IEEE International High Level Design Validation and Test Workshop 2009 (IEEE HLDVT 2009), pp. 160-165, Nov. 2009.

RTL DFT Techniques to Enhance Defect Coverage for Functional Test Sequences

Hongxia Fang[1], Krishnendu Chakrabarty[1] and Hideo Fujiwara[2]

[1]ECE Dept., Duke University, Durham, NC, USA {hf12, krish}@ee.duke.edu

[2]Nara Institute of Science and Technology, Nara, Japan fujiwara@is.naist.jp

***Abstract*— Functional test sequences are often used in manufacturing testing to target defects that are not detected by structural test. However, they suffer from low defect coverage since they are mostly derived in practice from existing design-verification test sequences. Therefore, there is a need to increase their effectiveness using design-for-testablity (DFT) techniques. We present a DFT method that uses the register-transfer level (RTL) output deviations metric to select observation points for an RTL design and a given functional test sequences. Simulation results for six $ITC'99$ circuits show that the proposed method outperforms two baseline methods for two gate-level coverage metrics, namely bridging and gate-equivalent fault coverage. Moreover, by inserting a small subset of all possible observation points using the proposed method, significant fault coverage increase is obtained for all benchmark circuits.**

Keywords: DFT, output deviations, RT-level, test-point insertion, unmodeled defects.

I. Introduction

Nanoscale CMOS technologies are leading to increasing defect rates for integrated circuits (ICs) [1] [2]. Since structural test alone is not sufficient to ensure high defect coverage, functional test is used in industry to target defects that are not detected by structural test [3]–[6]. Register transfer (RT)-level fault modeling, design-for-testablity (DFT), test generation and test evaluation are therefore of considerable interest [7]–[10].

A number of methods have been presented in the literature for test generation at RT-level. In [8], the authors proposed test generation based on a genetic algorithm (GA), targeting statement coverage as the quality metric. In [11]–[14], the authors used pre-computed test sets for RT-level modules (adders, shifters, etc.) to derive test vectors for the complete design. In [15], the authors presented a spectral method for generating tests using RT-level faults, which has the potential to detect almost the same number of faults as using gate-level test generation. In [16], [17], the authors proposed a fault-independent test generation method for state-observable finite state machines (FSMs) to increase the defect coverage.

To increase the testability of the complete design and to ease RT-level test generation, various DFT methods at RT-level have also been proposed. The most common methods are based on full-scan or partial scan. However, a scan-based DFT technique leads to long test application time and it is less useful for at-speeding testing. On the other hand, non-scan DFT technique [18]–[23] offer low test application time and they facilitate at-speed testing. In [18], non-scan DFT techniques are proposed to increase the testability of RT-level designs. In [19], the authors presented a method called orthogonal scan. It uses functional datapath flow for test data, instead of traditional scan-path flow; therefore, it reduces test application time. In [20], a technique was proposed to improve the hierarchical testability of the data path, which can aid hierarchical test generation. In [21], the authors presented a DFT technique for extracting functional control- and data-flow information from RT-level description and illustrated its use in design for hierarchical testability. In [22], the authors presented a method based on strong testability, which exploits the inherent characteristic of datapaths to guarantee the existence of test plans (sequences of control signals) for each hardware element in the datapath. Compared to the full-scan technique, this method can facilitate at-speed testing and reduce test application time. However, it introduces hardware and delay overhead. To reduce overhead, the authors proposed a linear-depth time-bounded testability-based DFT method in [23]. It ensures the existence of a linear-depth time expansion for any testable fault and it offers lower hardware overhead than the method in [22].

All the RT-level DFT methods described above attempt to increase the testability of the design to ease subsequent RT-level test generation. However, the functional test sequences for manufacturing test are derived in practice from existing verification test sequences. These test sequences are generated by designers using manual or semi-automated means [24]–[26]. However, they often suffer from low defect coverage since they are mostly derived in practice from existing design-verification test sequences. Therefore, DFT techniques are needed to increase the effectiveness of these test sequences. Despite the large body of published work on RTL testing, prior work on RTL DFT has not been targeted towards increasing the defect coverage of existing functional test sequences.

In this paper, we address the problem of improving the defect coverage of given functional test sequences for an RT-level design. The proposed method adopts the RT-level deviation metric from [27] to select the most appropriate observation test points. The deviation metric at RT-level has been defined and used in [27] for grading functional test sequences. The proposed RTL DFT approach can be used to insert both control points and observation points. The observation points provide more propagation paths to primary outputs for errors due to faults in the circuit, while the control points provide greater freedom in setting internal lines to desired values for fault activation and error propagation. For the control-point selection problem, we can use the notion of output deviations at RT-level to identify locations where control points can provide the maximum increase in defect coverage for a given functional test sequence. However, in this work, we limit ourselves to the selection of observation points.

