8th IEEE European Test Workshop, pp.207-208, May 2003

# Test Synthesis for Datapaths using Datapath-Controller Functions*

Michiko Inoue Kazuhiro Suzuki† Hiroyuki Okamoto‡
Hideo Fujiwara
Graduate School of Information Science
Nara Institute of Science and Technology

## Abstract

*This paper proposes a test syntheses methods for datapaths. The proposed method goes on design-for-testability while generating control sequences for justification and propagation at register-transfer level. Since the method fully utilizes functions of controllers as well as datapaths, it achieves small overhead for both area and delay.*

## 1. Introduction

High level test synthesis ([1, 3, 4, 5, 6, 7]) has potentiality to reduce test cost drastically by utilization of high level information or abstraction. This paper simultaneously considers DFT and test generation at RTL based on *strong testability*[7]. This testability guarantees all the modules in a datapath to be tested hierarchically. The hierachical test generation[2] combines test generation for combinational modules at gate level and path search for justification to and propagation from the modules at RTL.

We consider an RTL circuit composed of a controller and a datapath. We augment an original controller so as to provide test plans, which are sequences control signals that form control and observation pathes for modules under test, while making a datapath strongly testable. We consder the following test synthesis problem.

**Input:** an RTL circuit (a datapath and a controller)
**Output:** an augmented RTL circuit (a strongly testable datapath and a contller capable of providing test plans for each module) and test plans
**Objective:** minimization of area overhead

We can augment a controller by adding transitions, states, and primary inputs. For a datapath, we add hold function to registers or thru function to modules.

*This work is supported in part by Japan Society for the Promotion of Science (JSPS) under the Grant-in-Aid for Science Research No.14658092
†Currently, SANYO Electric
‡Curretly, Matsushita Electric Industrial

## 2. Overview

The proposed method finds justification and propagation paths for each module while adding some DFT elements if it is difficult to establish such paths. We augment a controller so that it can provide test plans as well as control signals for normal operations. For a datapath, we augment hold functions and thru finctions if necessary. Any augmentation requires some area overhead, and our objective is to find the augmentation with the minimum area overhead. For a datapath, we can easily estimate area overhead required to augment thru function or hold function. For a controller, we introduce a cost function to estimate such area overhead. We propose a branch-and-bound algorithm that implicitly searchs any possible sequences of transitions and finds the sequence that provides a test plan with the minimum augmentation. This method can enhance testability with small cost in the case where control and observation paths, which cannot be provided by the original controller but required for a test plan, can be realized only by changing some control signal.

In addition, our method has many advantages of test synthesis methods based on the strong testability. We can use a combinatinal ATPG and hence achieve high fault coverage and high fault efficiency. Test application time is much shorter than scan design since we do not need scan shifting, and furthermore, we can apply at-speed testing.

## 3. Experiments

We applied the proposed method to 4 benchmark circuits GCD, 4th Jaumann Wave Filter(JWF), 3rd Lattice Wave Filter(LWF) and Paulin. We compared the results with full scan design reported in [4]. Table1 shows the characteristic of these circuits. The columns #state, #status, #control, #PI, #PO, |bit|, #Reg. and #Mod. denote the numbers of states, status signal lines, control signal lines, primary inputs and primary outputs, the bit-width of a datapath, the numbers of registers and modules, respectively. The experiments used a logic synthesis tool AutoLogicII(Mentor Graphics) and an

Table 1. Circuit characteristics

| | controller | | | | datapath | | | | | | total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| circuit | #state | #status | #control | #gate | #PI | #PO | bit | #Reg. | #Mod. | #gate | #gate |
| GCD | 4 | 3 | 7 | 169 | 32 | 16 | 16 | 3 | 1 | 1350 | 1524 |
| JWF | 8 | 0 | 38 | 199 | 80 | 80 | 16 | 14 | 3 | 6671 | 6875 |
| LWF | 4 | 0 | 8 | 57 | 32 | 32 | 16 | 5 | 3 | 1924 | 1986 |
| Paulin | 6 | 0 | 16 | 123 | 64 | 64 | 32 | 7 | 4 | 24833 | 24965 |

