Using Assertions in an Active Way to Design and Verify Interface between Analog and Digital Blocks

HyunDon Kim, Samsung Electronics, Korea
hyundon.kim@samsung.com
Jiang Long, Mentor Graphics Corporation, USA
jiang_long@mentor.com
Wesley Park, Mentor Graphics Corporation, USA
wesley_park@mentor.com
ChiHo Cha, Samsung Electronics, Korea
ch.cha@samsung.com
JaeBeom Kim, Samsung Electronics, Korea
jay.b.kim@samsung.com
Byeong Min, Samsung Electronics, Korea
byeong.min@samsung.com
KyuMyung Choi, Samsung Electronics, Korea
kmchoi@samsung.com

Developing a specification is a very important process to reduce time and resource that are required to implement a design. However, developing a quality specification can not be achieved without passing difficult hurdles. A specification tends to be modified during its development and although its implementation has already been started, many times, re-development of a specification is necessary to incorporate any problems of implementation. It usually ends up with restarting its implementation to cope with the new specification.

Developing an interface specification between analog and digital blocks is more vulnerable to modification than developing a specification for pure digital blocks because the behavior of analog blocks is difficult to be modeled at the stage of developing its specification. Because of this frequent modification, we have experienced inefficiency through the entire design and verification process that must be improved by all means.

We have selected assertions as a fundamental tool to enhance our conventional design and verification process for analog and digital mixed design. An assertion is known as an executable specification of design intent. If we have a set of assertions that represent an entire specification, it enables us to verify the various behaviors of a design in cycle-accurate level, which is required by verification on mixed designs. However, the role of assertions is limited in the current design and verification environment because it is useless until its implementation counterpart, a design, is completed and ready to be verified.

In this paper, we use assertions in an active way along with enabling technologies. We use assertions to check the validity of the interface specification without any design. To enable this, we visualize the behaviors of a set of assertions without design. Once a set of assertions is ready for the interface between analog and digital blocks, we use it to generate testbenches that can verify both analog and digital blocks before its counterpart

is ready to be verified. With this approach, each analog and digital design team can start and continue to implement its own part without waiting for the other. It reduces the overall turnaround time that could have been incurred by any mismatches between analog and digital implementation.

Technologies

We utilize synthesis and formal technique with assertion behavior formulation, nondeterministic modeling, trace reduction, don't-care value computation, trace enumeration and reverse engineering of important assertion states etc. to visualize assertion behavior, compare and debug assertions through waveforms annotated with assertion activity symbols. The integration of these techniques presents an intuitive characterization and visualization of temporal behavior of assertions. The work establishes a methodological foundation that expedites the adoption and deployment of assertions in the industrial settings.

Methodologies

The following methodologies help designer use assertions in a more effective and active way.

1. Assertion characterization through automatic assertion activity scenario generation. Waveforms enhanced with assertion activity symbol provide an intuitive illustrate of a particular assertion evaluation behavior. However, each temporal assertion construct corresponds to a set of such waveforms, while a single waveform can only capture a fracture of the design intent. We utilize the tool to automatically generate a set of such waveforms visualizing passing and failing scenarios.

2. Quite so often, there is a necessity to change assertions during debugging or specification changes. It is natural to ask the question what is the exact difference between two assertions. We utilize assertion comparison technique with waveform visualization to concretely characterize the difference between similar assertions. Formal tools are utilized to compute relationship between two assertions. This process expedites the debug process tremendously.

3. One of the major tasks is to describe the interface protocol and quickly construct test bench to simulate with the analog design. The automatic generation and visualization of legal traces through a set of assertions helps identify incorrect behavior and unspecified scenario helps adding assertions quickly and effectively.