A Testability Analysis Method for Register-Transfer Level Descriptions

Mizuki TAKAHASHI, Ryoji SAKURAI, Hiroaki NODA, and Takashi KAMBE

Precision Technology Development Center, SHARP Corporation
Tenri, Nara 632 Japan
Tel: +81-7436-5-2531, Fax: +81-7436-5-4968
e-mail: {mizuki,sakurai,roba,kambe}@edag.ptdg.sharp.co.jp

Abstract— In this paper, we propose a new testability analysis method for Register-Transfer Level(RTL) descriptions. The proposed method is based on the idea of testability analysis in terms of data-flow and control structure which can be extracted from RTL designs. We analyze testability of RTL descriptions with more testability measures than those of conventional gate-level testability, so that the method provides information for design for testability(DFT). We have implemented the presented method and experimental results show that we can reduce circuit cost for test and achieve highly testable circuits by DFT using our RTL testability analysis.

I. Introduction

The rapid growth of the circuit size in a single LSI chip makes LSI testing difficult, and design for testability techniques becomes more important to guarantee high shipping quality of LSI's. The weight of test design effort to total design effort is considerably high in a large design and the quality of design for testability (DFT) is a key issue to realize short design term and low chip cost.

The most widely used DFT methodology is scan design, such as full scan[1] and partial scan[2]. Scan design is performed as a back-end design process after gate-level design by substituting flip-flops in the circuit to scan type flip-flops. This means that function level or RTL design can be done without consideration about test design, and scan design can provide highly testable design with a very small effort for test. In spite of these benefits, scan design has problems of timing and a test circuit cost (not only scan cell but also additional interconnections among scan cells to form scan paths) caused by scan insertion after gate-level design phase.

Another DFT methodology is *ad hoc*(non-scan) DFT. *Ad hoc* DFT is usually performed by adding test functionality to the RTL description or gate-level netlist, for example, adding a load function to a long counter or adding test outputs from an internal state of the circuit which is hard to observe. To do *ad hoc* DFT to the gate-level netlist, conventional gate-level testability analysis can be used, but in recent design flow, synchronous circuits are offen designed at RTL by using hardware description language such as VHDL and automatically synthesized to gate-level netlist by logic synthesis tool. In this design flow, it is difficult to do *ad hoc* test design at gate-level because designers are not familiar with the synthesized gate-level netlist.

For this reason, RTL testability analysis is necessary to do test design at RTL.

In this paper, we describe previous works on testability analysis in the next section, and we show basic concept of RTL testability analysis in Section III. Modeling of RTL operations for RTL testability analysis is shown in Section IV and calculation methods of RTL testability measures are described in Section V. We show experimental results in Section VI and we make concluding remarks in Section VII.

II. Previous Works

As a testability analysis method at gate-level for automatic test pattern generation or partial scan selection , SCOAP[3] is well known and most widely used. A number of researches on higher level testability analysis have been made, such as extension of SCOAP to RT level[4], topology or structure (feed-back loop, sequential depth) based method[5], and state machine based analysis of control circuit[4][6]. In these testability analysis methods, testability measures mean ATPG cost in the case of SCOAP. There are synthesis methods taking testability based on topology into account as a part of the objective functions. However, such methods do not provide a direct information for design for testability.

III. RTL Testability Analysis Method

The goal of our testability analysis method is to give useful information in RTL design. The conventional gate-level testability measures like SCOAP are calculated from the activation and propagation cost of each fault of gates. These are testability measures for ATPG and they are too microscopic to use for DFT because the source of testability problems is hard to locate.

Most of LSIs are designed at RTL and behavior of the circuit described at RTL is modeled in terms of higher abstracted data and operations among them, namely, data-flow model. Testability of a circuit is basically captured

ASP-DAC '97
0-89791-851-7$5.00

by controllability and observability of the inside of the circuit. Controllability and observability are related to the behavior of the circuit and testability should be analyzed on data-flow model in which behavior of the circuit can be more precisely captured than at gate-level.

Testability of a circuit on this model is evaluated by sufficiency and smoothness of data-flow. Sufficiency of data-flow is measured by the data amount which means controllability of data to arbitrary values. Smoothness of data-flow is evaluated by the implication cost to activate the data-flow. Evaluating the data-flow of a control path (the path from primary inputs to a register or an operation) gives controllability and that of an observation path (the path from a register or an operation to primary outputs) gives observability. If all data-flow in a circuit is sufficient and smooth, we consider that the circuit is easifly testable, and we propose testability measures using this concept.