Simulation results for six $ITC'99$ circuits show that the proposed method outperforms two baseline methods for two gate-

level coverage metrics, namely bridging and gate-equivalent fault coverage. Moreover, by inserting a small subset of all possible observation points using the proposed method, significant fault coverage increase is obtained for all benchmark circuits. Since functional test sequences are used to target unmodeled defects, especially when they are used in conjunction with structural testing for modeled faults, we do not use stuck-at and transition fault coverage to evaluate test effectiveness.

II. Problem formulation

We first formulate the problem being tackled in this paper.

Given:

- RT-level description for a design and a functional test sequence $\mathcal{S}$;
- A practical upper limit n on the number of observation points that can be added.

Goal: Determine the best set of n observation points that maximizes the effectiveness of the functional test sequence $\mathcal{S}$.

The functional test sequences are typically derived manually and used for design verification, or semi-automatically generated by RT-level test generation methods. These instructions can be instruction-based for processors or application-based for application specific integrated circuit (ASIC) cores such as an MPEG decoder. They can be in the format of high-level instructions or commands, or in the format of binary bit streams.

To increase testability, we can insert an observation point for each register output. We can obtain the highest defect coverage by inserting the maximum number of observation points. However, it is impractical to do so due to the associated hardware and timing overhead. In fact, the number of observation points that can be added is limited in practice. For a given upper limit n, the challenge is to determine the best set of n observation points such that we can maximizes the defect coverage of the given functional test sequence. Our main premise is that RT-level output deviation can be used as a metric to guide observation-point selection.

III. Observation-point selection

In this section, we first define the new concept of RT-level internal deviations. Next, we analyze the factors that determine observation-point selection. Finally, we present the observation-point selection algorithm based on RT-level deviations.

A. RT-level internal deviations

The RT-level output deviation [27] is defined to be a measure of the likelihood that error is manifested at a primary output. Here we define the RT-level internal deviation to be a measure of the likelihood of error being manifested at an internal register node, which means error being manifested at one or more bits of register outputs. In the calculation of RT-level output deviation, a transition in a register is meaningful only when it is propagated to a primary output. On the other hand, in the calculation of RT-level internal deviation, we do not care whether a transition in a register is propagated to a primary output. The method for calculating internal deviations for register can also be used to calculate internal deviations for each bit of a register.

B. Analysis of factors that determine observation-point selection

The selection of observation points is determined by three factors: RT-level internal deviations of registers, observability values of registers, and the topological relationship between registers. In this work, we only consider the insertion of observation points at outputs of registers.

For a register Reg, we have the following attributes attached with it: $I_{dev}(Reg)$, $O_{dev}(Reg)$, $obs(Reg)$, to represent its internal deviation, output deviation, and observability value, separately.

For two registers $Reg1$ and $Reg2$, when two attributes are close in value, we define the following observation-point-selection rules based on the third attribute:

Rule 1: If $I_{dev}(Reg1) > I_{dev}(Reg2)$, select $Reg1$;

Rule 2: If $obs(Reg1) < obs(Reg2)$, select $Reg1$;

Rule 3: If $Reg1$ is the logical predecessor of $Reg2$, select $Reg2$.

For Rule 1, the motivation is that if we select a register with higher I_{dev}, its observability will becomes 1. Thus, its O_{dev} will also becomes higher. The higher O_{dev} of this register will contribute more to the cumulative O_{dev} for the circuit. Since we have shown that the cumulative O_{dev} is a good surrogate metric for gate-level fault coverage [27], we expect to obtain better gate-level fault coverage when we select a register with higher I_{dev}.

For Rule 2, when two registers do not have a predecessor/successor relationship with each other, obviously we should select the register with lower observability. For Rule 3, if we select $Reg1$, $obs(Reg1)$ will become 1 but this will not contribute to the increase of observability of $Reg2$; if we select $Reg2$, $obs(Reg2)$ will become 1 and $obs(Reg1)$ will also be increased due to the predecessor relationship between $Reg1$ and $Reg2$. Therefore, it is possible that the selection of $Reg2$ yields better results than the selection of $Reg1$, i.e., the cumulative observability after the insertion of observation point on $Reg2$ is higher than for $Reg1$.

