Transactions of the Korean Nuclear Society Autumn Meeting
PyeongChang, Korea, October 30-31, 2008

Verification Factors of Programmable Logic Design for Digital Instrumentation and Control System

G. Y. Park[a], C. H. Jung[b] and D. I. Kim[a]
a Instrument & Control Division, Korea Institute of Nuclear Safety, 19 Kusong-dong, Yuseong, Daejeon, Korea.
b Research Division, Korea Institute of Nuclear Safety, 19 Kusong-dong, Yuseong, Daejeon, Korea.
{k703pgy, k148jch, dikim}@kins.re.kr

1. Introduction

As analog equipments in the nuclear power plant become obsolete, digital upgrades of nuclear facility are becoming popular. As a part of this trend, the programmable logic component, i.e., CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Logic Device) adopted in a nuclear facility is gradually increasing because of its flexibility. In this circumstance, the verification activity is an important part of programmable logic design to guarantee error-free product. Therefore, this paper addresses the important factors in the verification activity such as verification coverage goal, verification coverage measurement.

2. Verification Factors on Programmable Logic Design

In the programmable logic design, the objective of verification is ensuring the functional correctness of design [1, 2, 3, 4]. Namely, the verification is a process to demonstrate that the intent of a design is successfully preserved in its implementation [4]. However, because of the increasing complexity of the design, verification process is becoming more and more difficult to ensure the functional correctness of design. In this chapter, we demonstrate the important issues in the verification activity.

2.1 Verification Coverage Goal

In the verification activity, it is difficult to determine how much verification efforts are enough. In other words, knowing how long it will take to complete the verification activity is an important issue. To obtain a qualified product, it is reasonable to determine coverage goal. For example, the logic design having 90 percent of verification coverage is more confident than that having 50 percent. However, there are some factors to determine the coverage goal. The first is verification cost. In the previous example, the former needs more cost than that of the latter because of a trade-off relationship between the cost of verification and the coverage goal. The second is exponentially increasing test cases [6]. For example, in the case of the logic design having 8 inputs and 100 percent verification coverage goal, the number of test cases is 2^8 or 256 test cases. If the logic design includes 8 inputs with 4 internal state and 100 percent coverage goal, the test cases are 2048. In this issue, U.S NRC (Nuclear Regulatory Commission) defines a simple digital device as having 100 percent verification coverage [6]. And, for the simple digital device, every possible combination of inputs and internal state are tested and all outputs are verified for every case.

2.2 Verification Methodology

Choosing a verification methodology suitable for the design property is an important issue in verification activity. Therefore, the verification plan should clearly state which parts of the design will be subjected to code or functional coverage. The code coverage is a verification methodology identifying which code has been verified or not. The code coverage includes 4 types of coverage metric – statement, condition, and path coverage. The statement coverage measures how many lines of the source code are verified in the verification process. Similarly, the condition coverage identifies which conditions are executed in the verification activity and the path coverage verifies that every possible route in the code been executed. The code coverage is a necessary in the verification activity because it is easy to implement and to achieve the coverage goal. However, it is not sufficient as a reliable verification methodology because the code coverage would not ensure the functional correctness. Therefore additional coverage methodology such as functional coverage is required to verify functional correctness.

The functional coverage ensures whether the implementation is accordance with the requirement specification. Therefore, functional coverage metrics are derived from the functional or design specification. For collecting functional coverage metrics, the additional design effort such as assertion-based design is required in the implementation process. Contrary to the code coverage, it is difficult to implement the functional coverage. Furthermore, nowadays, there are no sufficient software tools for the functional coverage. However, there is an effort for the industrial unified hardware description and verification language (HDVL) such as IEEE Std 1800, “IEEE Standard for SystemVerilog-Unified Hardware Design, Specification, and Verification Language.” [7]

2.3 Verification Coverage Measurement

In order to measure progress of verification activity, verification coverage is measured in terms of metric. Therefore, coverage metric is essential in measuring

		Metric Kind	
		Implicit	**Explicit**
Metric Source	**Specification**	**Implicit Specification Coverage**	**Specification Functional Coverage**
	Implementation	**Code (or Structural) Coverage**	**Implementation Functional Coverage**

Figure 1. Verification Coverage Metric.

verification coverage [8]. As shown in Fig. 1, the verification metrics are classified into two categories – metric kind and source. The metric kind is either explicit or implicit. The implicit coverage metric is inherent in the abstraction level of verification target such as HDL code structure or coding style. The explicit coverage metric is usually derived from specification or implementation. For example, CRC, frame length, execution mode or addressing mode in the specification could be an explicit metric. The metric source is either specification or implementation. The specification metric is a metric derived from the specification.

Each of coverage metrics is used to observe device functions from a different perspective as shown in Fig. 1. The specification functional coverage indicates what features in the specification to be verified. Therefore the specification functional coverage covers input, output and internal interfaces described in the specification. The implicit specification coverage indicates how much of the device specification has been exercised. However, there are coverage metrics which is difficult to implement.

3. Conclusion

According to increasing the use of programmable logic component adopted in a nuclear facility, the verification activity is essential in the programmable logic design. Furthermore, because of the increasing design complexity, the verification process requires more time and effort to ensure the functional correctness of design. In this circumstance, the factors such as verification coverage goal, methodology and coverage measurement & analysis are becoming significant in the verification activity. Therefore, these issues are dominant factor to perform verification activity successfully. The design should be precisely examined in the verification factors. According to the derived result, a verification plan should be established and described in the view of the coverage issues.

REFERENCES

[1] J. A. Cercone, M. A. Beims, and K. G. McGill, Verification and Validation of Programmable Logic Devices, National Aeronautics and Space Administration, 2004.

[2] DOT/FAA/AR 95/31, Design, Test, and Certification Issues for Complex Integrated Circuits, Federal Aviation Administration, 1996.

[3] T. D. Tessier, Rethinking Your Verification Strategies for Multimillion-Gate FPGAs, XAPP408, Xilinx, 2002.

[4] J. Bergeron, Writing Testbenches: Functional Verification of HDL Models, Kluwer Academic Publishers, 2000.

[5] J. Bergeron, Writing Testbenches using SystemVeliog, Springer, 2006.

[6] T.W. Jackson, “Licensing of Simple Digital Devices”, ICAPP08, 2008.

[7] IEEE Std 1800, IEEE Standard for SystemVerilog-Unified Hardware Design, Specification, and Verification Language.

[8] Andrew Piziali, Functional Verification Coverage Measurement and Analysis, 2004