National Institute of
Advanced Industrial Science
and Technology
AIST

AIST-PS-2009-006

Formalization of System LSI Specification and Automatic Generation of Verification Items

Tatsuya Abe†, Takashi Higuchi‡, Rintaro Imai‡, Yoshiki Kinoshita†,
Satoshi Nakano‡, Keishi Okamoto†, Masaya Saito†, and Makoto Takeyama†

† 独立行政法人 産業技術総合研究所 システム検証研究センター
‡ 株式会社 ルネサステクノロジ

算譜科学研究速報
Programming Science Technical Report

Research Center
for Verification and Semantics

独立行政法人 産業技術総合研究所 システム検証研究センター 発行
Published by Research Center for Verification and Semantics
National Institute of Advanced Industrial Science and Technology

Formalization of System LSI Specification and Automatic Generation of Verification Items*
—Extended Abstract—

Tatsuya Abe† Takashi Higuchi‡ Rintaro Imai‡ Yoshiki Kinoshita†
Satoshi Nakano‡ Keishi Okamoto† Masaya Saito† Makoto Takeyama†

2009年7月9日

1 Introduction

The design process of a system LSI traditionally starts with an informal specification in a combination of a natural language and some pseudo programming language such as a pseudo C. This informal specification is used to generate items for verification of lower-level detailed designs. Engineers generating those verification items often face the problems of informality. Ambiguities, implicit assumptions, inconsistencies, etc. in the specification lead to wrong verification items or critical omission. Informality means that the generation is basically a manual process prone to simple errors. Besides those reliability issues, the cost of manual generation is a big issue in the productivity of design processes.

We aim at automatic generation of verification items from specifications. The target specification for this study is that of a system LSI under development at Renesas Technology Corp.

2 Formalization of system LSI specification

We first formulate the instruction set architecture of the system LSI written in Japanese and a pseudo C language as a state transition system in a straightforward manner. The transition system is then formalized in Agda language[1]. Agda is a language based on Constructive Type Theory (Martin-Löf type theory). It integrates computation and logic through Curry-Howard correspondence [3], and Agda is at the same time a programming language, a specification language, and a proof description language. Here we use Agda as a specification language.

Next, we implement an emulator of the system LSI in Agda in a manner similar to the CPU modeling by Xavier Leroy [2] using Agda as a programming language. Our emulator consists of a "formal specification" part and a "executor" part. This separation makes it easier to reuse the executor for similar system

* This research has been supported by Japan Science and Technology Agency (JST)
† CVS, AIST, 1-2-14 Shin-Senri Nishi, Toyonaka, Osaka, 560-0083, Japan
‡ Renesas Technology Corp., 4-1, Mizuhara, Itami-shi, Hyogo, 664-0005, Japan

LSI's. It also encourages to divide the work between LSI designers who concentrate on specification and programmers who provide simulators for validation. We have implemented only a limited number of representative instructions due just to the lack of time for this research project.

Comparing our formal specification in Agda with the traditional one in natural language and pseudo C, we found the following. The length of the Agda specification for the typical ADC instruction (Add with Carry) is about one-fourth of the traditional one with a similar detail. Although readability is hard to quantify numerically, engineers of Renesas Technology Corp. acknowledge that the Agda specification is generally more readable than the traditional one. Reliability is improved through Agda's "dependent type system": on many occasions type checking pointed us to inconsistencies and under-specifications while we edited the specification. This is important as it prevents specification bugs that are costly to fix when they manifest themselves in later stages of development.

3 Automatic generation of verification items

Renesas Technology Corp. uses a body of knowledge on how to generate verification items from a specification, but it is largely kept as "tacit knowing" in the minds of individual engineers. The first stage of automation is to make this tacit knowing an explicit algorithm by means of interviewing. It is then implemented as the verification-item generator program written in Agda.

Inputs to our generator are a formal specification of a system LSI and "know-how data" specifying what generation strategies to apply. Outputs are lists of verification items, which are intermediate-level descriptions to be expanded, in the end, to a huge number of test vectors. A compiler for Agda is developed for this project as the outputs are large even at this intermediate level. The number of know-hows implemented is so far limited, but the generator is designed to be extensible by test engineers.

Comparing our formal approach with the one traditional in Renesas Technology Corp., we conclude that our approach has the following advantages.

1. The cost of verification-item generation is reduced. In particular, the time needed for reflecting a specification change becomes one fifth of the traditional method.
2. Formalization makes generation procedures explicit. This makes it easier to improve the quality of tests such as checks for omissions of critical items.

参考文献

[1] Agda: An Interactive Proof Editor. http://unit.aist.go.jp/cvs/Agda/

[2] Xavier Leroy. Formal certification of a compiler back-end, or: programming a compiler with a proof assistant. In 33rd symposium Principles of Programming Languages, pages 42-54. ACM Press, 2006.

[3] Bengt Nordström, Kent Petersson and Jan M. Smith. Programming in Martin-Löf's Type Theory. Oxford University Press, 1990.

システム LSI 仕様の形式化と検証項目の自動生成 (in English)
(算譜科学研究速報)
発行日：2009 年 7 月 9 日
編集・発行：独立行政法人 産業技術総合研究所 (システム検証研究センター)
同連絡先：〒560-0083 大阪府豊中市新千里西町 1-2-14 三井住友海上千里ビル 5F
TEL：06-4863-5025
e-mail：informatics-inquiry@m.aist.go.jp

Formalization of System LSI Specification and Automatic Generation of Verification Items
(Programming Science Technical Report)
9 July 2009
(Research Center for Verification and Semantics (CVS))
National Institute of Advanced Industrial Science and Technology (AIST)
5F Mitsui Sumitomo Kaijo Senri Bldg., 1-2-14, Shinsenrinishi-machi, Toyonaka, Osaka 560-0083 Japan
TEL：+81-6-4863-5025
e-mail：informatics-inquiry@m.aist.go.jp

AIST01-J00022-96