Nuclear Engineering and Technology 48 (2016) 1047–1057

Available online at ScienceDirect

Nuclear Engineering and Technology

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

Technical Note

Development of Field Programmable Gate Array-based Reactor Trip Functions Using Systems Engineering Approach

*Jaecheon Jung** *and Ibrahim Ahmed*

Nuclear Power Plant Engineering, KEPCO International Nuclear Graduate School, 1456-1 Shinam-ri, Seosang-myeon, Ulju-gun, Ulsan 689-882, Republic of Korea

ARTICLE INFO

Article history:
Received 13 November 2015
Received in revised form
16 February 2016
Accepted 17 February 2016
Available online 21 March 2016

Keywords:
Field Programmable Gate Array
Finite State Machine with Data Path
Reactor Trip Functions
Systems Engineering

ABSTRACT

Design engineering process for field programmable gate array (FPGA)-based reactor trip functions are developed in this work. The process discussed in this work is based on the systems engineering approach. The overall design process is effectively implemented by combining with design and implementation processes. It transforms its overall development process from traditional V-model to Y-model. This approach gives the benefit of concurrent engineering of design work with software implementation. As a result, it reduces development time and effort. The design engineering process consisted of five activities, which are performed and discussed: needs/systems analysis; requirement analysis; functional analysis; design synthesis; and design verification and validation. Those activities are used to develop FPGA-based reactor bistable trip functions that trigger reactor trip when the process input value exceeds the setpoint. To implement design synthesis effectively, a model-based design technique is implied. The finite-state machine with data path structural modeling technique together with very high speed integrated circuit hardware description language and the Aldec Active-HDL tool are used to design, model, and verify the reactor bistable trip functions for nuclear power plants.

1. Introduction

In the nuclear domain, the field programmable gate array (FPGA) is the most recent electronic device that is being considered by stakeholders to replace the software-based systems in performing the trip functions of the reactor protection system (RPS) of nuclear power plants (NPPs) because of its potentials such as simplicity, testability, long-term support, and being easier to qualify. The RPS is the most safety-critical instrumentation and control (I&C) system in NPPs. It safely trips the reactor whenever one or more of the monitored plant processes exceed predefined limits.

Due to criticality of the RPS, the software used in programmable logic controllers (PLCs) is rated as high-integrity

* *Corresponding author.*
E-mail address: jcjung@kings.ac.kr (J. Jung).
http://dx.doi.org/10.1016/j.net.2016.02.011

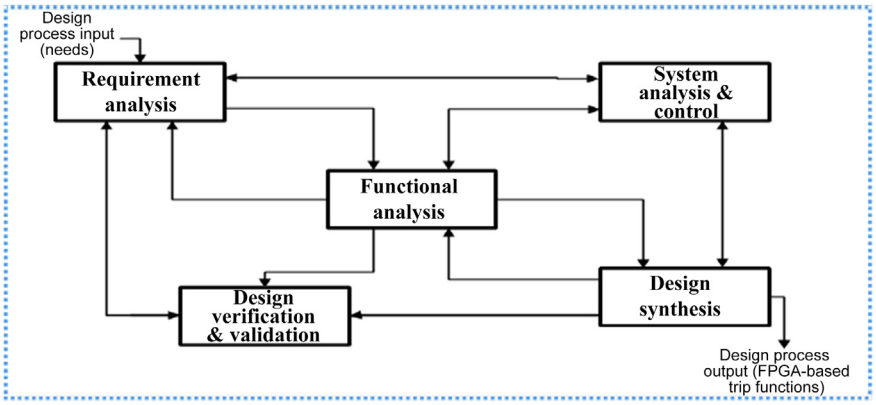

Fig. 1 – Design process (DOD MIL-STD-499B [4]). FPGA, field programmable gate array.

software, and therefore assigned the highest software integrity level: 4. The higher the software integrity level the higher the demand for verification and validation (V&V) activities. As indicated by IEEE Std. 1012 [1], the high-integrity software requires a larger set of V&V processes and a more rigorous application of V&V tasks.