Rule 2 and Rule 3 are in conflict with each other on the observability attribute. Rule 2 selects a register with lower observability while Rule 3 selects a register with higher observability. In this work, we assume that Rule 3 is given higher priority than Rule 2.

We use RT-level output deviations to guide the selection of observation points. We have determined that we should select a register with higher I_{dev}. Since O_{dev} is proportional to I_{dev} and obs, if I_{dev} factor contributes more to O_{dev}, we should select the register with higher O_{dev}. Also, by selecting a register with higher O_{dev}, we are implicitly satisfying the predecessor relationship rule: for two registers $Reg1$ and $Reg2$ whose I_{dev} values are comparable, if $Reg1$ is the predecessor of $Reg2$, we have $obs(Reg1) < obs(Reg2)$ and $O_{dev}(Reg1) < O_{dev}(Reg2)$. Then we will not select $Reg1$, which is in accordance with Rule 3.

C. RT-level deviation based observation-point selection

Based on the RT-level output deviations, we have developed a method for selecting best set of n (where n is a user-specified parameter) observation points for a given RT-level design and a

given functional test sequence. In the selecting of observation-points, we target the specific bits of a register. The calculation of I_{dev}, O_{dev}, obs for a register is carried out for each bit of a register. The selection procedure is as following:

- Step 0: Set the candidate list to be all bits of registers that do not directly drive a primary output.
- Step 1: Derive the topology information for the design and save this information in a look-up table. Obtain the weight vector, observability vector, and TCs for each register bit, and calculate RT-level output deviations for each register bit.
- Step 2: Select a register bit with the highest output deviations as an observation point. Remove this selected register bit from the candidate list.
- Step 3: If the number of selected observation points reaches n, terminate the selection procedure.
- Step 4: Update the observability vector using the inserted observation point (selected in Step 2) and the topology information. Re-calculate output deviations for each register bit using the updated observability vector. Go to Step 2.

In Step 1, the topology information of the design can be extracted using a design analysis tool, e.g., Design Compiler from Synopsys. It only needs to be determined once and it can be saved in a look-up table for subsequent use. In Step 4, after selecting and inserting an observation point, we need to update the observability vector because the observability of its upstream nodes will also be enhanced. There is no need to recompute TCs since these depend only on the functional test sequence, and they are not affected by the observation points. There is also no need to re-calculate the weight vector.

After the n observation points have been selected, they are inserted in the original RT-level design. The modified RTL design is synthesized to a gate-level netlist. To insert an observation point, we simply need to connect it directly to a new primary output. An alternative method is to use only one additional primary output and connect all observation points to this primary output through XOR gates (space compactor). By doing so, we can reduce the number of extra primary outputs to one. However, this method will lead to lower fault coverage due to error masking.

IV. Experimental results

We evaluated the efficiency of the proposed RT-level observation-point selection method by performing experiments on six $ITC'99$ [8] circuits. These circuits are translated into Verilog format and are taken as the experimental vehicles. The functional test sequences are generated using the RT-level test generation method from [8]. Our goal is to show that the RT-level deviation-based observation-point selection method can provide higher defect coverage than other baseline methods. Here, defect coverage is estimated in terms of the following two gate level coverage metrics:

- enhanced bridging fault coverage estimate ($BCE+$) [28] [29];
- gate-exhaustive (GE) score (GE score is defined as the number of observed input combinations of gates) [30] [31].

Since functional test sequences are usually used to target unmodeled defects that are not detected by structural test, we considered metrics $BCE+$ and GE score, which are more representative of unmodeled defect coverage, comparing to traditional stuck-at fault coverage and transition fault coverage. The GE score is defined as the number of the observed input combinations of gates. Here, "observed" implies that the gate output is sensitized to at least one of the primary outputs. We first compare the gate-level fault coverage for the original design to the design with all observation points inserted. Next we show the gate-level fault coverage for different observation-point selection methods.

TABLE I. Gate-level $BCE+$ of the design before and after inserting all observation points.

Circuit	Original design	Design with all observation points	
	BCE+%	#OP	BCE+%
$b09$	45.58	27	70.13
$b10$	28.04	14	55.07
$b12$	29.91	115	33.52
$b13$	23.11	43	47.12
$b14$	74.52	161	81.23
$b15$	4.4	347	10.63

A. Experimental setup

All experiments were performed on a 64-bit Linux server with 4 GB memory. Synopsys Verilog Compiler (VCS) was used to run Verilog simulation and compute the deviations. The Flextest tool was used to run gate-level fault simulation. Design Compiler (DC) from Synopsys was used to synthesize the RT-level descriptions as gate-level netlists and extract the gate-level information for calculating the weight vector. For synthesis, we used the library for Cadence 180nm technology. All other programs were implemented in C++ codes or Perl scripts.

