state, but care must be taken to perform the reconfiguration within the operation clock cycle to avoid glitching and unpredictable output.

6. DRFPGA Performance

The performance aspects of the DRFPGA favor security oriented operation. Security applications perform constant reconfigurations of granular modules; thus, the design requires efficient reconfiguration of small LB clusters. The area, speed, and power efficiency of the DRFPGA is greater than the conventional FPGA model for the target operations.

Synthesis was performed to validate the feasibility of the design and measure performance using the Cadence® Encounter tool set and 90 nm libraries.

6.1. DRFPGA Area

The overall size of the DRFPGA with the reconfiguration networks will be larger than a conventional FPGA given the same LB count. The absolute size and area increase attributable to the added reconfiguration hardware is summarized in Table 1. The increase of 2.10% over the expected area for a commercial Xilinx® FPGA is negligible for the added functionality.

Table 1. Area values for the conventional FPGA and the DRFPGA with the addition of the LBPB.

Area Results		
	Size(μm^2)	Area Increase
Conventional FPGA	468,462	NA
LBPB	9,845	2.10%

6.2. DRFPGA Speed

As discussed in the power usage section, the programming network requires less clock cycles for any operation where fewer than four LBs are reconfigured. The synthesis results obtained through the Cadence® software indicates the LBPB can operate at a higher clock cycle than the conventional FPGA hardware.

The LBPB is more efficient for LB counts that are not multiples of four because the DRFPGA does not activate an entire JTAG chain. The conventional method reconfigures columns of four, and it maintains advantages for any multiple of four due to the addressing overhead imposed by the LBPB. For a larger design, the conventional method's advantage would occur even less frequently. In general, the conventional method incurs less time overhead any time the total number of LBs to be reconfigured is an even multiple of the column size for the FPGA (given column-wise reconfiguration).

6.3. DRFPGA Power

The power measurement considers the conventional FPGA serial programming hardware and the LBPB for the DRFPGA design. The LBPB consumes 8.29% of the power

used by the conventional serial method while the circuit is reconfiguring a single LB. There is, however, a 2.12% increase in the leakage power drawn by the device due to the addition of the programming network. Since the design is a reconfigurable array and will be in a configuration state for most of the operation time, it is the desired outcome.

Because of the additional hardware implemented to allow granular reconfiguration, the overall efficiency per LB is lower than the conventional design for larger reconfigured LB counts (greater than six LBs for this research). Dynamic reconfiguration for security applications typically will not target entire columns of LBs; therefore, the power usage for targeted applications still favors the LBPN and LPN. Additionally, the efficiency of larger LB arrays will favor the LBPN and LPN due to the coarser granularity of the conventional programming selection.

7. Conclusion

The tests and evaluations have verified that the proposed DRFPGA platform yields a reliable method of implementing secure proprietary designs. The LBPN is significantly more efficient in time and power usage for reconfiguring the circuit for most situations related to security minded dynamic reconfiguration than the conventional FPGA's serial method of reconfiguration.

The primary advantage to the DRFPGA platform is its granular real-time reconfiguration capability. Active Protection provided by dynamic reconfiguration and active decoy circuits can mitigate the risk of successful bitstream interception, fault injection/passive analysis, altered bitstream and bitstream readback attacks against FPGA designs by denying adversaries a static target.

8. Future Research

In addition to the proposed security focused applications, the research model also has potential in other fields.

The LB level programming has potential in the realm of biological computing. By controlling the function and programming of the DRFPGA at the LB level, the DRFPGA achieves the desired granularity and the potential for autonomous LBs.

System hardware is not always in an easily accessible location and may be placed in the ocean, remote locations far from civilization, or even in space. The proposed design provides a means to select and implement designs with robust characteristics suitable to the environment and to provide maintainable options for locations for which routine maintenance access is not practical.

Hardware implementations may incur performance penalties with regards to fragmented construction and un-

optimized designs. The framework proposed in this research can be used to perform defragmentation and optimization when coupled with a control process.

9. Summary

The data collected on the DRFPGA operation confirms the platform provides substantial benefits over use of the serial JTAG programming method when numerous low-LB count reconfigurations are required. Implementation of the methods provided alleviate substantial risks posed to FPGA designs. The current DRFPGA design requires significant user interaction to perform the reconfigurations. In the future, reconfiguration commands issued through the VHDL test bench files must be handled by on-board computational resources.

Although this research focused on the DRFPGA for security applications, it also provides a conceptual framework supporting a variety of other applications. The flexibility of the design comes in its dynamic reconfiguration capabilities as well as the granular programming capabilities. Any design requiring dynamic reconfiguration can benefit from the proposed design methodologies.

References

- [1] S. Dueck and W. Kinsner. Netlist partitioning for fpga-based run-time reconfiguration. In *IEEE Canadian Conference on Electrical and Computer Engineering - CCECE02*, volume 2, pages 584–590, 2002.
- [2] D. N. M. T. H. Bar-El, H. Choukri and C. The sorcerer's apprentice guide to fault attacks. *Proceedings of the IEEE*, 94(2):370–382, February 2006.
- [3] Y.-C. Jiang. Minimum communication cost approaches for dynamically reconfigurable fpga. *Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on*, 1:1–4, 13–14 July 2007.
- [4] Z. Kamawal. Experimental power analysis attacks on an fpga. Oregon State University ECE679 Project, 2004.
- [5] W.-K. Mak and E. F. Y. Young. Temporal logic replication for dynamically reconfigurable fpga partitioning. In *ISPD '02: Proceedings of the 2002 international symposium on Physical design*, pages 190–195, New York, NY, USA, 2002. ACM.
- [6] J. Paul, S. Stone, Y. Kim, and R. Bennington. A method and architecture for real-time polymorphic reconfiguration. In *ICFPT07: International Conference on Field-Programmable Technology 2007*, Kokurakita, Kitakyushu, Japan, December 2007.
- [7] T. Wollinger and C. Paar. How secure are fpgas in cryptographic applications. Report 2003/119, German Federal Office for Information Security (BSI)., 2003.