A Scheduling Method for Hierarchical Testability Based on Test Environment Generation Results

Jun Nishimaki
Graduate School of Industrial Technology
Nihon University
Chiba, JAPAN
nishimaki.jun@gmail.com

Toshinori Hosokawa
College of Industrial Technology
Nihon University
Chiba, JAPAN
hosokawa.toshinori@nihon-u.ac.jp

Hideo Fujiwara
Faculty of Informatics
Osaka Gakuin University
Osaka, JAPAN
fujiwara@ogu.ac.jp

***Abstract*— A binding method for hierarchical testability has been proposed to increase the number of testable operational units in hierarchical testing using behavioral level circuits [2]. The method aims to synthesize many operational units which can be tested by generated test sequences using hierarchical test generation. In this paper, we propose a scheduling method for hierarchical testability to increase the efficiency of the binding method [2]. Experimental results show that the combination of our proposed scheduling method and the binding method [2] improves fault coverage by 11% on average in hierarchical testing.**

Keywords—hierarchical test generation; test environments; behavioral synthesis; scheduling; binding

I. INTRODUCTION

A hierarchical test generation method using behavioral level circuits has been proposed as an efficient test generation method for sequential circuits [1]. This method generates test sequences for each operation in a control data flow graph (CDFG). Test sequences generated for an operation *op* can test the operational unit that is assigned to *op*. In this paper, an operational unit which has test sequences generated by hierarchical test generation is referred to as a *hierarchically testable operational unit*, and an operational unit which does not have test sequences generated by hierarchical test generation is referred to as a *hierarchically untestable operational unit*. It is important to increase the number of hierarchically testable operational units to achieve high fault coverage in hierarchical testing.

A binding method for hierarchical testability has been proposed to increase the number of hierarchically testable operational units [2]. In this paper, we propose a scheduling method for hierarchical testability to increase the number of hierarchically testable operational units in the binding method for hierarchical testability.

II. HIERARCHICAL TEST GENERATION

The hierarchical test generation method [1] tries to obtain test environments for operations in a CDFG. The method can generate test sequences for operations with a test environment by substituting test patterns for test environments. On the other hand, the method cannot generate test sequences for operations without a test environment.

In hierarchical testing using behavioral level circuits, test sequences generated for an operation *op* can test the operational unit that is assigned to *op* at operational unit binding. Hence, an operational unit which is assigned to at least one operation with a test environment is a hierarchically testable operational unit. On the other hand, an operational unit which is assigned to only operations without a test environment is a hierarchically untestable operational unit.

III. BINDING FOR HIERARCHICAL TESTABILITY

The binding method for hierarchical testability proposed in [2] aims to test hierarchically untestable operational units by test sequences generated for other operational units. To test hierarchically untestable operational units by test sequences generated for other operational units is referred to as *rescue*.

Let U be a hierarchically untestable operational unit. To rescue U, the following two conditions must be satisfied.

Condition 1: U must be in idle state at a control step of rescue, i.e., no operation where U is assigned is executed at a control step of rescue.

Condition 2: Test pattern registers must be assigned to inputs of U, where a test pattern register is a register which is assigned to input variables of operations with a test environment.

978-1-4673-9659-2/16/$31.00 ©2016 IEEE

It is important to generate scheduled-CDFGs (SCDFG) which are easy to satisfy above two conditions to increase the number of rescued hierarchically untestable operational units.

IV. SCHEDULING FOR HIERARCHICAL TESTABILITY

In this section, we propose a scheduling method for hierarchical testability to increase the number of rescued hierarchically untestable operational units.

The *force directed scheduling* (FDS) method has been proposed as an efficient scheduling method [3]. Our proposed method is based on FDS, and includes two more scheduling strategies for rescue as follows.

