Primitive Instantiation for Speed-Area Efficient Architecture Design of Cellular Automata based Mageto Logic on FPGA with Built-In Testability

Ayan Palchaudhuri and Anindya Sundar Dhar
Department of Electronics & Electrical Communication Engineering,
Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, India-721302
E-mail: ayanpalchaudhuri@gmail.com, asd@ece.iitkgp.ac.in

I. Introduction

Random number generation is integral to information security in IoT based cyber-physical systems. One such recent scheme of random number generation was proposed in [1] called Mageto, which is based on the design principles of cellular automata (CA). CA are characterized as finite state machines (FSMs) which evolve in discrete time steps. In hardware, their architecture remains modular and cascadable, which is ideal for amicable mapping onto FPGA primitives, leading to a speed-area efficient realization [2], [3]. Fault localization and in-system testing of FPGAs are now assuming substantial importance [4]. Exploring the utilization ratio of the configured primitives is often essential for supplementing an original FPGA implementation with testable logic without appreciable hardware overhead and critical path delay, by adopting careful optimization practices. Primitive instantiation is one such technique to directly instantiate an FPGA primitive into a design through appropriate logic configuration. We believe that VLSI implementation of Mageto has never been discussed before, which we choose to address in this paper. Our proposed primitive instantiation based architectures for Mageto, whose design description generation has been automated, outperform the relatively high level behavioral implementations with respect to area (logic slices) and speed.

II. VLSI Implementation of Mageto

CA are typically characterized by a single or 2-3 registers, but Mageto assumes a slightly modified CA structure which involves multiple registers. Its operation is divided into two distinct sub-stages: *cell initialization* and *cell update*, as illustrated in [1]. The *initialization* phase involves loading 128 Parallel In Parallel Out (PIPO) registers with defined seed values, where each register comprises of 32 or 64 FFs. Every four adjacently indexed CA registers $A_j, A_{j+1}, A_{j+2}, A_{j+3}$ out of the 128 registers participate in determining the next state (NS) of A_j in every clock cycle. A_j and A_{j+1} drives the inputs of a greater-than comparator, XOR and adder logic. In addition to FFs, our proposed Mageto CA system is modeled using a Block RAM (BRAM) unit to model the 128 pairs of registers, to drastically reduce the overall FF count and target towards an efficient speed-area implementation. The modeling of the algorithm in hardware is done in a manner to exploit the hardwired fabric of the FPGA for faster signal propagation. Scan path was inserted to aid testability through suitable multiplexing arrangement within the underutilized pockets of the configured primitives. For pure combinational subcircuits in the Mageto CA system, we supplement with self dual complements which when XORed with the original function, alternates for alternating outputs [4].

III. Results and Discussions

The architectures were implemented on Xilinx Virtex-7 FPGA using ISE 14.7 design environment. The post route implementation results have been reported for all cases. We have compared the results of our proposed architectures realized through primitive instantiation with another implementation where the adopted design description is relatively behavioral in nature. To the best of our knowledge, no existing literature is available to compare our proposed VLSI architecture and implementation results. The speed of operation of our proposed implementations is on an average 14–16% higher than their behavioral counterparts for the original Mageto implementation, and 24–31% higher for the testable Mageto implementation. Behavioral mode of hardware realization for testable Mageto unit cannot coalesce the scan logic into the underutilized primitives of the sole Mageto implementation which increases the delay. The design automation assisting generation of circuit description files to be received as input by the CAD tool for Mageto implementation was carried out, thereby making it a commercially viable approach to realize other architectures by adopting a similar design philosophy.

References

[1] R. Vuckovac, "Secure and Computationally Efficient Cryptographic Primitive Based on Cellular Automaton," *Complex Systems*, vol. 28, no. 4, pp. 457–474, 2019.

[2] A. Palchaudhuri and A. S. Dhar, "Design and automation of VLSI architectures for bidirectional scan based fault localization approach in FPGA fabric aware cellular automata topologies," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 110–125, 2019.

[3] A. Palchaudhuri, A. A. Amresh, and A. S. Dhar, "Efficient Automated Implementation of Testable Cellular Automata Based Pseudorandom Generator Circuits on FPGAs," *Journal of Cellular Automata*, vol. 12, no. 3–4, pp. 217–247, 2017.

[4] A. Palchaudhuri and A. S. Dhar, "Built-In Fault Localization Circuitry for High Performance FPGA Based Implementations," *Journal of Electronic Testing*, vol. 33, no. 4, pp. 529–537, Aug. 2017.

2576-2621/20/$31.00 ©2020 IEEE
DOI 10.1109/FCCM48280.2020.00038

IEEE computer society