B. Comparison of gate-level fault coverage for the original design to the design with all observation points inserted

Table I compares the gate-level fault coverage ($BCE+$) for the original design to the design will all observation points inserted. The parameter $BCE+\%$ indicates the gate-level fault coverage for bridging fault estimate. #OP lists the number of observation points. Table II compares the gate-level GE score for the original design to the design with all observation points inserted.

From these two tables, we can see that the gate-level fault coverage is not very high even when all observation points are inserted. There are two possible reasons for this: one reason is that the design suffers form low controllability. The other reason is that the quality of the given functional test sequences are not so effective for modeled fault. We can increase the gate-level fault coverage by improving the quality of functional test sequences or by inserting control points to the design. However, we focus here only on selection of observation points so that the given functional test sequences can be made more useful for manufacturing test. Therefore, it is of interest to determine the maximum gate-level fault coverage when all possible observation points are inserted, and to normalize the fault coverage to this maximum value when we evaluate the impact of inserting a subset of all possible observation points.

TABLE II. Gate-level GE score of the design before and after inserting all observation points.

Circuit	Original design	Design with all observation points	
	GE score	#OP	GE score
$b09$	121	27	173
$b10$	132	14	330
$b12$	889	115	1005
$b13$	257	43	483
$b14$	8601	161	8934
$b15$	806	347	1987

C. Comparison of normalized gate-level fault coverage for different observation-point selection methods

By considering the fault coverage of a design with all observation points inserted to be 100%, we normalize the fault coverage of designs with a smaller number of observation points. Similarly, the normalized GE score is obtained by taking the GE score of a design with all observation points inserted as the reference. In this section, we compare the normalized gate-level fault coverage and normalized GE score for different observation-point selection methods.

For each circuit, we select the same number of n (for various values of n) observation points using different methods. Results for normalized gate-level fault coverage and normalized GE score are shown in the Figure 1-2. We compare the proposed method to [32] and to a baseline random observation-point insertion method. An automatic method to select internal observation signals for design verification was proposed in recent work [32]. Since this method is also applicable for observation-point selection in manufacturing test, we take it as an example of recent related work.

The results show that the proposed method outperforms the two baseline methods for all six circuits. By inserting a small fraction of all possible observation points using the proposed method, significant increase in fault coverage and GE score are obtained for all circuits. For each circuit, it only costs several seconds to calculate RT-level deviations and select observation points. These results highlight the effectiveness of the RT-level, deviation-based observation-point selection method.

V. Conclusions

We have presented an RT-level output deviations metric and shown how it can used to select and insert the observation points for a given RT-level design and a functional test sequence. This DFT approach allows us to increase the effectiveness of functional test sequences (derived for pre-silicon validation) for manufacturing testing. Experiments on six $ITC'99$ benchmark circuits show that the proposed RT-level DFT method outperforms two baseline methods for enhancing defect coverage. We also show that the RT-level deviations metric allows us to select a small set of the most effective observation points.

ACKNOWLEGEMENTS

This research was supported in part by the Semiconductor Research Corporation under Contract no. 1588, and by an Invitational Fellowship from the Japan Society for the Promotion of Science. This paper is based in part on a presentation (unpublished) made at the IEEE Workshop on RTL and High Level Testing, 2009.

References

[1] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Timing-based delay test for screening small delay defects," in *Proc. Design Automation Conference*, 2006, pp. 320–325.

[2] X. Lin *et al.*, "Timing-aware atpg for high quality at-speed testing of small delay defects," in *Proc. Asian Test Symposium*, 2006, pp. 139–146.

[3] P. C. Maxwell, I. Hartanto, and L. Bentz, "Comparing functional and structural tests," in *Proc. ITC*, 2000, pp. 400–407.

[4] A. K. Vij, "Good scan= good quality level? well, it depends..." in *Proc. ITC*, 2002, p. 1195.

[5] J. Gatej *et al.*, "Evaluating ATE features in terms of test escape rates and other cost of test culprits," in *Proc. ITC*, 2002, pp. 1040–1049.

[6] J. Rearick and R. Rodgers, "Calibrating clock stretch during AC scan testing," in *Proc. ITC*, 2005, pp. 266–273.