Table 3. Test generation

| | our method + controller DFT[4] | | | | full scan | |
|---|---|---|---|---|---|---|
| circuit | C-ATPG (sec) | test plan gen. (sec) | fault eff. (%) | test appl. (cycle) | test gen. (sec) | test appl. (cycle) |
| GCD | 0.70 | 3.71 | 100.0 | 774 | 171.51 | 6629 |
| JWF | 0.54 | 156.62 | 100.0 | 2779 | 2.88 | 20519 |
| LWF | 0.51 | 0.96 | 100.0 | 394 | 0.47 | 4066 |
| Paulin | 1.80 | 252.82 | 100.0 | 2703 | 4.68 | 16187 |

Table 2. Hardware overhead

| circuit | area overhead(%) | | | | | | #test pin | |
|---|---|---|---|---|---|---|---|---|
| | our method + controller DFT[4] | | | | | full scan | ours | full scan |
| | C | DP | C' | MUX | total | | | |
| GCD | 9.2 | 0 | 1.1 | 7.5 | 17.8 | 39.7 | 4 | 3 |
| JWF | 3.4 | 2.9 | 0.4 | 4.7 | 11.4 | 28.4 | 4 | 3 |
| LWF | 2.2 | 6.1 | 0.8 | 4.0 | 14.1 | 42.3 | 3 | 3 |
| Paulin | 0.6 | 2.2 | 0.2 | 0.6 | 3.7 | 6.3 | 4 | 3 |

ATPG tool TestGen(Synopsys) on SunBlade1000 (Sun Microsystems). The columns #gate denote the areas of circuits in gates after logic synthesis.

Table2 shows hardware overhead, where C and DP denote area overhead of augmentation for controllers and datapaths, respectively. After the test synthesis for datapaths, we apply the DFT method for controllers proposed in [5]. The column C' denote area overhead for this controller DFT, and MUX denote area overhead for MUXs to isolate controllers from datapath including MUXs to bring the values of status signals to the primary output of datapaths. The combination of two methods enables test generation using combinational ATPG for the whole RTL circuits.

For these circuits, our method achieves small area overhead, especially for Paulin. Table3 shows the result of test generation and test plan generation by our method and full scan design. We generated tests for both datapaths and controllers. Since our method enables us to apply combinational ATPG for each module, time for combinational ATPG is short while achieving 100% fault efficiency. Test application time is much shorter than full scan design. However, for some circuits, our search algorithm consumes time. This is because our algorithm implicitly searches all possible paths.

## 4. Conclusions

We proposed a test synthesis method that simultaneously considers DFT and test generation for RTL circuits. The proposed method utilizes functions of controller and datapath as much as possible. Therefore, we can achieve 100% fault efficiency with small area overhead.

## References


[1] I. Ghosh, A. Raghunathan, and N. K. Jha. Hierarchical test generation and design for testability methods for ASPPs and ASIPs. *IEEE Trans. on Computer Aided Design*, 18(3):357–370, March 1997.

[2] B. T. Murray and J. P. Hayes. Hierarchical test generation using precomputed tests for modules. *IEEE Trans. on Computer Aided Design*, 9(6):594–603, June 1990.

[3] S. Nagai, S. Ohtake, and H. Fujiwara. A method of design for hierarchical testability for data flow intensive circuits at register-transfer level. *Journal of Information Processing Society of Japan*, 43(5):1278–1289, May. 2002.

[4] S. Ohtake, T. Masuzawa, and H. Fujiwara. A non-scan approach to DFT for controllers achieving 100% fault efficiency. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 16(5):553–566, Oct. 2000.

[5] S. Ohtake, H. Wada, T. Masuzawa, and H. Fujiwara. A non-scan DFT method at register-transfer level to achieve complete fault efficiency. In *Asia and South Pacific Design Automation Conference 2000 (ASP-DAC2000)*, pages 599–604, Jan. 2000.

[6] S. Ravi, G. Lakshminarayana, and N. K. Jha. TAO: Regular expression based register-transfer level testability analysis and optimization. *IEEE Trans. on VLSI Systems*, 9(11):357–370, Dec. 2001.

[7] H. Wada, T. Masuzawa, K. K. Saluja, and H. Fujiwara. Design for strong testability of RTL data paths to provide complete fault efficiency. In *13th International Conference on VLSI Design*, pages 300–305, Jan. 2000.