When we analyze testability of a bundled signal (a word) on data-flow model, analyzing testability of each single bit signal independently can not reflect the relation among signals which compose the word. This may lead a contradicting result because the behavior of the word is not taken into account.

To avoid such a problem and to realize precise testability analysis method at RTL, we present a new testability analysis method which can utilize data-flow information at RTL and can handle a word as it is (word-based analysis). Our basic testability measure is based on the data amount which is fed to words (registers, inputs or outputs of RTL operations) .

Another advantage of RTL is that control part of the circuit can be identified. Control part is a part of a circuit which generates a condition for a data transfer and is composed of single bit operations and registers. In testability analysis of control part, necessary information is controllability for activating the data transfer, then controllability of a signal to the specific value is required, not to arbitrary values. For this reason, we partition a circuit into these two parts and apply a distinctive analysis method to each part.

A. RTL Circuit Modeling

First we introduce RTL circuit model on which testability analysis is performed.

RTL circuit model is composed of nodes which represent primary inputs/outputs, registers, constant values and RTL operations, and directed edges which represent data transfers between the nodes(Fig. 1). A register has its word length and RTL operation has the number of its input words and their word length and output word length. We divide RTL operations into two classes. One class is an exclusive operation class and the other is an intersection operation class. An exclusive operation means the operation whose output can be controlled by a single input word regardless other input words' state. An intersection operation means the operation whose output needs to be controlled by all the input words. Arithmetic operations and a logical exclusive OR operation belong to the exclusive operation class, and other logical operations and a guard operation belong to the intersection operation class.

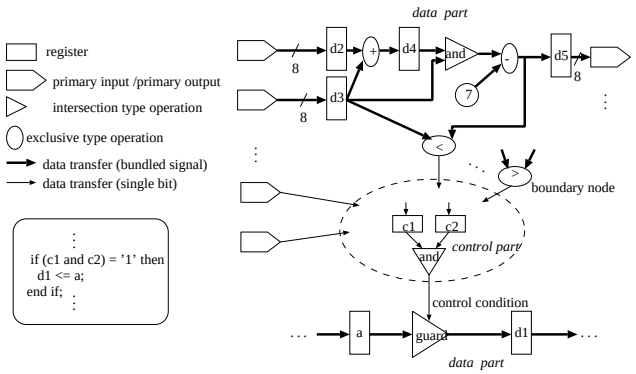

Fig. 1. RTL model

A guard operation is used to model control conditions of data transfers. Guard operation has a control input, a data input and a data output. A data transfer through a guard operation is activated when the value of a control input is '1'.

The RTL model is divided into two parts, data part and control part. Control part is a part of a circuit which generates a condition for a data transfer and is composed of single bit operations and registers. Identification of control part is done by tracing single bit registers or single bit operations from all control conditions back to primary inputs. The nodes which feed input data to the identified control part are single bit primary inputs and operations which have a single bit output word and a multiple bit input word. We refer to the latter operations as boundary nodes. The rest of the model other than control part is data part.

B. Modeling of Data Amount

In our method, we consider a word as the target of analysis, not a single bit which composes a word. By evaluating how many patterns the word can take as its value (the data amount the word has), we model behavior of the words for feeding patterns into the circuit and for observing the circuit's internal state. The data amount of the word is the number of bit which is necessary for expressing the patterns which the word can take as its values. Consider the word w with n bits. If w is fully controllable, w can take 2^n patterns as its value and the data amount of w is n. If w is not fully controllable and can take only p patterns as its values then the data amount of w is $log_2(p)$.

C. RTL Testability Measures

Controllability Measures In our testability analysis, we evaluate controllability of a register or the output of an operation by three controllability measures shown below.

1. *control data amount*
   The estimated number of patterns which the output word of the specified register or operation can take as its value.

2. *control implication data amount*
   Sum of word length of registers whose values need to be determined to control the output word of the specified register or operation from primary inputs.

3. *control step count*
   The ratio of *control implication data amount* to sum of word length of primary inputs which are used to feed the *control implication data amount* .

Control data amount gives an ability of a control path to feed data from primary inputs, *control implication data amount* gives combinational difficulties to activate the control path, and *control step count* gives sequential difficulties to activate the control path. These three controllability measures give us information for DFT from three different aspects, and enable us to locate the source of controllability problems.

For a register or the output of an operation of control part, these controllability measures are calculated for two cases. One is controllability measures for controlling the output value to '1' (controllability measure for 1-control), and the other is that for controlling the output value to '0' (controllability measure for 0-control).

