An Integer Programming Formulation for Generating High Quality Transition Tests

Tsuyoshi IWAGAKI[†] and Mineo KANEKO[†]

[†] School of Information Science, Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
E-mail: †{iwagaki,mkaneko}@jaist.ac.jp

Abstract This paper describes a test generation method to derive high quality transition tests for combinational circuits. It is known that, for a transition fault, two-pattern tests which propagate the errors to all the primary outputs reachable from the fault site can enhance the detectability of unmodeled defects. In this paper, to generate high quality transition tests, the test generation problem is formulated as a problem of integer linear programming, where a metric expressing the above fact is optimized. The proposed formulation guarantees that minimum two-pattern tests for a transition fault are generated such that the errors are observed at all the primary outputs reachable from the fault site.

Key words combinational circuit, test generation, high quality transition test, integer linear programming, minimum test set

1. Introduction

Fault coverage is a basic criterion to evaluate a given test set under a target fault model. When a target fault model is specified, test engineers try to generate tests with 100% fault coverage under the fault model. Several fault models such as the stuck-at fault model and the transition fault model are usually targeted during test generation phases. Obtained test sets are then applied to actual circuits for defect screening. However, some defective circuits can pass the screening due to the presence of unmodeled defects even though the fault coverage of the applied tests is 100%. One way to avoid this undesirable situation is to develop a dedicated fault model for such defects. However, since it is costly to do so in general, several alternatives which assume conventional fault models have been discussed to enhance the detectability of unmodeled defects [1]–[5].

Multiple-detection tests [1] have been shown to have an ability of detecting unmodeled defects. In order to clarify how effective multiple-detection tests are, some metrics were discussed in [2]–[5]. This paper focuses on the metric in [2]. In [2], the authors considered a test set for transition faults that propagates the errors of each transition fault to all the primary outputs reachable from the fault site, and showed it is effective in screening defective circuits compared to a conventional test set. To derive such a test set, some test generation procedures have been proposed in [6], [7]. The procedures in [6], [7] used a Boolean satisfiability technique with some heuristics, and an existing test generation tool, respectively. When we have a combinational circuit and a transition fault in the circuit, the following question can arise:

- What is the minimum number of two-pattern tests that detect the fault at all the primary outputs reachable from the fault site?

To the best of our knowledge, there has been no answer to this question yet. One goal of this paper is to give an answer to it. In this paper, we try to tackle this problem by using a technique of integer linear programming (ILP).

The rest of this paper is organized as follows. Section 2 gives the concept of test generation using ILP, then, in Section 3, an ILP formulation is presented to derive a minimum test set for a transition fault that meets the above property. Finally, Section 4 concludes the paper and describes our future work.

2. Preliminaries

Our test generation method is based on integer linear programming (ILP). In this section, we describe how to trans-

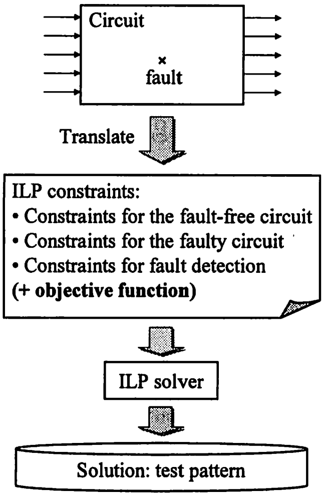
Circuit
fault
Translate
ILP constraints:
• Constraints for the fault-free circuit
• Constraints for the faulty circuit
• Constraints for fault detection
(+ objective function)
ILP solver
Solution: test pattern

Figure 1: Concept of ILP-based test generation

late the test generation problem for a transition fault in a combinational circuit into an ILP problem.

2.1 Concept of ILP-based test generation

ILP-based test generation has first been presented in [9]. Figure 1 represents the concept of ILP-based test generation. In this framework, given a combinational circuit and a fault, the circuit and the detection condition of the fault are first translated into the corresponding constraints that consist of inequalities and equalities with integer variables (especially 0-1 variables). Then, a feasible assignment to the variables that meets the constraints is obtained by an ILP solver. The assigned values of the variables that correspond to the circuit inputs form a test for the fault. If one wants to optimize some property during test generation, one can add it as an objective function to the ILP problem. In the following, we explain how to translate the test generation problem for a transition fault in a combinational circuit into an ILP problem by using some examples. More formal descriptions of ILP-based test generation can be found in [9], [10].

2.2 Transition Test Generation Using ILP

A two-pattern test for a transition fault satisfies the following two conditions.

(1) The first vector sets an appropriate value to the fault site.

