Verilog HDL, a Replacement for CSP

Arash Saifhashemi

In our presentation we show how it is possible to use a standard HDL language, such as Verilog HDL along with some PLI (Programming Language Interface) routines, to describe asynchronous circuits at the behavioral level (CSP level). Using PLI, We have made it possible to make communication actions in Verilog HDL become abstract. Therefore, the designer can write a CSP program in Verilog-HDL without being worried about the implementation of communication actions. From the designer’s view, there is no need to implement a handshaking protocol or to define any additional signal. All the designer has to do is to issue READ and WRITE actions on channels, which are standard Verilog input/output ports (or nets) without any limitation on the size of the channels. READ and WRITE actions on a single channel will be executed in parallel and they correspond (match) with each other.

We believe that other asynchronous designers can use the same approach and simulate their CSP programs on standard Verilog simulators instead of ‘ad-hoc’ CSP simulators. The suggested algorithm and PLI routines are so simple (less than 100 lines of code in our case) that other asynchronous designers can not only implement it quickly, but also they can further improve and optimize the method. One can customize the way that CSP features are added to Verilog HDL using PLI. Furthermore, it does not need any preprocessing or extra tool. The designer can write the program in Verilog HDL from the very beginning steps of the design, while two important CSP features are available: channel communications become atomic actions, and fine-grained concurrency within processes is available.

As another advantage, designers can use a single test-bench at all levels of abstraction. It means that using some PLI routines one can write a test-bench at behavioral level (CSP level) and use it with a circuit that is described at any level of abstraction, from behavioral down to switch level.

More importantly, it lets designers to exchange their ‘standard’ codes with each other. At the worst case, all they have to do is to attach their PLI routines to their Verilog code.

We believe that this method will replace developing new CSP-like languages and new asynchronous modeling tools.