Observability Measures We evaluate observability of a register or the output of an operation by four observability measures shown below.

1. *observation data amount*
   The minimum word length of the observation path through which the value of output word of the specified register or operation propagates to primary outputs.

2. *observation implication data amount*
   Sum of word length of registers whose values need to be determined to observe the output word of specified register or operation from primary outputs.

3. *observation step count*
   The ratio of *observation implication data amount* to sum of word length of primary inputs which are used to feed the *observation implication data amount* .

4. *observation path activation ratio*
   The ratio of the data amount given at an input of the observation path to the data amount observable at outputs of the observation path (primary outputs).

The first three observability measures give similar meanings to those of controllability measures and *observation path activation ratio* gives efficiency of data propagation of the observation path. These four observability measures enable us to locate the source of observability problems.

D. Outline of Testability Analysis

A primary input has *control data amount* equal to its word length and a primary output has *observation data amount* equal to its word length. This means that primary inputs and outputs of the circuit are fully testable in terms of controllability and observability, respectively. We calculate testability of the circuit by propagating these data amount into the inside of the circuit.

Controllability calculation is done by propagating data amount from primary inputs to the inside of the circuit so as to find the largest *control data amount* at each node of RTL model. Implication costs are calculated along with *control data amount* . Because the different controllability calculation methods are used for data part and control part, controllability of boundary nodes are used as controllability of both '1'-control and '0'- control in control part, and controllability for '1'-control of control conditions (control input to guard operations) is used for controllability calculation of guard operations in data part.

Observability analysis is done by calculating *observation data amount* from primary output back to the inside of the circuit by using controllability previously calculated.

IV. Data Propagation Function of Operations

To analyze testability using the data amount of the word, we define data propagation function P to model the data amount through operations and to calculate the data amount of the operation's output word from that of input words as follows.

Each operation has a dependency graph which represents functional dependencies among bits of input and output words. A dependency graph of an operation consists of nodes which represent a signal composing input/output words and edges which represent that the output bit is dependent on the input bit. An example of a

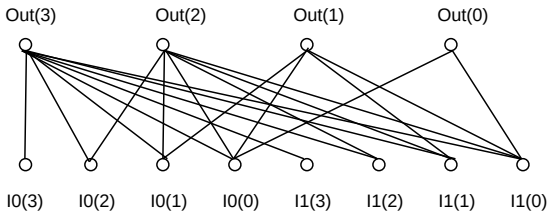

Fig. 2. An example of a dependency graph(4-bit addition operation)

dependency graph of 4-bit addition is shown in Fig. 2. 4-bit input words are $I0$ and $I1$, and output word is Out. $W(3)$ is the most significant bit and $W(0)$ is the least significant bit of a word W.

Consider the RTL operation OP which has N input word $in_i (0 \leq i \leq N-1)$ and output word out. d_i denotes the word length of each input word in_i ($\boldsymbol{d} = (d_0, d_1, ..., d_{N-1})$) and d_{out} denotes the word length of output word out(Fig. 3). x_i gives the data amount of the input word in_i ($\boldsymbol{x} = (x_0, x_1, ..., x_{N-1})$). OP also has an attribute $\boldsymbol{a}$ ($= (a_0, a_1, ..., a_{N-1})$). a_i is the average activation data amount of in_i and indicates the influence of other input words to propagate the data amount of in_i to the output word. a_i is calculated from the dependency graph of the operation OP as follows :

$$a_i = (\#\text{ of edges not connected to } in_i)/d_{out}$$

For example, the average activation data amount of input I0 in Fig. 2 is 2.5.

First, we define I which represents the data amount propagating from input words to the output word of the operation.

$$I(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}) = \sum_{0 \leq i \leq N-1} (x_i \cdot Infl_i(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a})) \qquad (1)$$

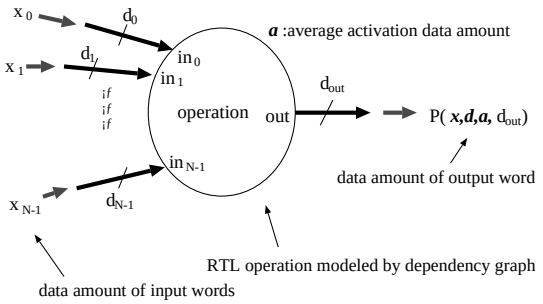

Fig. 3. Data propagation function

where

$$Infl_i(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}) = 2^{a_i \cdot (SX_i/SD_i - 1.0)} \quad (2)$$

and $SX_i = \sum_{0 \le j \le N-1, j \ne i} x_j$, $SD_i = \sum_{0 \le j \le N-1, j \ne i} d_j$.

