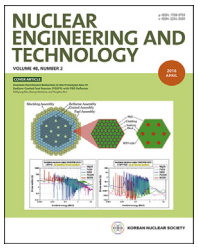
Nuclear Engineering and Technology 48 (2016) 470–481

Available online at ScienceDirect

Nuclear Engineering and Technology

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



Original Article

An Integrated Software Testing Framework for FPGA-Based Controllers in Nuclear Power Plants☆

Jaeyeob Kim [a], *Eui-Sub Kim* [a], *Junbeom Yoo* [a,*], *Young Jun Lee* [b], *and Jong-Gyun Choi* [b]

[a] *Division of Computer Science and Engineering, Konkuk University, 1 Hwayang-dong, Gwangjin-gu, Seoul, 143-701, Republic of Korea*

[b] *MMIS Lab., Korea Atomic Energy Research Institute, 989-111 Deadeok-daero, Yuseong-gu, Daejeon, 305-353, Republic of Korea*

ARTICLE INFO

Article history:
Received 1 September 2015
Received in revised form
28 November 2015
Accepted 3 December 2015
Available online 21 January 2016

Keywords:
Co-simulation
FPGA
Simulation
Testing
Verification

ABSTRACT

Field-programmable gate arrays (FPGAs) have received much attention from the nuclear industry as an alternative platform to programmable logic controllers for digital instrumentation and control. The software aspect of FPGA development consists of several steps of synthesis and refinement, and also requires verification activities, such as simulations that are performed individually at each step. This study proposed an integrated software-testing framework for simulating all artifacts of the FPGA software development simultaneously and evaluating whether all artifacts work correctly using common oracle programs. This method also generates a massive number of meaningful simulation scenarios that reflect reactor shutdown logics. The experiment, which was performed on two FPGA software implementations, showed that it can dramatically save both time and costs.

1. Introduction

Programmable logic controllers (PLCs) [2] are widely used to implement safety-critical systems for digital instrumentation and control (I&C) of Nuclear power plants (NPPs). The increasing complexity of newly developed systems and maintenance costs are now demanding more powerful and cost-effective implementation, such as field-programmable gate arrays (FPGAs) [3]. The nuclear industry is now eagerly researching FPGA-based digital I&Cs [4–6] to replace PLC-based systems. In turn, international standards [7–9] require more rigorous demonstrations of the safety of these new systems.

The FPGA software, which this paper concerns, was first modeled with hardware description languages (HDLs), such as Verilog and VHSIC HDL (VHDL), by software designers manually, and then subsequently synthesized into gate-level designs and physical layouts by software synthesis tools provided by FPGA vendors (e.g., ISE Design Suite (Xilinx, San Jose, CA, USA) [10], Quartus Prime (Altera, San Jose, CA, USA) [11], and Libero SoC (Microsemi, Aliso Viejo, CA, USA) [12]).

☆ A preliminary version of this paper was presented at the Transactions of the Korean Nuclear Society Autumn Meeting, Pyeongchang, Korea, in 2014 [1].

* *Corresponding author.*
E-mail address: jbyoo@konkuk.ac.kr (J. Yoo).

