Static Analysis for SystemC with Scoot From Verification to Simulation

Author(s):
Blanc, Nicolas

Publication Date:
2010

Permanent Link:

https://doi.org/10.3929/ethz-a-006042704 →

DISS. ETH No. 18851

Static Analysis for SystemC with Scoot: From Verification to Simulation

A dissertation submitted to

ETH ZÜRICH

for the degree of

DOCTOR OF SCIENCES

presented by

NICOLAS BLANC

Master of Science, EPFL

born on the 24th of June 1979
in Sion / VS

accepted on the recommendation of

Prof. Dr. Gustavo Alonso,
Dr. Daniel Kröning, and
Dr. Luke Ong

2010

Summary

SystemC is a description language for computer systems that is based on C++. The language is used for modeling electronic devices at arbitrary levels of abstraction. In particular, SystemC can describe models with both hardware and software aspects. As today's electronic designs are incredibly large and complex, validation of new products remains time consuming as ever; hence the need to keep improving state-of-the-art verification techniques.

We describe verification solutions for SystemC to improve reliability of computer devices by combining formal reasoning and simulation at high-level of abstraction. In particular, our research indicates that formal methods have applications beyond property checking, namely for code optimizations, simulation, and testing. We show that verification engines can be integreted into compilation flows to compute information valuable at runtime, statically.

More specifically, our results demonstrate that an application of formal engines at high abstraction level is practical provided that:

1. the system is partitioned into "small" and independent verification tasks, as monolithic approaches are more subject to scalability issues,

2. the procedure takes advantage of today's parallel computers to distribute those tasks among many cores and

3. authorizes trading precision for time: It is often preferable to provide partial results rapidly instead of complete results after a long period of time.

We investigate solutions for the analysis of systems with hardware and software components, exploring how formal methods, classic static analysis techniques, and simulation can be combined to improve state-of-the-art verification. We have implemented these techniques in a compiler prototype for the analysis of SystemC designs called SCOOT. Given a SystemC model written in C++, SCOOT statically extracts the module hierarchy and generates a model in an intermediate representation that is suitable for formal verification and simulation. SCOOT is the first compiler for SystemC that uses formal methods to improve simulation performance and coverage. In particular, experimental evaluation indicates that SCOOT can significantly speedup the execution of models relevant for industry.

Résumé

SystemC est un langage de modélisation de systèmes informatiques basé sur C++ pouvant être utilisé pour des spécifications de circuits logiques à haut et bas niveaux d'abstraction. En particulier, SystemC est souvent employé pour modéliser des systèmes qui comportent à la fois des aspects matériels et logiciels.

Comme la taille et la complexité des nouvelles générations de systèmes informatiques embarqués continuent de croître, les processus de validation ralentissent toujours plus les cycles de développement de ces nouveaux produits; d'où la nécessité de poursuivre des recherches afin d'améliorer les techniques de vérification actuelles.

Dans cette thèse, nous décrivons des solutions nouvelles de validation de modèles pour SystemC qui combinent à la fois des techniques classiques de simulation et des méthodes récentes de raisonnement formel à haut niveau d'abstraction. En particulier, nos recherches montrent que le domaine d'application des méthodes formelles va au-delà de la simple vérification et que ces méthodes peuvent aussi fournir des résultats utiles pour l'optimisation de code et la simulation.

De manière plus spécifique, nos travaux démontrent qu'une utilisation pragmatique des méthodes formelles à haut niveau d'abstraction est possible dès lors que:

1. le système est partitionné en de petites tâches indépendantes de vérification, car les approches de vérification monolithiques sont fortement sujettes au phénomène d'explosion exponentielle de l'espace de recherche

2. ces tâches sont exécutées en parallèle sur plusieurs machines

3. le degré de précision de l'analyse peut être ajusté: il est souvent préférable de procéder rapidement avec des résultats partiels que d'être bloqué en attente de résultats complets.

Nous présentons des solutions pour l'analyse des systèmes informatiques qui comportent des aspects logiciels et matériels, en explorant comment les développements récents des méthodes formelles peuvent être combinés aux techniques d'analyse statique et dynamique traditionnelles afin d'améliorer les procédures de vérification actuelles. Nous avons construit un prototype de compilateur pour SystemC nommé SCOOT pour évaluer ces solutions. Etant donné un modèle SystemC écrit en C++, SCOOT peut extraire de manière statique des informations concernant la structure du système et construire un modèle dans une représentation intermédiaire adéquate pour la vérification formelle et la simulation. SCOOT est le premier compilateur pour SystemC qui utilise des méthodes formelles pour améliorer les performances de simulation.