By replacing the PLC-based system with the FPGA-based system, the use of OS and complex software applications during plant operation can be minimized if not completely eliminated. An FPGA is a digital semiconductor device that can be used as a replacement for the current microprocessor-based software systems. It is a digital programmable integrated circuit (IC) that contains thousands or millions of logic gates and interconnections that can be configured to implement desired functionality. Even though FPGA design process involves the use of configuration/programing software, the end product of the design can be regarded as a hardware-based system [2,3].

However, to replace PLC functionalities with FPGA to perform the trip functions, the development of FPGA-based bistable trip algorithms is essential. Without the development of proper algorithms for FPGA, the replacement is completely impossible.

Applying an FPGA to perform RPS functions requires proper and accurate RPS bistable algorithms development. If a proper and well-defined design process is applied to FPGA-based RPS design, the V&V tasks can easily be achieved and design error can be minimized. Therefore, the main focus area of this work is to make the V&V of FPGA-based RPS functions simpler using systems engineering approach in combination with finite-state machine with data path (FSMD) structural modeling techniques.

In order to develop an FPGA-based reactor trip functions, the systems engineering approach defined by DOD MIL-STD-499B [4] is applied (Fig. 1). The rectangular boxes represent the stages for the development process. There are also inputs to and outputs from the design process. The inputs are needs from need/system analysis to the requirement analysis phase, and the output is the final design outcome from design synthesis.

The development life cycles recommended by IEC 62566 [5] and EPRI TR1019181 [3] for FPGA development in NPP are based on the traditional software V-model. The design of FPGA involves both hardware and software design process. However, the classical software development life cycle is not suitable for the FPGA design life cycle. The Y-model is known for a hardware–software codesign. The suitability of Y-cycle for safety critical software for I&C system in NPP was demonstrated by Jung et al. [6] using the 3-Step software development process, and concluded that around 50% of development time savings is expected to be achieved by adopting Y-Cycle. This indicates Y-model transformed from the traditional V-model for FPGA-based trip function design (Fig. 2).

In the design and development of an FPGA system, the code is compiled and mapped on the target architecture. The resulting intermediate implementation is then tested and evaluated with respect to timing, power consumption, cost, etc., using simulation and analysis. Based on these metrics, the designer decides about architecture and/or code adaptations. This process is iteratively repeated until a satisfactory design is found. Therefore, according to Hamann [7], the risk that is linked to the design flow due to the Y-model is relatively small, since the designer can react in each iterations to performance problem and solve them.

The design synthesis phase, which comprises design and implementation stages of FPGA-based RPS functions, is

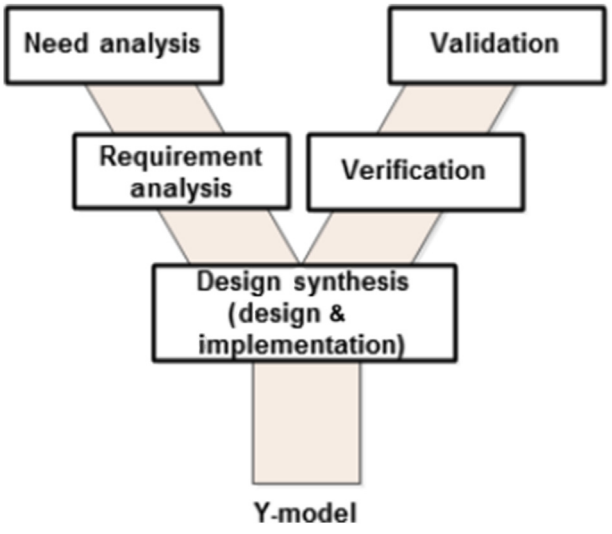

Fig. 2 – Y-model transformed from traditional V-model for field programmable gate array based trip function design.