Next we define the data amount of the output word propagating from all input words considering the word length of the output word.

$$P_0(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}, d_{out}) = \min(I(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}), d_{out}) \quad (3)$$

$$P_1(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}, d_{out}) = \max(I(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}) - \max(IWL - d_{out}, 0), 0) \quad (4)$$

where $IWL = \sum_{0 \le i \le N-1}(d_i)$.

P_0 is calculated based on the model that d_{out} of I is enough to make the data amount of output word to be equal to d_{out}. On the other hand, P_1 is calculated based on the model that I needs to be IWL (this means all the input words have the full data amount) to make the data amount of output word to be equal to d_{out}. Data propagation function P should be formulated using P_0 and P_1 to each type of operations, but we use the average of P_0 and P_1 as an approximation of P as follows :

$$P(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}, d_{out}) = \frac{P_0(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}, d_{out}) + P_1(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{a}, d_{out})}{2} \quad (5)$$

V. Calculation of RTL Testability Measures

A. Calculation of Controllability Measures of Data Part

Controllability measures of each node are calculated from primary inputs to primary outputs. In this section, we show the calculation method of these measures at each node.

In the description below, cdx_w, cid_w, and CPI_w denote *control data amount* , *control implication data amount* and a set of primary inputs used to feed cid_w respectively, where w is an input or output word of a node. *Control step count* is calculated by cid_w divided by the sum of word length of primary inputs in CPI_w. These measures for input word of the node are equal to those of output word of the node connected by data transfer edge in RTL model.

- **primary inputs** For the output word w of a primary input node p, set cdx_w and cid_w to the word length of w. Let $CPI_w = \{p\}$.
- **constant value** For the output word w of a constant value node, set cdx_w and cid_w to 0. Let $CPI_w = \phi$.
- **exclusive type operation** For the output word w of an exclusive type operation, select the most controllable input word i_k and adopt the controllability measures of i_k as the controllability measures of w. As to cdx_w, if the word length of w is less than cdx_{i_k}, cdx_w is bounded to the word length.

  The criteria for selecting the most controllable input word are as follows. The listed order of the criteria shows its priority.

  1. Select the input word which has the largest *control data amount* .
  2. Select the input word which has the smallest *control step count* .
  3. Select the input word which has the smallest *control implication data amount* .
- **intersection type operation** For the output word w of intersection type operation, set cid_w to sum of *control implication data amount* of all input words and set CPI_w to sum of primary input sets of all input words. cdx_w is calculated by the data propagation function P in Eq. (5).
- **register** For the register, its input word's controllability measures are used as its output word's controllability measures.

B. Calculation of Controllability Measures of Control Part

In the testability analysis of control part, *control implication data amount* supplied from single bit primary inputs directly to control part can be calculated more precisely by tracing required value at the primary input than *control implication data amount* calculated by the the previously described procedure for data part. To realize this calculation, we use *cpid* which is the set of quadruplet $(cid, pi, level, polarity)$. *Control implication data amount* and *control step count* are derived from *cpid*.

For the abbreviation, we use $cdx0_w$ for *control data amount* of input or output word w for 0-control, $cdx1_w$ for *control data amount* of w for 1-control, $cpid0_w$ for *cpid* of w for 0-control, $cpid1_w$ for *cpid* of w for 1-control.

By calculating 1-controllability and 0-controllability from the input nodes of control part to control conditions, finally we can get 1-controllability of control conditions.

The calculation method at each node of control part is as follows. The nodes included in control part are register nodes and logic operation nodes. On the calculation procedures for logic operations, we show only the AND operation's one and NOT operations one, but other logic operation's calculation procedures are realized by combination of these two procedures.

- **primary input** For output word w of a primary input p which feed input to control part, set $cdx0_w = cdx1_w = 1$ and initialize $cpid0_w = cpid1_w = \{(1, p, 0, 1)\}$.

· **boundary node** For output word w of a boundary node p, $cdx0_w$ and $cdx1_w$ are set to cdx_w which is previously calculated *control data amount* of w by controllability analysis for data part. $cpid0_w$ and $cpid1_w$ are also initialized by using cid_w and CPI_w and polarity components of the element of $cpid0_w$ and $cpid1_w$ are set to X.

