VALIDATION OF A VERILOG ASSERTION LIBRARY

Leonardo Firmino Otaviano
CS Dept. - UFMG - Brazil
otaviano@dcc.ufmg.br

Leandro Maciel F. G. de Paula
CS Dept. - UFMG - Brazil
leandro@dcc.ufmg.br

José Augusto M. Nacif
CS Dept. - UFMG - Brazil
jnacif@dcc.ufmg.br

Claudionor N. Coelho Jr.
CS Dept. - UFMG - Brazil
coelho@dcc.ufmg.br

Antônio Otávio Fernandes
CS Dept. - UFMG - Brazil
otavio@dcc.ufmg.br

Abstract

Assertion based verification is a technique that locates a failure during a design simulation without the need to propagate the failure to the I/O pins. An extention of this technique, where you can synthesize assertions in the final IC is called run time verification. In this paper, we present the validation process of an assertion library modified to support this run time verification. We present results of tests comparing the behavior of the original and modified library.

1. Introduction

The objective of this work is to validate an assertion library used in integrated circuit (IC) design. The goal of a system validation is to assure that it behaves as it was specified. There are several verification techniques and the use of each one depends on IC type and complexity. The system verification and validation is becoming more important every day, about to consume approximately 70% of a design development effort [1]. Today, the number of verification engineers is approximately the double of design engineers. These data show the importance and the demand of work required during validation processes. This paper is outlined as follows. Section 2 describes an architecture to extend OVL to support on chip run time debug, while Section 3 presents the method used to validate the modified library. In Section 4, we present the results. Finally, in Section 5 we conclude with our remarks.

2. On Chip Run Time Verification

Due to the increasing complexity on today's circuits designs, the use of conventional validation tools isn't always enough to assure that an IC is free of errors. Then, it was proposed a run time error detection methodology. This methodology is based in assertions provided by OVL (Open Verification Library). OVL initially was developed to be used with verification tools during the design simulation stage. Table 1 lists assertions included in OVL.

assert_always
assert_always_on_edge
assert_change
assert_cycle_sequence
assert_decrement
assert_delta
assert_even_parity
assert_fifo_index
assert_frame
assert_handshake
assert_implication
assert_increment
assert_never
assert_next
assert_no_overflow
assert_no_transition
assert_no_underflow
assert_odd_parity
assert_one_cold
assert_one_hot
assert_proposition
assert_quiescent_state
assert_range
assert_time
assert_transition
assert_unchange
assert_width
assert_win_change
assert_win_unchange
assert_window
assert_zero_one_hot

Table 1. OVL assertions

This run time architecture was created to be incorporated in the IC to provide assertion failure information through I/O pins. The modified library was implemented in verilog and it was based in three components: a set of modified assertions based on OVL; an assertion processor, which is an appointed circuit to process the assertions results and to take the appropriate action; and an automatic routing mechanism that directs the assertions information for the assertion processor. Further details on this methodology can be found on [2, 3, 4].

3. Validating

The strategy that will be adopted for validation is called functional verification. Its main intention is to assure that the project implementation matches its specification. Verification can only show the presence of errors, not their absence. The black box approach, which will be used to validate the modified library, don't have direct access to the internal structures of the circuit that is being verified. In this technique, stimuli are generated, exercising the circuits' inputs. Then, the output is compared with expected results. This method is illustrated in Figure 1. More aggressive techniques, like white box verification, require more observability of the circuit internal structures, but these techniques should be used in high complexity circuits. As assertions are simple circuits, the black box verification is more appropriate.

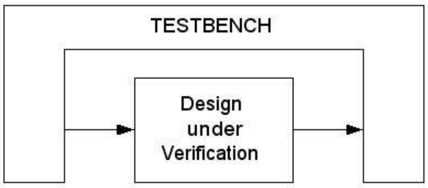

Figure 1. Application of stimuli and verification of the output of a device under verification

A more detailed model of the proposed verification methodology can be observed in Figure 2. In this model, it is possible to verify the correctness of the simulation results in parallel with stimuli generation when the design is running. The stimuli generation will be made randomly and the outputs will be compared. So, the two outputs should present the same value to assure that DUV (Design Under Verification) behaves like the reference model.

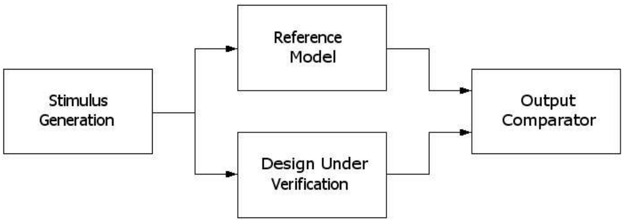

Figure 2. Architecture proposal for verification of the modified version of the assertion library

4. Results

Comparing the Figure 2, our reference model is the OVL library and the modified library is the DUV. To compare the outputs, we needed to adapt the reference model creating a variable to store its error status. This variable will be compared with the error output signal of the DUV. We implemented testbenches to validate the assertions listed in Table 1. Each program has assertions in pairs, the DUV (modified library) and the reference model (OVL library). Both tests the same expression. As mentioned in section 3, we generated random stimuli to circuits' inputs and monitored the outputs. Some signals like clock and reset weren't generated randomly. They were kept with valid values. This procedure was taken for each assertion. Each simulation had lasted about one day and the tests didn't find errors.

5. Conclusions

In this paper we described a validation process of a verilog assertion library. To do so we used a black box technique with reference model. No errors were found and the assertion library was successfully validated.

6. Acknowledgements

This paper is partially supported under grants CNPq 183064/2003-8 and CNPq PNM 830107/2002-9.

References

[1] J. Bergeron. *Writing Testbenches Functional Verification of HDL Models*. Kluwer Academic Publishers, 2000.

[2] M. C. de Oliveira, J. A. Nacif, C. C. Jr., and A. O. Fernandes. Xroach: A tool for generation of embedded assertions. *Student Forum 2003, Chip in Sampa, Brazil*, 2003.

[3] J. A. Nacif, F. M. de Paula, H. Foster, C. C. Jr., F. C. Sica, D. C. da Silva, and A. O. Fernandes. An assertion library for on-chip white-box verification at run-time. *Proceedings of Latin American Test WorkShop*, 2003.

[4] J. A. Nacif, F. M. de Paula, H. Foster, C. C. Jr., F. C. Sica, D. C. da Silva, and A. O. Fernandes. The chip is ready. am i done? on-chip verification using assertion processors. *Proceedings of VLSI-SoC 2003, Darmstadt, Germany*, 2003.