http://dx.doi.org/10.1016/j.net.2015.12.008

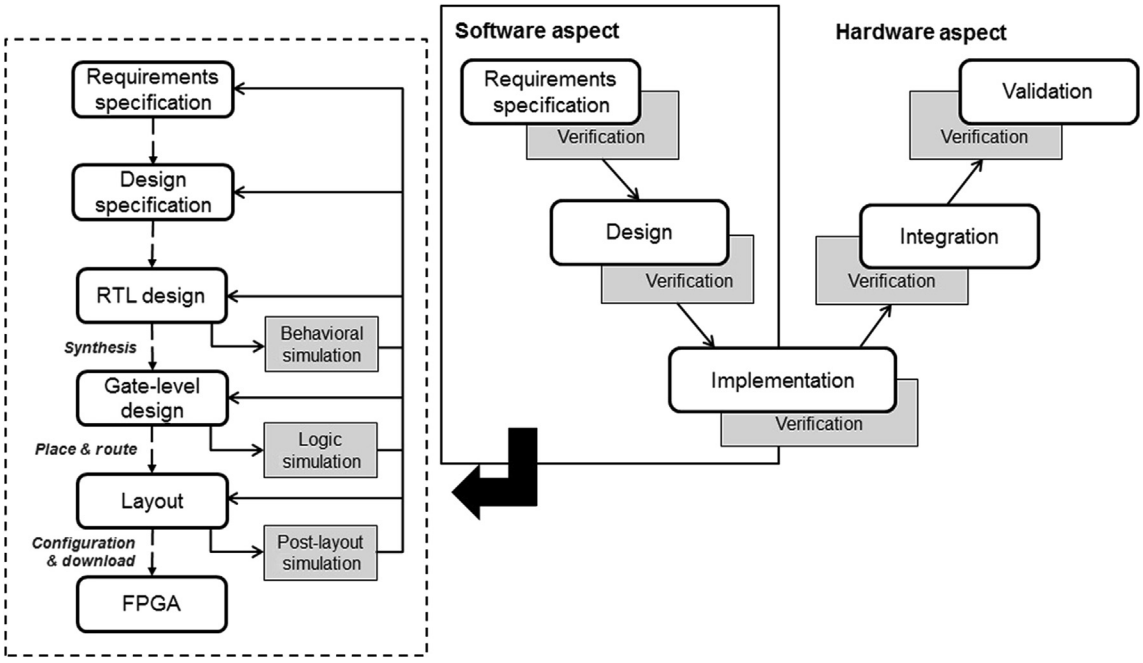


Fig. 1 – The V-shaped life cycle and a typical FPGA development process. FPGA, field-programmable gate arrays; RTL, register-transfer level.

FPGA tools make the synthesis process fully automatic, and software designers largely focus on HDL designs to implement FPGA requirements correctly.

FPGA software designers also use verification techniques, such as "simulation" [13–15], in order to check if high-level designs are correctly synthesized into low-level ones. At each step [i.e., register-transfer level (RTL), gate-level, and layout], designers perform three common activities. They first develop test scenarios, then simulate each target in a test bench, and finally evaluate (i.e., observe) the simulation results against specified requirements. The problem on which this study focused is that the verification activity should be performed at each step individually and repetitively. Furthermore, the individual preparation for each verification step, such as developing test scenarios and test benches, takes considerable time and money.

This paper proposed an integrated software testing framework for FPGA software developments (IST-FPGA). It allows us to perform the three activities of the simulation-based verification only once and in one step. For all design artifacts at every step, it generates common and meaningful test scenarios mechanically, simulates all designs simultaneously, and finally evaluates the simulation results against expected ones altogether. If any one of the designs show different (i.e., incorrect) behavior from the expected one (i.e., a comparison oracle program), IST-FPGA analyzes and compares the incorrect case in detail. IST-FPGA is also supported by CASE tools, such as Verilog/VHDL Scenario Generator and Co-simulator.

In order to demonstrate the effectiveness of IST-FPGA, we performed an experiment with two FPGA-based I&C systems that are under development by the Korea Atomic Energy Research Institute (Daejeon, Korea). This experiment successfully demonstrated how IST-FPGA can reduce the time and cost for the simulation-based verification of FPGA software. The remainder of the paper is organized as follows: Section 2 provides background information on FPGA verification and simulation techniques. Various standards and guidelines for developing and verifying FPGA-based digital I&Cs are briefly surveyed. Section 3 proposes the integrated software testing framework for FPGA as well as assisting tools we developed. Experiment results are presented in Section 4, and Section 5 surveys related research. Section 6 concludes the paper and provides remarks on future research extension and direction.

2. Background

2.1. *FPGA development and software verification*

The system development life cycle of FPGA-based I&Cs should follow IEC-61513 [9]. An FPGA-based system has a specific feature that the portion of the development life cycle that uses HDL be classified as software, then once it is downloaded to a chip, it is classified as hardware. FPGA, therefore, should be developed to meet both IEC-60880 [8] in terms of software and IEC-60987 [16] in terms of hardware. Fig. 1 depicts the V-shaped life cycle of FPGA development explained in IEC-62566 [17], consisting of software and hardware aspects. The software aspect also has a typical development life cycle [18] presented on the left-hand side of the figure.

At each step of the FPGA software development life cycle, designers perform a simulation-based verification in order to confirm that each artifact satisfies its required specification. The first simulation on RTL designs, called behavioral simulation, aims to confirm that all requirements are implemented