

Verification of Event-Based Synchronization of SpecC Description Using Difference Decision Diagrams

Thanyapat Sakunkonchak and Masahiro Fujita

Department of Electronic Engineering, University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
thong@cad.t.u-tokyo.ac.jp, fujita@ee.t.u-tokyo.ac.jp

We introduce an on-going work that tries to develop a technique for the verification of synchronization issues in SpecC language, a C-based system level description language. It is summarized as followed.

- Based on the concept of the boolean program, the SpecC source description is translated into *boolean SpecC*. We concern only with the synchronization-related statements, hence, statements like conditional branching statements (**if/else**) and synchronization semantics containing statements (**notify/wait**) will be sustained. The other statements will be abstracted away.
- From the boolean SpecC, we translate it again into the C++ language where the relationship among the statements in the boolean SpecC is expressed in terms of inequalities, e.g. $stmnt1 < stmnt2 < \dots$, and proportional logic on conditional variables introduced when translating from SpecC to boolean SpecC. This will be incorporating with the DDD package (able to handle the inequalities expression and proportional variables) for verifying the synchronization of the original SpecC. The verification result must be one of these: *Satisfied* or *Unsatisfied*. When the result is *Satisfied*, the verification process terminates with the satisfied result. Otherwise, we have to further verify the result whether the synchronization is really unsatisfied or unable to determine because of the limitation of our verification tools.
- We are verifying the result from the previous stage. Some other tools like SVC or Program Slicing will be used in order to help checking and tracing to the *Unsatisfied* verification result. The causes of the *Unsatisfied* result are either the synchronization of the original source is wrong or there is too much abstraction when translating from SpecC to boolean SpecC. This stage is under implementation.

Let us note that the verification process should be completely automatic. As a final remark, the proposed technique clearly defines synchronization semantics of SpecC descriptions, which is one of most important issues for system level description languages.