Transactions of the Korean Nuclear Society Autumn Meeting
Jeju, Korea, October 21-22, 2010

Life Cycle V&V Process for Hardware Description Language Programs of Programmable Logic Device-based Instrumentation and Control Systems

K. H. Cha and D. Y. Lee
Korea Atomic Energy Research Institute
1045, Daedeok-Daero, Yusong, Daejon, 305-600, Republic of KOREA
{khcha, dylee}@kaeri.re.kr

1. Introduction

Programmable Logic Device (PLD), especially Complex PLD (CPLD) or Field Programmable Logic Array (FPGA), has been growing in interest in nuclear Instrumentation and Control (I&C) applications [1, 2, 3]. PLD has been applied to replace an obsolete analog device or old-fashioned microprocessor, or to develop digital controller, subsystem or overall system on hardware aspects. This is the main reason why the PLD-based I&C design provides higher flexibility than the analog-based one, and the PLD-based I&C systems shows better real-time performance than the processor-based I&C systems. Due to the development of the PLD-based I&C systems, their nuclear qualification has been issued in the nuclear industry.

Verification and Validation (V&V) is one of necessary qualification activities when a Hardware Description Language (HDL) is used to implement functions of the PLD-based I&C systems. The life cycle V&V process, described in this paper, has been defined as satisfying the nuclear V&V requirements, and it has been applied to verify Correctness, Completeness, and Consistency (3C) among design outputs in a safety-grade programmable logic controller and a safety-critical data communication system.

Especially, software engineering techniques such as the Fagan Inspection, formal verification, simulated verification and automated testing have been defined for the life cycle V&V tasks of behavioral, structural, and physical design in VHDL.

2. Life Cycle V&V Process

2.1 Codes and Standards

The V&V shall be performed as satisfying the requirements defined in the nuclear codes and standards including the IEEE Std. 7-4.3.2, IEEE Std. 1012, and NUREG-0800/SRP-14, IEC 60880, IEC 61508. Regulatory position on HDL programs is that the development and V&V requirements for software should be equally applied to the HDL programs for PLD-based I&C systems.

2.2 Life Cycle V&V Process and Their Tasks

The life cycle V&V process for HDL programs is based on the life cycle V&V process defined in the IEEE Std. 1012, and consists of blued ones in Figure 1. The V&V tasks have been defined for each V&V stage, reflecting software V&V tasks introduced in the IEEE Std. 1012. For example, the V&V tasks of behavioral description in HDL is mapped to requirements V&V in software, consequently the V&V tasks of structural description in HDL to architectural V&V, and the V&V tasks of physical design in HDL to detailed design V&V and implementation V&V.

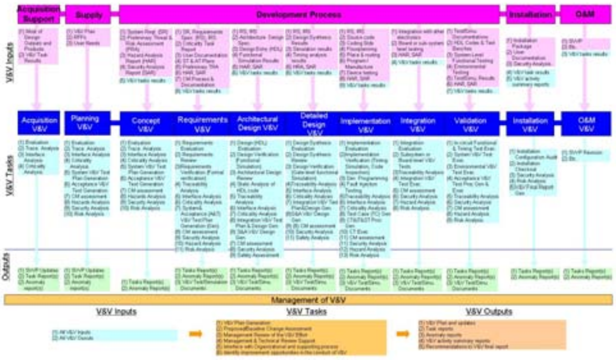

Figure 1. Life Cycle V&V Process and Its Tasks for HDL Programs

2.3 V&V Methods and Techniques

Simulation, emulation, and testing have widely been used for the V&V of behavioral, structural, and physical design in HDL. In addition, software V&V techniques such as the review and inspection, formal verification, automated software testing can be applied to verify and validate the design and coding in HDL.

Review and Inspection have widely been used for software V&V. For more formal review, Fagan Inspection can be applied for all phases of software life cycle. Formal verification using model checking and theorem proving are required when a system's behavior is specified formally by using mathematical representation such as logic or algebra.

A software life cycle testing consisting of test plan generation, test design generation, test cases generation, test procedure generation and test execution can be applied for the life cycle testing of the HDL programs.

3. Applications

The life cycle V&V process described in section 2 was applied for verifying 3C of the Very High Speed Integrated Circuit HDL (VHDL) programs, compliant of the IEEE Std. 1074. The VHDL programs were developed for the safety-grade PLC platform embedding Xilinx's CPLDs [4] and Broadband-Nuclear Safety Data Network embedding Altera's FPGAs.

