Using STIL to Describe Embedded Core Test Requirements

Brion Keller
IBM

Abstract

STIL was developed primarily to facilitate the exchange of test patterns from the source of the patterns to automatic test equipment (ATE) that will apply the patterns to the circuit. It seems a natural extension to allow preexisting tests for an embedded core to be provided in STIL format. Additional information is considered essential to allow the re-use of such test patterns. The IEEE P1500 Core Test Language task force has chosen to extend STIL so that a description of an embedded core’s test requirements can be supplied in STIL syntax – possibly along with the actual test patterns for the embedded core.

A Core Test Language

The basic problem of trying to generate tests for a black box embedded core is very difficult. Even when preexisting tests are available for the core, it is important to know how to get the test stimulus into the core from the chip pins and how to get the core responses visible at the chip pins.

The traditional approach is to provide a MUX on each input and output of a core that will allow the core I/O pins to become directly controlled and observed at chip I/O pins. The core is also isolated from other logic and cores on the chip. In many cases, multiplexing all core pins out to chip pins is overkill. For cores with built-in self test (BIST), it may be possible to test the core using only a few of the core’s pins. In some cases cores have more pins than the chip – making it impossible to simultaneously MUX all core pins to the chip I/Os.

This is where the requirement comes from for describing each core pin that will be involved in the test application for the core. By describing the core test requirements for each pin, we can determine which pins require sensitized paths to be established to chip pins. It is also possible that some core pins can be accessed through use of scan chains within the surrounding logic. Yet another possibility is that scan chains internal to the core can be connected into the middle of scan chains in the surrounding logic. These and many other possibilities exist.

To allow core integration tools and core test pattern generation tools to know how to deal with each pins’ requirements, a language is being proposed by the Core Test Language task force of the IEEE P1500 Embedded Core Test standard working group. The intent is for this language to describe various core test requirements in terms of core pins and general operational modes of the core.

Because the members of the Core Test Language (CTL) task force believed that STIL would become a highly used standard for passing test vectors, we made the tacit assumption that preexisting tests for a core will be supplied in a STIL format. As such, it makes sense that we allow the additional contextual information about the core test requirements be specified in the same syntactic format and even perhaps within the same file as the test vectors themselves.

Thus, by extending the semantic constructs of STIL, we hope to provide a single mechanism to allow descriptions of:

1. Operational (test) modes for the core and their purpose.
2. Isolation requirements for each pin for each mode.
3. Protocols for applying the tests to the core.
4. The test vectors for the core.
5. Core boundary scan function (black box) to specify how logic surrounding the core can be observed and provided with stimulus from the core pins.

STIL is an approved IEEE standard. CTL, as an extension to STIL, is still being defined. STIL must become well used and understood in order for its use in core testing to become accepted.