Applied Soft Computing xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Generating synthetic benchmark circuits for accelerated life testing of field programmable gate arrays using genetic algorithm and particle swarm optimization

Q1 L. Srivani, N.H.V. Krishna Giri, Shankar Ganesh, V. Kamakoti*

Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai 600036, India

ARTICLE INFO

Article history:
Received 9 April 2012
Received in revised form 4 October 2014
Accepted 3 November 2014
Available online xxx

Keywords:
Synthetic benchmark circuits (SBC)
Accelerated life testing (ALT)
Field programmable gate arrays (FPGA)
Genetic algorithms (GA)
Particle swarm optimization (PSO)

ABSTRACT

Accelerated life testing (ALT) of a field programmable gate array (FPGA) requires it to be configured with a circuit that satisfies multiple criteria. Hand-crafting such a circuit is a herculean task as many components of the criteria are orthogonal to each other demanding a complex multivariate optimization. This paper presents an evolutionary algorithm aided by particle swarm optimization methodology to generate synthetic benchmark circuits (SBC) that can be used for ALT of FPGAs. The proposed algorithm was used to generate a SBC for ALT of a commercial FPGA. The generated SBC when compared with a hand-crafted one, demonstrated to be more suitable for ALT, measured in terms of meeting the multiple criteria. The SBC generated by the proposed technique utilizes 8.37% more resources; operates at a maximum frequency which is 40% higher; and has 7.75% higher switching activity than the hand-crafted one reported in the literature. The hand-crafted circuit is very specific to the particular device of that family of FPGAs, whereas the proposed algorithm is device-independent. In addition, it took several man months to hand-craft the SBC, whereas the proposed algorithm took less than half-a-day.

1. Introduction

Programmable Logic Devices (PLD) are widely used as basic building modules in high integrity systems, considering their robust features such as gate density, performance, speed, etc. Field programmable gate array (FPGA) and complex programmable logic device (CPLD) are some of the popularly used PLDs. A FPGA consists of an array of programmable logic modules (LM) and a programmable interconnect area. In modern FPGAs the LMs are realized using Lookup Tables (LUTs). Typically, PLDs are used to (1) program a bus interface logic; (2) programme a glue logic; (3) as a co-processor to the CPU; and/or, (4) as a custom hardware that can offload some of the work done by a CPU with an objective to achieve higher performance levels. The reliability of FPGAs on the other hand, especially when they are used to build safety-critical systems, has been an increasingly interesting phenomenon drawing attention of the industry, researchers and consumers. The reliability of a PLD is so critical that it needs to be quantified and certified by the manufacturers. Many manufacturers use the Mean Time Between Failure (MTBF) value to quantify the reliability of their devices. The MTBF value needs to be very high for FPGAs used in construction of safety-critical systems. One common procedure to validate the MTBF value is to subject the given FPGA to an accelerated life test (ALT). The ALT effort involves the following steps:

1. Generating a design that when configured on the given FPGA under test shall
   - maximize the usage of logic and routing resources for the device;
   - maximize the usage of I/O pins for the device;
   - maximize the frequency at which it can execute on the device;
   - maximize the dynamic power dissipation on execution; and
   - detect faults in the circuit.
2. Subjecting the device programmed with such a circuit to different combinations of environmental parameters.

This paper addresses the problem stated in step (1) above. The challenge is to arrive at such a design as stated in step (1) above that shall *stress* the given FPGA under test. Typically benchmark circuits are used for this purpose. Hand-crafting such a design is a herculean task as the conditions that need to be satisfied to stress many of the parameters mentioned above are orthogonal to each other demanding a complex multivariate optimization. This paper presents a genetic algorithm (GA) aided by particle swarm

Q2 * Corresponding author. Tel.: +91 4422574368; fax: +91 4422574352.
Q3 *E-mail address:* veezhi@gmail.com (V. Kamakoti).

http://dx.doi.org/10.1016/j.asoc.2014.11.002
1568-4946/

Please cite this article in press as: L. Srivani, et al., Generating synthetic benchmark circuits for accelerated life testing of field programmable gate arrays using genetic algorithm and particle swarm optimization, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.002

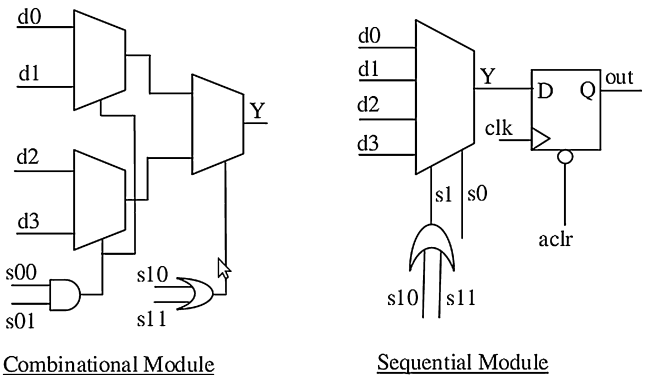

Fig. 1. Combinational and sequential blocks on target FPGA architecture.

optimization (PSO) methodology to generate synthetic benchmark circuits (SBC) that can be used for ALT of FPGAs. The proposed algorithm is device-independent and can be used to generate SBCs for ALT of any FPGA.