· **AND operation** For N-input AND operation node m, let $I = \{in_0, in_1, ..., in_{N-1}\}$ be a set of input word and out be an output word of m. Controllability measures of m are calculated by the equations blow.

$$\begin{aligned} cdx0_{out} &= cdx0_k \\ cpid0_{out} &= cpid0_k \\ cdx1_{out} &= \min_{i \in I}(cdx1_i) \\ cpid1_{out} &= \bigcup_{i \in I} cpid1_i \end{aligned}$$

k is the selected input word which is the most 0-controllable. The criteria for selecting the most 0-controllable input word are as follows. The listed order of the criteria shows its priority.

1. Select the input word w which has the largest $cdx0_w$.
2. Select the input word w which has the smallest *control step count* for 0-control. This is derived from $cpid0_w$.
3. Select the input word w which has the smallest *control implication data amount* for 0-control. This is calculated by sum of cid components of quadruplets in $cpid0_w$.

· **NOT operation** For NOT operation node m, let in and out be input word and output word of m respectively. Controllability measures of m are calculated by the equations blow.

$$\begin{aligned} cdx0_{out} &= cdx1_{in} \\ cpid0_{out} &= Polarity\_inv(cpid1_{in}) \\ cdx1_{out} &= cdx0_{in} \\ cpid1_{out} &= Polarity\_inv(cpid0_{in}) \end{aligned}$$

where $Polarity\_inv$ is a procedure that inverts (1 to 0, 0 to 1, and X to X) all the *polarity* components of quadruplets in given $cpid$.

· **register** For a register node r, let in and out be input word and output word of r respectively. Controllability measures of r are calculated by the equations blow.

$$\begin{aligned} cdx0_{out} &= cdx0_{in} \\ cpid0_{out} &= Lev\_inc(cpid0_{in}) \\ cdx1_{out} &= cdx1_{in} \\ cpid1_{out} &= Lev\_inc(cpid1_{in}) \end{aligned}$$

where Lev_inc is a procedure that increases all the *level* components of quadruplets in given $cpid$ by one.

C. Calculation of Observability Measures

Observability measures of each node are calculated from primary outputs back to primary inputs after the controllability measure calculation. In this section we show the calculation method of these observability measures at each node.

In the description below, odx_w, oid_w $oact_w$, and OPI_w denote *observation data amount* , *observation implication data amount* , *observation path activation ratio* , and a set of primary inputs used to feed oid_w, respectively, where w is the input or output word of a node. *Observation step count* is calculated by oid_w divided by the sum of word length of primary inputs in OPI_w.

· **primary output** For the input word w of primary output node p, set odx_w to the word length of w. Let $oid_w = 0, oact_w = 1, OPI_w = \phi$.

· **exclusive type operation** For all input word w of an exclusive type operation p, set observability measures of w to those of output word of p. As to odx_w, if the word length of w is less than *observation data amount* of the output word, odx_w is bounded to the word length of w.

· **intersection type operation** Let p is a N-input intersection type operation, $iw_k (0 \leq k \leq N-1)$ be an input word of p and out be an output word of p. $odx_{iw_k}, oid_{iw_k}, and OPI_{iw_k}$ are calculated as follows :

$$\begin{aligned} odx_{iw_k} &= \min(odx_{out}, \text{word length of } iw_k) \\ oid_{iw_k} &= \sum_{0 \leq j \leq N-1, j \neq k} (cid_{iw_j}) + oid_{out} \\ OPI_{iw_k} &= \bigcup_{0 \leq j \leq N-1, j \neq k} (CPI_{iw_j}) \cup OPI_{out} \end{aligned}$$

In the above equations, cid_{iw_k} and CPI_{iw_k} are controllability measures defined in the previous section.

Observation path activation ratio $oact_{iw_k}$ is calculated by the equation below.

$$oact_{iw_k} = P_{iw_k} \cdot oact_{out} / d_{out}$$

where P_{iw_k} is an evaluation result of data propagation function P in Eq. (5) on the assumption that *control data amount* of iw_k is equal to word length of iw_k, d_{out} is word length of output word out.

· **register** For the register, its output word's observability measures are used as its input word's observability measures.

· **fanout point** When the output word w of a node (register or operator) is connected to multiple input words by data transfer edges, selection of the most observable input word is made and observability measures of w are set to those of the selected input word.

VI. Experimental Results

We have implemented the presented method as RTL testability analysis system.