A. Strategy for Idle States of Operational units

Our first scheduling strategy is one for the *condition1* described in the above section. In the binding method [2], rescue is performed at control steps assigned to operations with a test environment. Here, a control step assigned to an operation with a test environment is referred to as a *rescue control step*. The first strategy decreases the number of operations assigned to rescue control steps. As a result, the number of operational units which are in idle state at rescue control steps increases. Hence, the binding method [2] becomes easy to satisfy Condition 1.

B. Strategy for Test Pattern Registers

Our second scheduling strategy is one for Condition 2 described in the above section. The binding method [2] must assign test pattern registers to inputs of rescue target operational units. Hence, lifetime length of test pattern registers should be as short as possible. The second strategy shortens lifetime length of input variables for operations with a test environment. As a result, lifetime of test pattern registers becomes short. Hence, the binding method [2] becomes easy to satisfy Condition 2.

V. EXPERIMENTAL RESULTS

We performed the experiments for three circuits to evaluate the effectiveness of our proposed method. In the experiments, first, we performed hierarchical test generation to obtain test sequences. This hierarchical test generation is based on the method of [1]. Next, we synthesized RTL circuits from CDFG. To clarify the effectiveness of our proposed method, we compared two cases. In one case, the RTL circuits are synthesized by using our proposed scheduling method and the binding method [2], and in the other case, the RTL circuits are synthesized by using FDS and the binding method [2]. Next,

Table I Experimental Results

Circuit	#OU	Proposed			FDS		
		#Testable OU	FC(%)	Area	#Testable OU	FC(%)	Area
ARF	6	6	99.96	35,900	4	84.37	35,491
FIR+MPEG	7	7	99.96	38,497	4	91.78	37,859
DWT+MPEG	7	7	99.96	36,816	5	91.47	35,617

we synthesized gate-level circuits by Design Compiler™. Finally, we evaluated the fault coverage by using test sequences generated by the hierarchical test generation and gate-level circuits synthesized by Design Compiler™.

Table I shows the experimental results. We used three circuits ARF [4], FIR+MPEG, and DWT+MPEG in the experiments. FIR+MPEG and DWT+MPEG are ones which are combined two benchmarks from [4].

In FDS, hierarchically untestable operational units were synthesized in all circuits. On the other hand, in our proposed scheduling method, all operational units were synthesized as hierarchically testable operational units for all circuits. Therefore, our proposed scheduling method achieved higher fault coverage for all circuits compared to FDS. The fault coverage was improved by approximately 11% on average.

The increase in the circuit area was suppressed to approximately 2% on average.

VI. CONCLUSION

This paper proposed a scheduling method for hierarchical testability to increase the number of hierarchically testable operational units in the binding method proposed in [2]. In the experiments, by performing our proposed scheduling method, the fault coverage was improved approximately 11% on average compared to FDS. Future work includes evaluating the proposed method for practical circuits.

REFERENCES

[1] S. Bhatia, and N. K. Jha, "Integration of Hierarchical Test Generation with Behavioral Synthesis of Controller and Data Path Circuits," IEEE trans. on Very Large Scale Integration Systems, Vol. 6, No. 4, Dec. 1998.

[2] J. Nishimaki, T. Hosokawa, and H. Fujiwara, "A Binding Method for Hierarchical Testability Using Results of Test Environment Generation," 2nd IEEE Workshop on Design Automation for Understanding Hardware Designs (DUHDe'15), Mar. 2015.

[3] P. G. Paulin and J. P. Knight, "Force-directed scheduling for the behavioral synthesis of ASIC's," IEEE Trans. on Computer-Aided Design, Vol.8, No.6, pp661-679, Jun 1989.

[4] S. P. Mohanty, N. Ranganathan, E. Kougianos, and P. Patra, Low-Power High-Level Synthesis for Nanoscale CMOS Circuits, Springer, 2008.

Acknowledgements

This work was supported in part by Japan Society for the Promotion of Science (JSPS) under Grants-in-Aid for Science Research C (No.26330071).