A Component Composition Framework for Intellectual Property Blocks in Embedded Systems

Rajesh Gupta
UCSD

Embedded system design -- particularly for system-chip implementations -- is increasing dominated by abstraction and reuse of hardware and software intellectual property blocks. However, a posteriori validation of component compositions is a difficult problem. Techniques are needed to ensure correctness through the composition process itself. But what does it mean for a component to be composable? We, of course, know answer this question for circuit blocks, software pieces, but seem to get stuck when dealing with blocks above RT level. Component composition frameworks (CCF) have received attention by researchers looking for capabilities to improve the modeling and the reuse of semiconductor IP.

We approach system-level composition of IP or virtual components as composition of high level description of a virtual architecture that describes blocks, interconnections, buses, and their abstract functionalities. This virtual description can be then instantiated with real IP components from existing IP repository. The instantiated system model will work only when the IP blocks are composable at their interfaces. Matching of data types generally constitutes a first step towards automated component selection and matching. However, data-type matching is not enough to construct correct composition of IP blocks. For instance, the behaviors of the interfaces may not match even if type composition is clean. Conversely, port type composition may sometimes be too restrictive in composition of similar interfaces. Earlier research has developed techniques for creating ‘wrappers’ around components to make them compatible. These wrappers usually help resolving data type matching, signal/pin matching etc. This project goes one step further in capturing the interface behaviors using behavioral types. We propose an assume-guarantee type system that uses type inference and type justification as a part of the IP composition process. This will allow automation of many synthesis and validation tasks that are currently performed in an ad hoc manner as a design is composed. For instance, behavioral type mismatches are resolved by automatic synthesis of protocol adapters for types that are supported by a polymorphic type management system. Similarly, we will use interface automata formalism (for example, as an efficient representation of the set of possible sequence of interface events) to describe the structure of the behavior at the interfaces. The assume-guarantee type system proposed here will allow us to apply model
checking techniques to achieve the goal of creating provably correct component composition.

There are both theoretical and practical implications of the proposed research. Formally, the problem of IP modeling and compositional correctness are open issues that require fundamental rethinking of the capabilities that the language provides (by means of static type checking) and their relationship to validation tools (both static as well as dynamic verification through assertions). Fortunately, recent advances in formal verification and the growing acceptance of high-level modeling makes this project feasible and relevant to the future growth of the microelectronic system design and exploration. In practical terms, the outcomes of this research will be directly applicable to – and indeed demonstrated using IP component libraries using – the evolving industry standards on modeling language(s) for system functionality as well as validation constraints.