(2) The second vector detects the corresponding stuck-at fault.

Since there is no correlation between the first vector and the second vector, they can be considered separately during test generation. Before describing how to generate a two-pattern test for a transition fault, we first explain how to express the circuit behavior by using ILP constraints.

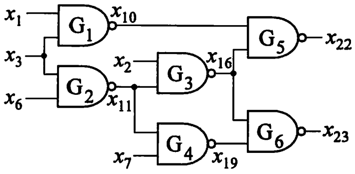

Figure 2: ISCAS '85 benchmark circuit c17

Table 1 shows inequalities in ILP constraints to express the behaviors of primitive gates with one or two inputs. In the first column of the table, y represents a gate output and each of x, x_1 and x_2 represents a gate input, where they can take '0' or '1.' A feasible assignment to the variables of the inequalities for a gate corresponds to the behavior of the gate. For example, a 2-input AND gate produces '0' if at least one input has '0.' This behavior corresponds to the first and second inequalities in Table 1. Indeed, if x_1 or x_2 takes '0,' y has to be '0' in those inequalities. Furthermore, if both inputs take '1,' the AND gate has to produce '1.' This behavior is expressed as the last inequality in the table. In this way, each gate in a combination circuit can be interpreted as inequalities in ILP constraints. Given a combinational circuit, we can obtain ILP constraints for the whole circuit by replacing each gate with its corresponding inequalities repeatedly. Now, let us consider the circuit shown in Figure 2. For example, we can obtain the following constraints for c17.

$$\mathrm{G}_1\colon\ x_1 + x_{10} \geqq 1, x_3 + x_{10} \geqq 1, -x_1 - x_3 - x_{10} \geqq -2$$
$$\mathrm{G}_2\colon\ x_3 + x_{11} \geqq 1, x_6 + x_{11} \geqq 1, -x_3 - x_6 - x_{11} \geqq -2$$
$$\mathrm{G}_3\colon\ x_2 + x_{16} \geqq 1, x_{11} + x_{16} \geqq 1, -x_2 - x_{11} - x_{16} \geqq -2$$
$$\mathrm{G}_4\colon\ x_{11} + x_{19} \geqq 1, x_7 + x_{19} \geqq 1, -x_{11} - x_7 - x_{19} \geqq -2$$
$$\mathrm{G}_5\colon\ x_{10} + x_{22} \geqq 1, x_{16} + x_{22} \geqq 1, -x_{10} - x_{16} - x_{22} \geqq -2$$
$$\mathrm{G}_6\colon\ x_{16} + x_{23} \geqq 1, x_{19} + x_{23} \geqq 1, -x_{16} - x_{19} - x_{23} \geqq -2$$

Any feasible assignment for these constraints simulates the behavior of c17. In Figure 2, when we have $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_6 = 1$ and $x_7 = 1$, the circuit behaves as follows: $x_{10} = 1$, $x_{11} = 1$, $x_{16} = 0$, $x_{19} = 0$, $x_{22} = 1$ and $x_{23} = 1$. These values satisfy the above constraints, and *vice versa*.

Given a combinational circuit C and a transition fault f in C, the following tasks are performed to generate a two-pattern test in this paper.

Table 1: Inequalities in ILP constraints expressing the behaviors of primitive gates

Gate types	Inequalities
$y = \mathrm{AND}(x_1, x_2)$	$x_1 - y \geqq 0,\ x_2 - y \geqq 0,\ -x_1 - x_2 + y \geqq -1$
$y = \mathrm{NAND}(x_1, x_2)$	$x_1 + y \geqq 1,\ x_2 + y \geqq 1,\ -x_1 - x_2 - y \geqq -2$
$y = \mathrm{OR}(x_1, x_2)$	$-x_1 + y \geqq 0,\ -x_2 + y \geqq 0,\ x_1 + x_2 - y \geqq 0$
$y = \mathrm{NOR}(x_1, x_2)$	$-x_1 - y \geqq -1,\ -x_2 - y \geqq -1,\ x_1 + x_2 + y \geqq 1$
$y = \mathrm{XOR}(x_1, x_2)$	$x_1 - x_2 + y \geqq 0,\ -x_1 + x_2 + y \geqq 0,\ x_1 + x_2 - y \geqq 0,\ -x_1 - x_2 - y \geqq -2$
$y = \mathrm{XNOR}(x_1, x_2)$	$x_1 - x_2 - y \geqq -1,\ -x_1 + x_2 - y \geqq -1,\ x_1 + x_2 + y \geqq 1,\ -x_1 - x_2 + y \geqq -1$
$y = \mathrm{NOT}(x)$	$x + y \geqq 1,\ -x - y \geqq -1$
$y = \mathrm{BUFFER}(x)$	$x - y \geqq 0,\ -x + y \geqq 0$

(1) Extract the fanin cone C^{g1} reachable to f and the fanout cone C^{f} reachable from f, from C.