TABLE I
EXPERIMENTAL RESULTS

circuit		DFT selection a	b	c	d	gate count (ratio to original)	# of scan FF/# of FF (scan FF%)	fault coverage	pattern length
A	original	-	-	-	-	9100 (100.0%)	0/782 (0.0%)	13.53%	56
	EX1	◯	×	×	×	9351 (102.8%)	12/782 (1.5%)	91.36%	962
	EX2	◯	◯	×	×	9640 (105.9%)	112/782 (14.3%)	98.74%	778
	EXA	-	-	-	-	9994 (109.8%)	246/782 (31.5%)	98.51%	381
B	original	-	-	-	-	6357 (100.0%)	0/473 (0.0%)	11.0%	111
	EX1	◯	×	×	×	6695 (105.3%)	85/473 (18.0%)	88.75%	10678
	EX2	◯	◯	×	×	6800 (107.0%)	113/473 (23.9%)	92.95%	11051
	EX3	◯	◯	◯	×	6956 (109.4%)	151/473 (31.9%)	98.63%	12714
	EX4	◯	◯	×	◯	6949 (109.3%)	153/473 (32.3%)	98.53%	10304
	EXA	-	-	-	-	7443 (117.1%)	284/473 (60.0%)	98.63%	1434

We show our experimental results in Table 1. In this experiment, to show the effectiveness of DFT based on our testability measures, we adopt partial scan insertion as DFT for selected register based on our testability measure, because we make comarison with the test circuit cost of automatic partial scan insertion based on conventional gate-level testability.

Circuit A (original) is described by 1035 lines of HDL and it has 99 registers, and Circuit B is described by 2044 lines of HDL and it has 85 registers.

In Table 1, EX1 to EX4 are the result of our method and EXA is the result of automatic partial scan insertion based on conventional testability. Our DFT criteria for selecting scan registers used in case EX1 to EX4 are as follows :

a Improvement of *control data amount* . Make all registers' *control data amount* to their word length.

b Improvement of *observation data amount* . Make all registers' *observation data amount* to their word length.

c Improvement of *implication data amount* . Make all registers' *control and observation implication data amount* less than 5 times of their word length.

d Improvement of *step count*. Make all registers' *control and observation step count* to 1.

In EX1 and EX2, we perform a DFT based on *control/observation data amount* . These testability measures are the most important for achieving high fault coverage and improvement of only these testability measures brings sufficient coverage in the case of circuit A. In the case of circuit B, EX2's fault coverage is less than that of EXA, then we tried improvement of *implication data amount* and *step count* in EX3 and EX4. In both EX3 and EX4, fault coverages nearly equal to that of EXA are achieved and the number of flip-flops replaced by scan flip-flops is reduced 53% compared with EXA.

VII. Conclusion

We propose a new testability analysis method for Register-Transfer Level(RTL) descriptions. This testability analysis method makes the most use of the nature of the RTL descriptions, such as behavior of words not just a single bit signal and information on datapath part and control part.

We analyze testability of RTL description in terms of various kinds of testability measures and this makes it possible to give a useful information for design for testability in RTL. Experimental results shows that DFT based on our testability measures make it possible to achieve high fault coverage with low test circuit cost.

In our experimental results, we use scan insertion to evaluate the effectiveness of our testability measures. In addition to that, if we use DFT to control conditions (which is not possible by scan insertion), the results can be greatly improved. We are going to confirm this improvement in the next step.

References

[1] E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability," in *Proc. 14th DAC.*, pp. 462–468, 1977.

[2] H-K. T. Ma, S. Devadas, A. R. Newton, and A. Sangiovanni-Vincentelli, "An Incomplete Scan Design Approach to Test Generation for Sequential Machines," in *Proc. Int. Test Conf.*, pp. 730–734, 1988.

[3] L. H. Goldstein, "Controllability/Observability Analysis of Digital Circuits," IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 685–693, Sep. 1979.

[4] Akira Motohara, *et al.*, "Design for Testability Using Register-Transfer Level Partial Scan Selection," in *Proc. of the International Conference on Computer-Aided Design*, pp. 640–645, 1994.

[5] T-C. Lee, N. K. Jha, and W. H. Wolf, "Behavioral Synthesis of Highly Testable Data Paths under the Non-Scan and Partial Scan Environments," in *Proc. 30th DAC.*, pp. 292–297, 1993.

[6] Abhijit Ghosh, Srinivas Devadas and A. Richard Newton, "Sequential Logic Synthesis for Testability Using Register-Transfer Level Descriptions," in *Proc. of International Test Conference*, pp. 274–283, 1990.