Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 10-11, 2007

A Formal Verification Method of Function Block Diagram

Kwang Yong Koh[1], Poong Hyun Seong[1], Eun Kyoung Jee[2], Seung Jae Jeon[2], Gee Yong Park[3], Kee-Choon Kwon[3]
[1]*Department of Nuclear and Quantum Engineering, KAIST*
[2]*Division of Computer Science, Department of Electrical Engineering and Computer Science, KAIST*
373-1 Guseong-dong, Yuseong-gu, Daejeon, 305-701, Korea
[3]*Korea Atomic Energy Research Institute 150 Deokjin-dong Yuseong-gu, Daejeon, 305-353, Korea*
goeric1@kaist.ac.kr, phseong@kaist.ac.kr, ekjee@dependable.kaist.ac.kr, sjjeon@dependable.kaist.ac.kr,
gypark@kaeri.re.kr, kckwon@kaeri.re.kr

1. Introduction

Programmable Logic Controller (PLC), an industrial computer specialized for real-time applications, is widely used in diverse control systems in chemical processing plants, nuclear power plants or traffic control systems [1]. As a PLC is often used to implement safety–critical embedded software, rigorous safety demonstration of PLC code is necessary. Function block diagram (FBD) is a standard application programming language for the PLC [2] and currently being used in the development of a fully-digitalized reactor protection system (RPS), which is called the IDiPS, under the KNICS project [3]. Therefore, verification issue of FBD programs is a pressing problem, and hence is of great importance.

In this paper, we propose a formal verification method of FBD programs; we defined FBD programs formally in compliance with IEC 61131-3, and then translate the programs into Verilog model, and finally the model is verified using a model checker SMV. This approach is illustrated in Figure 1. To demonstrate the feasibility and effectives of this approach, we applied it to IDiPS which currently being developed under KNICS project.

The remainder of this paper is organized as follows. Section 2 briefly describes Verilog [4] and Cadence SMV [5]. In Section 3, we introduce *FBD2V* which is a tool implemented to support the proposed FBD verification framework. A summary and conclusion are provided in Section 4.

2. Background

2.1 Verilog

Verilog is one of the most popular Hardware Description Languages (HDL) used by integration circuit designer. It has several types of variables. A **wire** represents a physical wire in a circuit and is used to connect gates or modules. A wire does not store its value, but should be driven by the **assign** statement or by connected output of a gate or a module. On the other hand, a **reg** is a data object holding its value. Reg variables are assigned only in **always** and **initial** block.

A module is a principal design entry in Verilog. Module declaration specifies the name and list of I/O ports. The first part of a module defines I/O and data type of each port. Keyword **input** and **output** declare the input and output ports of a module. Data type is generally represented as the size of a bit vector. Module declarations are templates from which one creates actual instantiations. Modules are instantiated inside other modules and each instantiation creates a unique object from the template except main module.

2.2 Cadence SMV

Cadence SMV is a model checker based on symbolic model checking technique. It can verify a model programmed in Synchronous Verilog (SV) [6], which has a slight variation of the Verilog language in terms of cycle-based behavior. It converts SV into SMV input language, and then performs model checking.

3. *FBD2V* (FBD to Verilog)

FBD2V [7] automates the FBD verification framework described in Figure 2. It takes *.lda files, which is FBD storing format of a tool pSET currently being developed under KNICS project, as input, and then converts the FBD programs into Verilog model. User can adjust bit sizes and initial values of variables during the translation. After the system properties to be checked are defined as CTL formulae and are embedded in *FBD2V*, *FBD2*V executes Cadence SMV and model checking is performed. To enhance readability of counterexample, timing graph form is displayed. Variables are highlighted in different color and shape for better visualization.

4. Summary and Conclusion

In this paper, we proposed a formal verification method of FBD programs. FBD programs are automatically translated into Verilog model with the supporting tool, *FBD2V,* which is developed to support the FBD verification framework. The model is verified using Cadence SMV, and counterexamples are displayed in the form of timing graph to enhance their readability.

REFERENCES

[1] MADER A, *A classification of PLC models and applications,* Proceedings of WODES 2000: 5th Workshop on Discrete Event Systems, Gent, Belgium; August p. 21–23 (2000).

[2] IEC, *IEC Standard 61131-3: PLC programming languages*, (1993).

[3] J. H. Park, D. Y. Lee, C. H. Kim, *Development of KNICS RPS Prototype*, Proceedings of ISOFIC 2005, Session 6, pp.160-161, Tongyeong, Korea, Nov. 1~4 (2005).

[4] IEEE Std 1364-2005, *IEEE Standard for Verilog® Hardware Description Language*.

[5] K. L. McMillan, "The SMV system for SMV version 2. 5. 4", Nov. 6, 2000.

[6] Ching-Tsun Chou. *Synchronous Verilog*: A Proposal. Fujitsu Laboratories of America, 1997.

[7] Seung Jae Jeon, *Verification of Function Block Diagram through Verilog Translation*, Master Thesis, Department of Electrical Engineering & Computer Science, Division of Computer Science, KAIST, Dec. 2007.

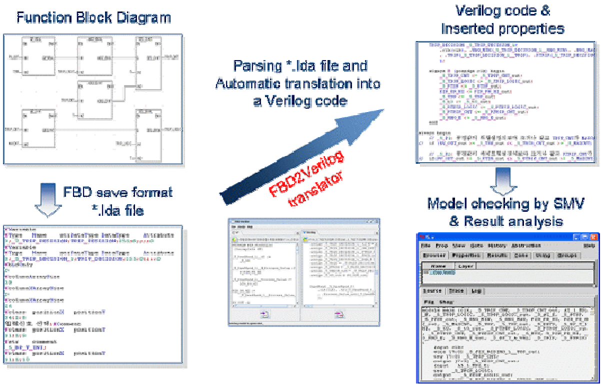

Figure 1. The proposed approach for the verification of FBD programs

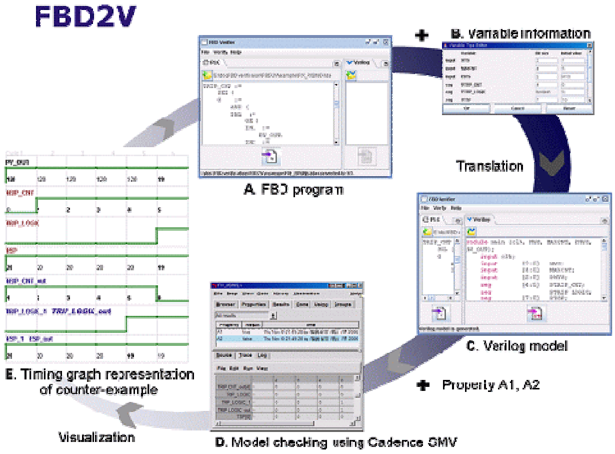

Figure 2. FBD verification framework supported by *FBD2V*