(2) Copy C as C^{g2}.

(3) Translate C^{g1}, C^{g2} and C^{f} into the corresponding ILP constraints, and create additional constraints to express the connection between C^{g2} and C^{f}.

(4) Create the constraints for detecting f.

(5) Apply an ILP solver to the above constraints.

Here, we consider Figure 2 and the slow-to-rise transition fault on x_{11}. To generate a two-pattern test for the fault, we first perform (1) and (2) of the above procedure. Figure 3 shows the obtained three circuits, and the corresponding ILP constraints are as follows:

$$\mathrm{G}_1^{\mathrm{g1}}\text{: } x_3^{\mathrm{g1}} + x_{11}^{\mathrm{g1}} \geqq 1, x_6^{\mathrm{g1}} + x_{11}^{\mathrm{g1}} \geqq 1, -x_3^{\mathrm{g1}} - x_6^{\mathrm{g1}} - x_{11}^{\mathrm{g1}} \geqq -2$$

$$\mathrm{G}_1^{\mathrm{g2}}\text{: } x_1^{\mathrm{g2}} + x_{10}^{\mathrm{g2}} \geqq 1, x_3^{\mathrm{g2}} + x_{10}^{\mathrm{g2}} \geqq 1, -x_1^{\mathrm{g2}} - x_3^{\mathrm{g2}} - x_{10}^{\mathrm{g2}} \geqq -2$$
$$\mathrm{G}_2^{\mathrm{g2}}\text{: } x_3^{\mathrm{g2}} + x_{11}^{\mathrm{g2}} \geqq 1, x_6^{\mathrm{g2}} + x_{11}^{\mathrm{g2}} \geqq 1, -x_3^{\mathrm{g2}} - x_6^{\mathrm{g2}} - x_{11}^{\mathrm{g2}} \geqq -2$$
$$\mathrm{G}_3^{\mathrm{g2}}\text{: } x_2^{\mathrm{g2}} + x_{16}^{\mathrm{g2}} \geqq 1, x_{11}^{\mathrm{g2}} + x_{16}^{\mathrm{g2}} \geqq 1, -x_2^{\mathrm{g2}} - x_{11}^{\mathrm{g2}} - x_{16}^{\mathrm{g2}} \geqq -2$$
$$\mathrm{G}_4^{\mathrm{g2}}\text{: } x_{11}^{\mathrm{g2}} + x_{19}^{\mathrm{g2}} \geqq 1, x_7^{\mathrm{g2}} + x_{19}^{\mathrm{g2}} \geqq 1, -x_{11}^{\mathrm{g2}} - x_7^{\mathrm{g2}} - x_{19}^{\mathrm{g2}} \geqq -2$$
$$\mathrm{G}_5^{\mathrm{g2}}\text{: } x_{10}^{\mathrm{g2}} + x_{22}^{\mathrm{g2}} \geqq 1, x_{16}^{\mathrm{g2}} + x_{22}^{\mathrm{g2}} \geqq 1, -x_{10}^{\mathrm{g2}} - x_{16}^{\mathrm{g2}} - x_{22}^{\mathrm{g2}} \geqq -2$$
$$\mathrm{G}_6^{\mathrm{g2}}\text{: } x_{16}^{\mathrm{g2}} + x_{23}^{\mathrm{g2}} \geqq 1, x_{19}^{\mathrm{g2}} + x_{23}^{\mathrm{g2}} \geqq 1, -x_{16}^{\mathrm{g2}} - x_{19}^{\mathrm{g2}} - x_{23}^{\mathrm{g2}} \geqq -2$$