The rest of this paper is organized as follows: Section 2 presents a literature survey on SBCs. Section 3 presents the proposed GA for generating SBCs for ALT of FPGAs and the corresponding experimental results. Section 4 presents the PSO-based enhancement to the GA and the corresponding experimental results. Section 5 concludes the paper.

2. SBCs: a survey

Evolution of new FPGA architectures and EDA tools is important and necessary to meet the requirements of tomorrow's designs. What are tomorrow's designs? How big and complex will they be? Realistic answers to these questions are needed to tune the evolution process mentioned above to the futuristic requirements. The industry and academia addresses the same using SBCs.

One of the earliest reported results that has formed the basis of many SBC generation techniques is *Rent's law* [1]. This empirical law advocates the following relationship between the average number of elementary blocks B in the modules of a partitioned circuit, and the average number of external connections (terminals) T from modules:

$$T = tB^p;$$

where t is the average number of terminals per logic block and p is called the *Rent exponent*. The Rent exponent p, is a measure of the interconnection complexity of the circuit ($0 < p < 1$). A larger value of p indicates larger interconnection complexity. Thus, Rent's law addresses the notion of interconnection complexity of the circuit. It is interesting to observe that, over years the real designs indeed followed Rent's law in terms of interconnect complexity except in some cases where a deviation to Rent's rule was observed for large values of T and B. These cases were said to belong to the Rent's II region. The first attempt towards generating SBC is reported in [2], wherein, the proposed technique generated a random mapped circuit `rmc`. It was based on a top-down recursive partitioning approach. In this method, the design is considered in its entirety to start with and further partitioned into sub-blocks based on certain criteria. Generally the criteria will be to minimize the number of *net-cut*, i.e., the number of nets crossing the borders of the sub-blocks in the partition. It has been observed that minimizing the number of terminals is a better criterion than minimizing the number of net-cut and that both these criteria are not equal for multi-terminal external nets. The drawback of the `rmc` method was that it could not generate a realistic sequential circuit. Even the targeted Rent's exponent was not achieved. This showed that more focused methods were needed for generating SBCs.

The recent SBC generation methods can be broadly classified into two: *cloning-based* and *mutant-based*. In the cloning approach, a minimal set of characteristic parameters are identified and the SBC is derived from them. In the mutant approach, a circuit is drawn from a set of all possible circuits and perturbed to arrive at the new SBC. This derived circuit shall meet the new requirements of the user while adhering to some properties of the original circuit.

Many cloning-based approaches are reported in the literature. In [3] a bottom-up clustering approach named `gnl` (generate netlist) is presented, which is based on Rent's rule. In this method, first a set of primitive modules are selected. Then, two different primitive modules are combined to form a cluster or a new module. Two different clusters are combined to form a new cluster. They were successful in generating true sequential circuits, but failed in Rent region II. Hutton et al [4] came out with two tools named `CIRC` and `GEN`. The `CIRC` was used to analyse the sequential circuits to generate a profile which in turn was used by `GEN` to generate the SBC. The *delay distribution* of the circuits generated by *GEN* and *CIRC* was more realistic than what was realized in randomly generated circuits. The delay distribution of a given circuit is a collection of the maximum path delay values for all inputs of its sequential gates and all its primary outputs [5]. However, the drawback of `CIRC` was that it was not able to extract the Rent characteristic of the analysing circuit.

Ghosh et al [6] introduced the mutant-based SBC generation approach named signature invariant mutants. They further classified them as the wiring signature-invariant (WSI) classes and functional perturbation-invariant (FPI) classes. WSI mutants were derived by doing the wiring perturbation, whereas FPI mutants were derived by adding extra logic to the existing circuit.

In addition to Rent's law, some approaches [7,8] attempt to build SBCs with *net-degree* distributions similar to those in given set of real circuits. The first attempt to validate whether a generated SBC is indeed a representative of the real circuits that shall be encountered in future or not was made in [7]. In addition, the methodologies discussed so far used only the topological parameters. On the other hand, there is a need to generate SBCs that have certain functional properties. This is needed, for example, in the case of validating logic synthesis/optimization tools. Such an attempt is reported in [8].

In 2005, Pecenka et al [9] introduced an evolutionary algorithm for generating SBCs targeted for ATPG tools. They were able to generate RTL circuits with 150, 000 gates. A benchmark suite named FITTest_BENCH06 set, consisting of 20 synthetic sequential circuits with maximum 310, 610 gate complexity was proposed in [10]. A methodology for generating benchmark circuits with predefined testability was proposed in [11,12]. The fitness function used in [12] for grading the chromosomes included testability of the generated

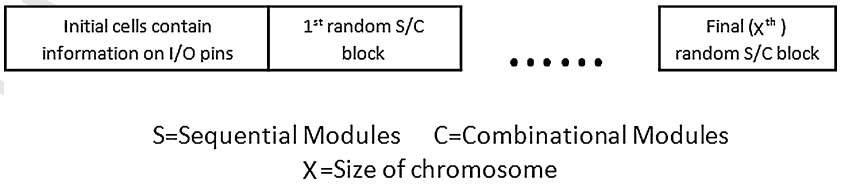

Fig. 2. Chromosome structure.

Please cite this article in press as: L. Srivani, et al., Generating synthetic benchmark circuits for accelerated life testing of field programmable gate arrays using genetic algorithm and particle swarm optimization, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.002