[7] P. A. Thaker *et al.*, "Register-transfer level fault modeling and test evaluationtechniques for VLSI circuits," in *Proc. ITC*, 2000, pp. 940–949.

[8] F. Corno *et al.*, "RT-level ITC'99 benchmarks and first ATPG result," *IEEE Design & Test of Computers*, vol. 17, pp. 44–53.

[9] M. B. Santos *et al.*, "TL-based functional test generation for high defects coverage in digital systems," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 17, pp. 311–319.

[10] W. Mao and R. K. Gulati, "Improving gate level fault coverage by RTL fault grading," in *Proc. ITC*, 1996, pp. 150–159.

[11] S. Ravi and N. K. Jha, "Fast test generation for circuits with RTL and gate-level views," in *Proc. ITC*, 2001, pp. 1068–1077.

[12] I. Ghosh and M. Fujita, "Automatic test pattern generation for functional RTL circuits using assignment decision diagrams," in *Proc. Design Automation Conference*, 1999, pp. 43–48.

[13] H. Kim and J. P. Hayes, "High-coverage ATPG for datapath circuits with unimplemented blocks," in *Proc. ITC*, 1998, pp. 577–586.

[14] O. Goloubeva *et al.*, "High-level and hierarchical test sequence generation," in *Proc. HLDVT*, 2002, pp. 169–174.

[15] N. Yogi and V. D. Agrawal, "Spectral RTL test generation for gate-level stuck-at faults," in *Proc. Asian Test Symposium*, 2006, pp. 83–88.

[16] T. Hosokawa, R. Inoue, and H. Fujiwara, "Fault-dependent/independent test generation methods for state observable FSMs," in *Proc. Asian Test Symposium*, 2007, pp. 275–280.

[17] R. Inoue, T. Hosokawa, and H. Fujiwara, "A test generation method for state-observable FSMs to increase defect coverage under the test length constraint," in *Proc. Asian Test Symposium*, 2008, pp. 27–34.

[18] S. Dey *et al.*, "Non-scan design-for-testability of RT-level data paths," in *Proc. ICCAD*, 1994, pp. 640–645.

[19] R. B. Norwood and E. J. McCluskey, "Orthogonal scan: Low overhead scan for data paths," in *Proc. ITC*, 1996, pp. 659–668.

[20] I. Ghosh, A. Raghunathan, and N. K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," in *Proc. IEEE International Conference on Computer Design*, 1995, pp. 173–179.

[21] ——, "A design for testability technique for RTL circuits using control/data flow extraction," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, pp. 706–723, 1998.

[22] H. Wada, T. Masuzawa, K. K. Saluja, and H. Fujiwara, "Design for strong testability of RTL data paths to provide complete fault efficiency," in *Proc. IEEE International Conference on VLSI Design*, 2000, pp. 300–305.

[23] H. Fujiwara, H. Iwata, T. Yoneda, and C. Y. Ooi, "A nonscan design-for-testability method for register-transfer-level circuits to guarantee linear-depth time expansion models," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 1535–1544, Nov. 2008.

[24] J. P. Grossman*et al.*, "Hierarchical simulation-based verification of Anton, a special-purpose parallel machine," in *Proc. IEEE International Conference on Computer Design*, 2008, pp. 340–347.

[25] D. A. Mathaikutty *et al.*, "Model-driven test generation for system level validation," in *Proc. HLDVT*, 2007, pp. 83–90.

[26] O. Guzey and L.-C. Wang, "Coverage-directed test generation through automatic constraint extraction," in *Proc. HLDVT*, 2007, pp. 151–158.

[27] H. Fang, K. Chakrabarty, A. Jas, S. Patil, and C. Trimurti, "RT-level deviation-based grading of functional test sequences," in *Proc. VTS*, 2009, pp. 264–269.