$$\mathrm{G}_3^{\mathrm{f}}\text{: } x_2^{\mathrm{f}} + x_{16}^{\mathrm{f}} \geqq 1, x_{11}^{\mathrm{f}} + x_{16}^{\mathrm{f}} \geqq 1, -x_2^{\mathrm{f}} - x_{11}^{\mathrm{f}} - x_{16}^{\mathrm{f}} \geqq -2$$
$$\mathrm{G}_4^{\mathrm{f}}\text{: } x_{11}^{\mathrm{f}} + x_{19}^{\mathrm{f}} \geqq 1, x_7^{\mathrm{f}} + x_{19}^{\mathrm{f}} \geqq 1, -x_{11}^{\mathrm{f}} - x_7^{\mathrm{f}} - x_{19}^{\mathrm{f}} \geqq -2$$
$$\mathrm{G}_5^{\mathrm{f}}\text{: } x_{10}^{\mathrm{f}} + x_{22}^{\mathrm{f}} \geqq 1, x_{16}^{\mathrm{f}} + x_{22}^{\mathrm{f}} \geqq 1, -x_{10}^{\mathrm{f}} - x_{16}^{\mathrm{f}} - x_{22}^{\mathrm{f}} \geqq -2$$
$$\mathrm{G}_6^{\mathrm{f}}\text{: } x_{16}^{\mathrm{f}} + x_{23}^{\mathrm{f}} \geqq 1, x_{19}^{\mathrm{f}} + x_{23}^{\mathrm{f}} \geqq 1, -x_{16}^{\mathrm{f}} - x_{19}^{\mathrm{f}} - x_{23}^{\mathrm{f}} \geqq -2$$

In (c) of Figure 3, since x_2^{f}, x_7^{f} and x_{10}^{f} are floating signals, we must have the following constraints.

$$x_2^{\mathrm{g2}} - x_2^{\mathrm{f}} = 0$$
$$x_7^{\mathrm{g2}} - x_7^{\mathrm{f}} = 0$$
$$x_{10}^{\mathrm{g2}} - x_{10}^{\mathrm{f}} = 0$$

These constraints mean, whenever a vector is applied to the circuit, the values of x_2^{f}, x_7^{f} and x_{10}^{f} must be identical with the values of the corresponding signals in (b) of Figure 3.

Now, we consider the detection conditions for the slow-to-rise transition fault on x_{11}. According to (1) of the detection conditions mentioned before, x_{11} must be set to '0' under the first vector of a two-pattern test. Hence, the following constraint is required.

$$x_{11}^{\mathrm{g1}} = 0$$

Moreover, according to (2) of the detection conditions, in order to detect the corresponding stuck-at fault, we need to differentiate the fault-free circuit from the faulty one. To translate this condition into ILP constraints, we introduce variables e_{22}, e_{23} and the following constraints.

$$x_{11}^{\mathrm{f}} = 0$$
$$x_{11}^{\mathrm{g2}} = 1$$

$$x_{22}^{\mathrm{g2}} - x_{22}^{\mathrm{f}} + e_{22} \geqq 0,\ -x_{22}^{\mathrm{g2}} + x_{22}^{\mathrm{f}} + e_{22} \geqq 0,$$
$$x_{22}^{\mathrm{g2}} + x_{22}^{\mathrm{f}} - e_{22} \geqq 0,\ -x_{22}^{\mathrm{g2}} - x_{22}^{\mathrm{f}} - e_{22} \geqq -2$$
$$x_{23}^{\mathrm{g2}} - x_{23}^{\mathrm{f}} + e_{23} \geqq 0,\ -x_{23}^{\mathrm{g2}} + x_{23}^{\mathrm{f}} + e_{23} \geqq 0,$$
$$x_{23}^{\mathrm{g2}} + x_{23}^{\mathrm{f}} - e_{23} \geqq 0,\ -x_{23}^{\mathrm{g2}} - x_{23}^{\mathrm{f}} - e_{23} \geqq -2$$

$$e_{22} + e_{23} \geqq 1$$

The last constraint represents that the error must be propagated to at least one primary output.

In this way, a two-pattern test can be generated by applying an ILP solver to all the above constraints.

3. Proposed Method

3.1 Our Test Generation Problem

We formally state our test generation problem as follows.

- Input: A combinational circuit C and a transition fault f in C
- Output: A two-pattern test set T_f that propagates the errors caused by f to its all reachable primary outputs

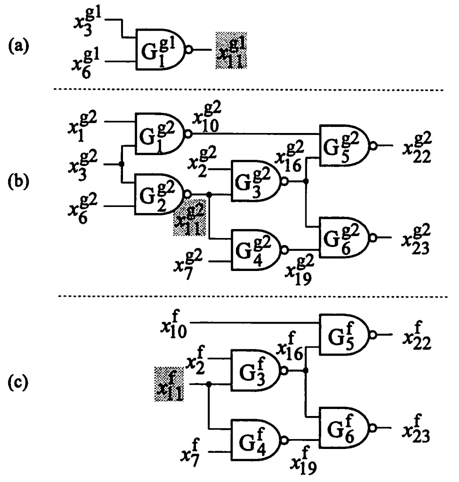

Figure 3: Three circuits for fault detection: (a) Fault-free circuit for generating the first vector of a two-pattern test; (b) Fault-free circuit for generating the second vector of a two-pattern test; and (c) Faulty circuit for generating the second vector of a two-pattern test

