Development of a Software Engineering Lifecycle Framework Incorporating Traceability for FPGA-Based Platforms

Hangyu Kim [a*], Dongil Lee [a]
[a]*KHNP, Central Research Institute, 70, Yuseong-daero 1312beon-gil, Yuseong-gu, Daejeon, 34101, South Korea*
[*]*Corresponding author: hangyukim@khnp.co.kr*

***Keywords :** Field Programmable Gate Array, Software Configuration Management, Requirements Traceability

1. Introduction

As the Instrumentation and Control (I&C) systems of nuclear power plants become increasingly digitalized, the importance of software Verification and Validation (V&V) is growing to eliminate software errors. Software V&V activities ensure that the software requirements are correctly implemented in the software deliverables throughout the entire software engineering lifecycle. Typically, the software V&V process for nuclear power plants is conducted in accordance with IEEE Std. 1012-2004 [1].

Recently, FPGA-based platforms are being developed for I&C systems of nuclear power plants. FPGA-based platforms shall be designed and verified throughout the entire software engineering lifecycle, considering the hardware design characteristics of FPGA in accordance with IEC 62566: 2012 [2]. To enhance the design quality of FPGA-based platforms affecting safety of nuclear power plants, a systematic software engineering lifecycle framework is being established.

This paper describes a systematic software engineering approach for performing software configuration management and requirements traceability analysis, specifically considering the FPGA environment.

2. Software Engineering Lifecycle Framework

Software management configuration activities are traditionally grouped into five functions: configuration identification, configuration control, configuration status accounting, configuration evaluations and reviews, and release management and delivery in accordance with IEEE Std. 828-2005 [3]. Source codes, software documents and elements to be managed for project are identified as configuration management items.

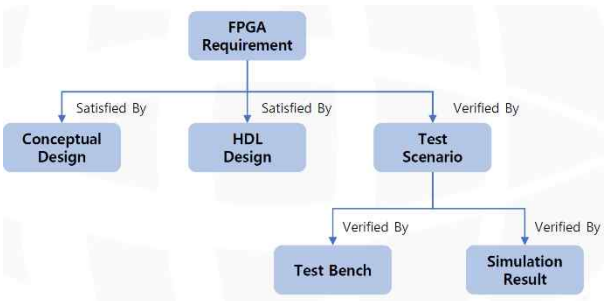


Fig. 1. Schema for FPGA requirements management

Figure 1 shows the schema for requirements management applied to the FPGA-based platform development project. At the top level, FPGA requirements are identified to satisfy the functional of the circuit card assembly. And then traceability for these FPGA requirements is established as follows:

1) Traceability from FPGA requirements to conceptual design documents
2) Traceability from FPGA requirements to Hardware Description Language (HDL) design (including implementation code)
3) Traceability from FPGA requirements to test scenario
4) Traceability from test scenario to test bench
5) Traceability from test scenario to simulation result

Figure 2 shows the requirements traceability analysis for the FPGA-based platform development project, based on V-model of the software life cycle. The abbreviations used in Figure 2 are listed in Table 1.

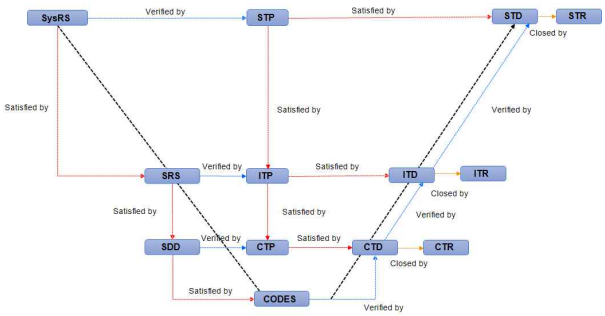


Fig. 2. Requirements traceability analysis based on V-model

Table 1: List of acronyms

Acronyms	Full name
CTD	Component Test Design
CTP	Component Test Plan
CTR	Component Test Report
ITD	Integration Test Design
ITP	Integration Test Plan
ITR	Integration Test Report
SDD	Software Design Description
SRS	Software Requirements Specification
STD	System Test Design
STP	System Test Plan
STR	System Test Report
SysRS	System Requirements Specification

Requirements traceability analysis consists of vertical and horizontal traceability analysis. Vertical traceability analysis ensures that the software is correctly designed and implemented based on the defined software requirements.

The traceability analysis begins with the identification of system requirements that are to be implemented completely or partially by software. And each software requirement shall be traceable to a system requirement. Also, all design elements shall be traceable to the software requirements, and all source code components shall be traceable to the design elements.

Horizontal traceability analysis is performed to verify how software requirements and design elements are tested at each phase of the software life cycle.

The traceability among the test plan, test design, and test report is analyzed for correctness and completeness. The test design is incorporated into the test procedure documents. For each phase of the software life cycle, the test procedures shall be traceable to the test plans, and the test reports shall be traceable to the test procedures.

The component testing is performed with developed test procedures to verify and validate that the software correctly implements the software design. The integration testing is performed with developed test procedures to verify and validate that the software component integration correctly implements the software requirements and design. The system testing is performed with developed procedures to verify and validate that the completely integrated system meets the requirements.

This requirements traceability analysis is managed through a Requirements Traceability Matrix (RTM) as shown Figure 3.

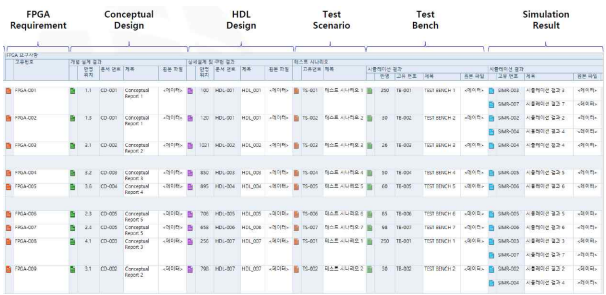


Fig. 3. Requirements traceability matrix

3. Conclusions

Software configuration management and requirements traceability analysis are very important elements in software V&V activities. The software engineering lifecycle framework presented in Section 2 allows for the systematic software configuration management of development and verification deliverables throughout the entire software engineering lifecycle. Also, this framework ensures traceability throughout the entire software engineering lifecycle, from initial requirements identification to final test results.

This approach enables regulatory compliance by meeting the IEEE Std. 1012 and IEC 62566 technical standards for FPGA development and verification, and it contributes to enhancing the design quality and ensuring the integrity of FPGA-based platforms. Consequently, the application of reliability-assured FPGA-based platforms to the I&C systems will ensure the safety of the nuclear power plant.

REFERENCES

[1] IEEE Std. 1012-2004, “IEEE Standard for Software Verification and Validation”
[2] IEC 62566: 2012, “Nuclear power plants - Instrumentation and control important to safety - Development of HDL-programmed integrated circuits for systems performing category A functions”
[3] IEEE Std. 828-2005, “IEEE Standard for Software Configuration Management Plans”