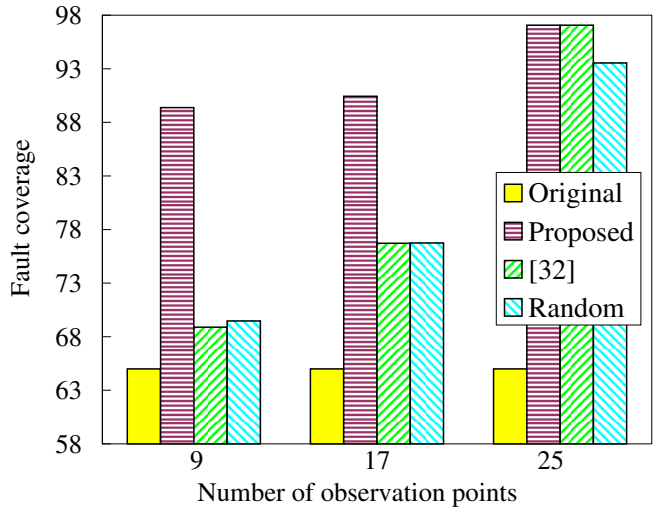
[28] B. Benware, C. Schuermyer, N. Tamarapalli, K.-H. Tsai, and J. Rajski, "Impact of multiple-detect test patterns on product quality," in *Proc. ITC*, 2003, pp. 1031–1040.

[29] H. Tang *et al.*, "Defect aware test patterns," in *Proc. DATE*, 2005, pp. 450–455.

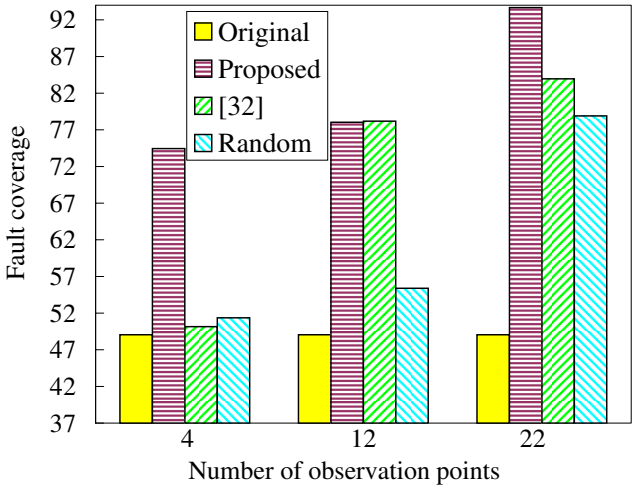
[30] K. Y. Cho, S. Mitra, and E. J. McCluskey, "Gate exhaustive testing," in *Proc. ITC*, 2005, pp. 771–777.

[31] R. Guo*et al.*, "Evaluation of test metrics: Stuck-at, bridge coverage estimate and gate exhaustive," in *Proc. VTS*, 2006, pp. 66 – 71.

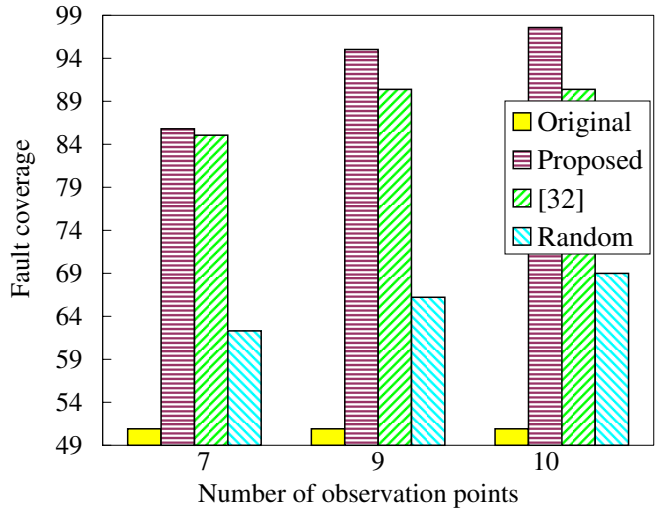
[32] T. Lv, H. Li, and X. Li, "Automatic selection of internal observation signals for design verification," in *Proc. VTS*, 2009, pp. 203–208.