In order to improve efficiency in the life cycle V&V, the well-structured V&V procedures were written for each life cycle phase, and the V&V tasks, methods, techniques and tools were defined in the V&V procedures. Figure 2 depicts the functional and process V&V properties (a), V&V procedures (b), and V&V process (c) of the VHDL programs.

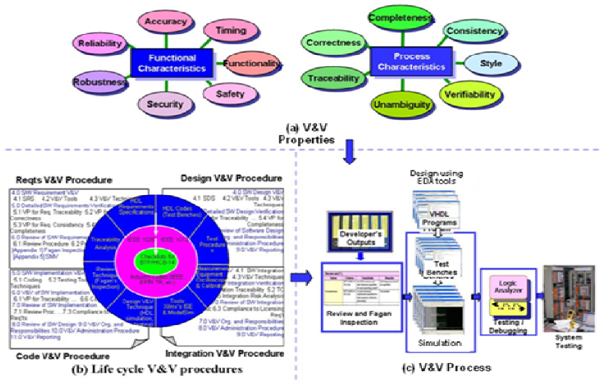

Figure 2. V&V process of VHDL Programs

All of outputs produced through the development process of the VHDL programs were reviewed with the checklists defined in V&V procedures, and inspected through the Fagan's Inspection process consisting of inspection planning, product overview, inspection preparation, examination meeting, defect rework, and resolution follow-up.

Simulation was applied for the V&V of behavioral, structural, and physical design in VHDL. Simulation capabilities in behaviors/functions and timing were provided by ModelSim tool. Equivalence between requirements and design, design and coding, and coding and implementation, were checked for signal variables and functions using the simulation input data.

Test techniques such as white box test, black box test and regression test were utilized for the component test, integration test, and system test. Figure 3 shows the simulations by using ModelSim and Projector Navigator tools.

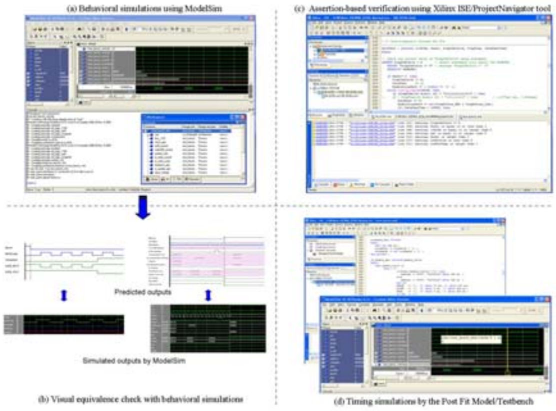
(a) Behavioral simulations using ModelSim
(c) Assertion-based verification using Xilinx ISE/ProjectNavigator tool
Predicted outputs
Simulated outputs by ModelSim
(b) Visual equivalence check with behavioral simulations
(d) Timing simulations by the Post Fit Model/Testbench

Figure 3. Simulated Verification of VHDL Programs

4. Conclusion

The life cycle verification and validation process in software engineering aspects can be applied for Hardware Description Language programs of the Programmable Logic Device-based instrumentation and control applications. Thus we have defined the life cycle verification and validation tasks, methods, techniques and supported tools for Hardware Description Language programs, and applied to a couple of programmable logics for control and data communication. The applications show that the technical review and Fagan Inspection, simulated verification, and life cycle testing among software verification and validation techniques are very useful for the life cycle verification and validation of Hardware Description Language programs in Programmable Logic Device-based instrumentation and control applications.

REFERENCES

[1] J. G. Choi, Codes and Standards on CPLD/FPGA (in Korean), KNS 2009 Spring Conf. - Workshop on Nuclear Applications of CPLD/FPGA Technology, 2009.
[2] J. H. Lee, V&V Cases of CPLD/FPGA for Digital I&C Systems (in Korean), KNS 2009 Spring Conf. - Workshop on Nuclear Applications of CPLD/FPGA Technology, 2009.
[3] J. K. Lee, CPLD/FPGA Application in SMART (in Korean), KNS 2009 Spring Conf. - Workshop on Nuclear Applications of CPLD/FPGA Technology, 2009.
[4] K. H. Cha, J. G. Choi, and K. C. Kwon, Nuclear Qualification of the I/O Modules for a Safety-Grade PLC, Tr. ANS/ENS Int'l Meeting, pp. 25-26, 2007.
[5] F. Vahid and T. Givargis, Embedded System Design – A Unified Hardware/Software Introduction, John Wiley & Sons, Inc., 2002.