Testing Asynchronous Circuits in a Synchronous Environment

Frank te Beest, Ad Peeters, Hans Kerkhoff
MESA+ Research Institute, Testable Design and Test Group
University of Twente, Enschede, The Netherlands
(in co-operation with Philips Research)
E-mail: F.J.tebeest@el.utwente.nl
Phone: +31-53-4894007

To enable the use of asynchronous modules or cores in large synchronous chips, the testing of these modules is of vital importance. Not only should the asynchronous modules be tested, these tests have to be compatible with synchronous test tools and methods as much as possible. Since the test methods used for synchronous circuits are usually based on full-scan, we have implemented a full scan tool for asynchronous circuits designed with the Tangram toolset from Philips Research, that is compatible with synchronous test tools.

In the test world there are two paradigms that enable tests to be created on a modular basis. One of the oldest is the macro test flow, dating back to the 1980s. Its target was to implement defect oriented testing of different circuit styles (logic, register files, SRAM etc.). For every circuit style, dedicated test generation tools are used to generate the test patterns. Key to the macro test concept is the distinction between test protocol and test patterns. A test protocol contains information about how to apply a test pattern to a circuit under test. The test protocol concept is very useful when the circuit is built of several (hierarchical) blocks. To test a block embedded in several other blocks, a protocol can be expanded to a higher level by finding access paths through the surrounding blocks. This continues until every protocol is expanded to the top level. Then the protocols can be scheduled for optimal test time and finally they are assembled with the test patterns resulting in the final test for the circuit.

During the 1990s the core test paradigm was introduced. The core based test method targets the reuse of modules and the accompanying tests. Such a module can for example internally be tested with the macro test method. The core based methods standardizes the test access to the cores, enabling the mixing of cores from various vendors.

At present we have developed a full-scan method for asynchronous circuit that is compatible with the macro test paradigm. The test pattern generation, protocol expansion and the test assembly tools are reused. Asynchronous circuits generated by Tangram can be easily divided into a control part and the data path. The latches in the data path are clocked by local clocks originating from the control part. This coupling can result in interference between an external test clock and the local clock. To solve this problem we chose to test the control and the data path separately and use the macro test expansion concept to generate a top level test for the asynchronous circuit. Other (synchronous or memory) macro blocks can be directly expanded in the same step. It was also necessary to include test a clock and scan inputs to the C-elements used in the control. All modifications to the circuit will result in a large area overhead and reduced performance. However if the asynchronous circuit will comprise only a small part of the total chip, the influence on the total area and performance is acceptable.

The current (unoptimized) version of the full-scan tool leads to roughly a 100% increase in area. After the implementation of various optimizations a decrease to ~60% for the full scan method will become feasible. Further reduction is possible by using partial scan techniques, but this is probably not possible without dedicated tool support. The overall chip overhead is of course determined by the percentage of asynchronous logic on it and depending on the application between 10 and 20% seems to be acceptable.