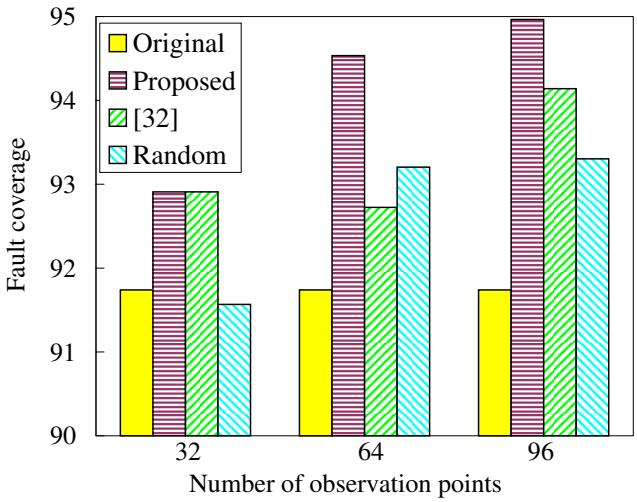
b09



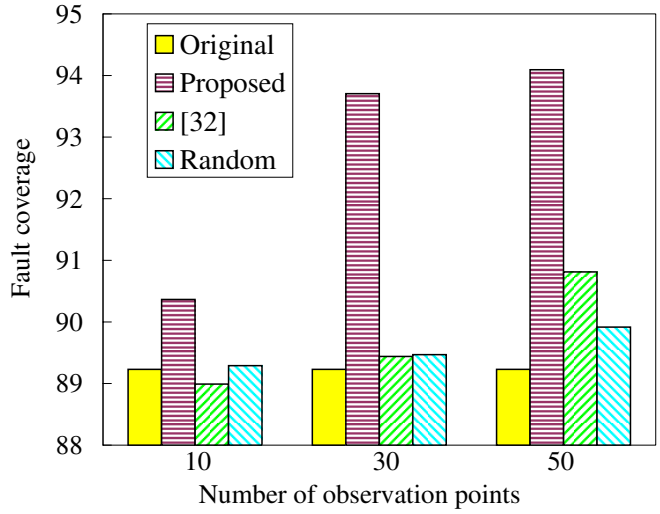
b13



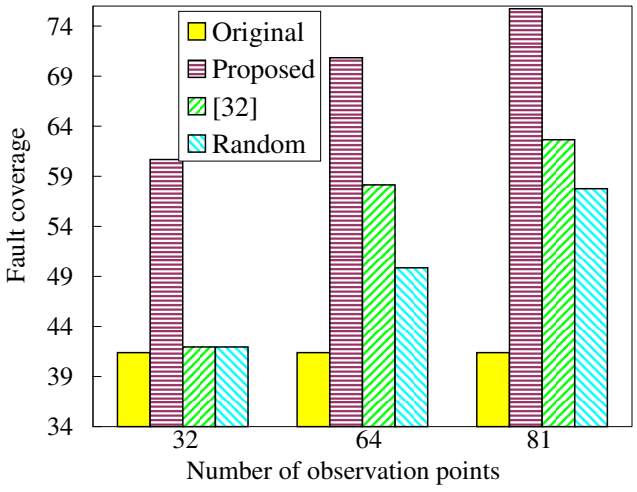
b10



b14

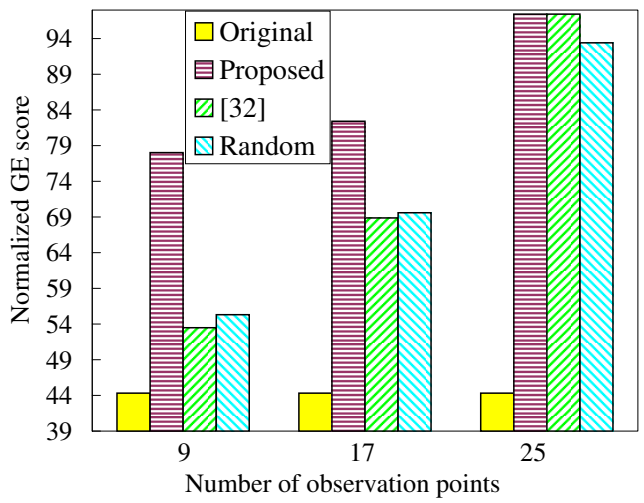


b12

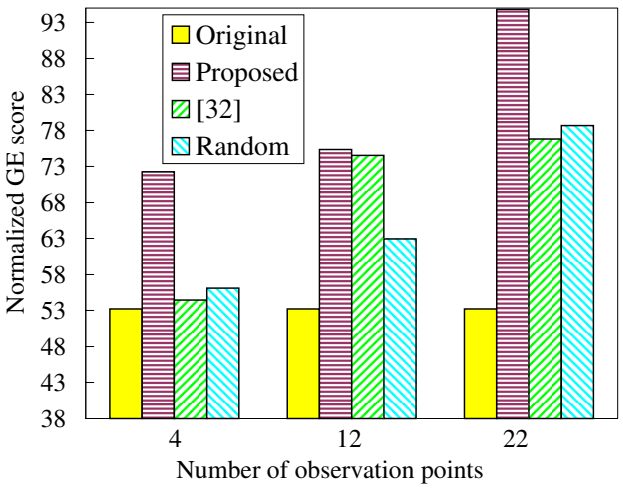


b15

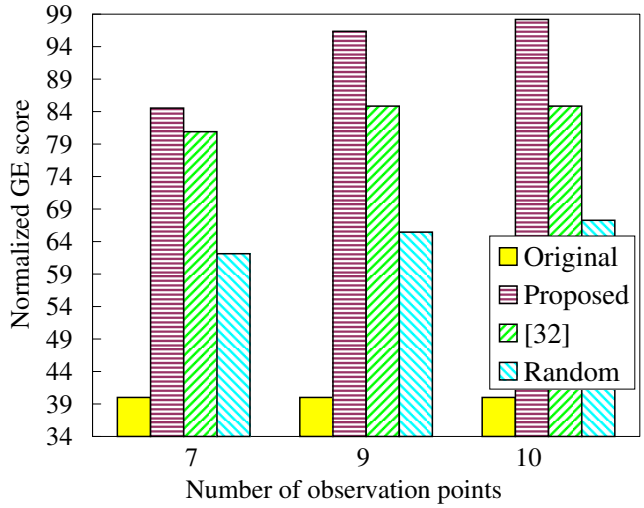
Fig. 1. Results on the gate-level normalized $BCE+$ metric.



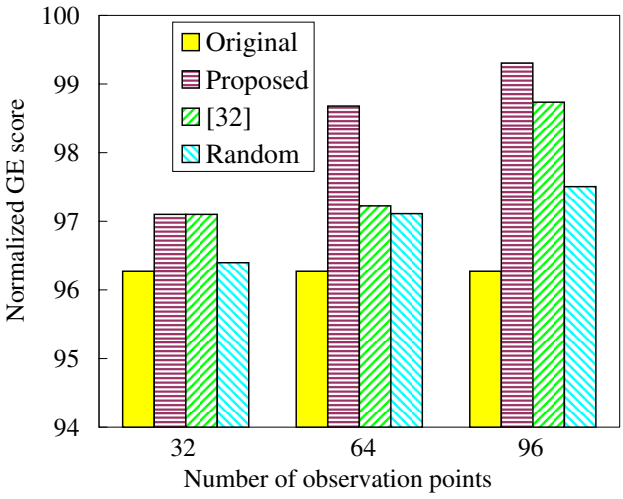
b09



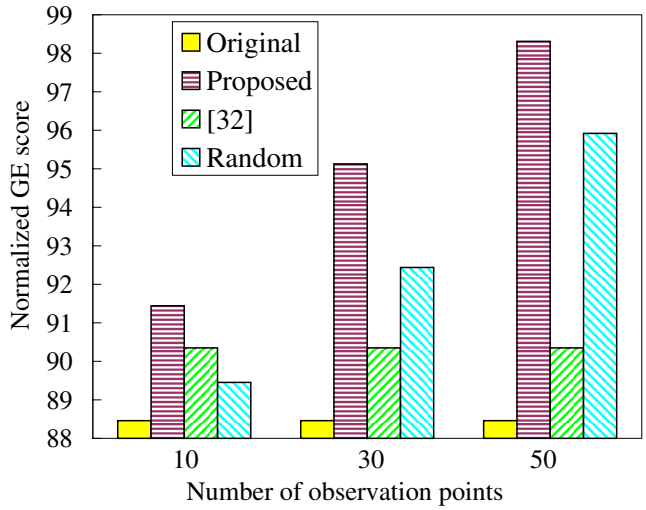
b13



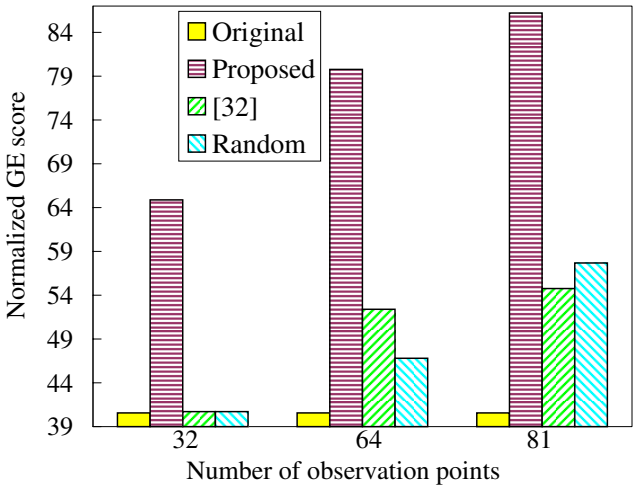
b10



b14



b12



b15

Fig. 2. Results on gate-level normalized GE score.