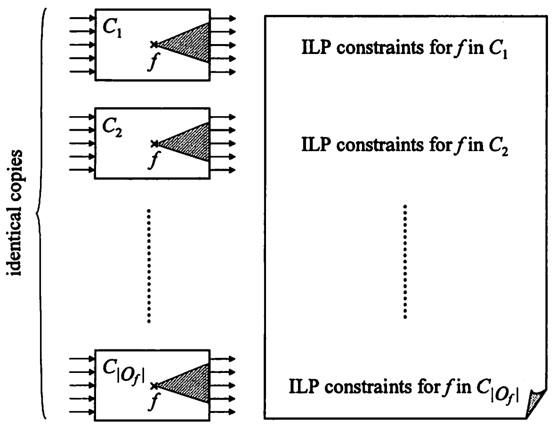
identical copies
C_1
f
C_2
f
$C_{|O_f|}$
f
ILP constraints for f in C_1
ILP constraints for f in C_2
ILP constraints for f in $C_{|O_f|}$

Figure 4: ILP model for generating a minimum test set

- Objective: Minimizing $|T_f|$

To solve this problem, we derive the following formulation.

3.2 ILP Formulation

The upper bound of $|T_f|$ is $|O_f|$, where O_f represents all the primary outputs reachable from f, because one test is enough to propagate the error of f to each reachable primary output. We make use of this upper bound to formulate an ILP problem. Here, we prepare $|O_f|$ copies of the given circuit, and associate ILP constraints to detect f with each copy (Figure 4). This implies that, for f, $|O_f|$ tests can be generated simultaneously. If we identify useless copies of them as much as possible, we will finally obtain a minimum test set for f. To achieve this, we consider additional constraints in the following.

We introduce 0-1 variables $e_{i,j}$, where $1 \leqq i \leqq |O_f|$ and $1 \leqq j \leqq |O_f|$. Variable $e_{i,j}$ takes 1 if the error of f is propagated to the j-th primary output in C_i, otherwise takes 0. In general, there is a redundant primary output at which the error of f is never observed. For such a primary output, we prepare 0-1 variables r_j, where $1 \leqq j \leqq |O_f|$. Equation $r_j = 1$ indicates the error of f does not reach at the j-th primary output of any copy, and $r_j = 0$ indicates the error of f reaches at the j-th primary output of at least one copy. By using these variables, we have the following constraints for all j.

$$\sum_{i=1}^{|O_f|} e_{i,j} + r_j \geqq 1 \qquad (1)$$

This means that, the j-th primary output of every copy is redundant, or the error of f is propagated to the j-th primary output of at least one copy. Since $e_{i,j} = r_j = 1$ never happen for all i, j, we also have the following constraints.

$$e_{i,j} + r_j \leqq 1 \qquad (2)$$

Now, in order to identify useless copies, we introduce 0-1 variables $e'_{i,j}$, where $1 \leqq i \leqq |O_f|$ and $1 \leqq j \leqq |O_f|$. For all i (or $i \geqq 2$), j, the following constraints are defined.

$$e'_{i,j} - e_{i,j} \geqq 0 \qquad (3)$$

$$e'_{i,j} - e'_{i-1,j} \geqq 0 \qquad (4)$$

In inequality (3), $e_{i,j} = 1$ implies $e'_{i,j} = 1$. In inequality (4), whenever $e'_{k,j} = 1$ for some k, all $e'_{i,j}$ take 1 for $i > k$.

We are now ready to identify which copy of the circuit not to be needed. Variables u_i are defined for all $i \geqq 2$, j.

$$-e'_{1,j} + u_1 \geqq 0 \qquad (5)$$

$$e'_{i-1,j} - e'_{i,j} + u_i \geqq 0 \qquad (6)$$

Whenever the error of f reaches at the j-th primary output of C_i for the first time, i.e., the error does not reach at the j-th primary output of C_k $(k < i)$, $u_i = 1$.

Finally, we have to minimize the following equation.

$$\sum_{i=1}^{|O_f|} u_i + |O_f| \cdot \sum_{j=1}^{|O_f|} r_j \qquad (7)$$

The first term counts the number of copies that are used for propagating the errors to all the reachable primary outputs.

Table 2: Values of $e_{i,j}$

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$i=1$	0	1	0	0	0
$i=2$	0	1	0	0	1
$i=3$	0	0	1	0	0
$i=4$	1	0	0	0	1
$i=5$	0	1	1	0	0

Table 3: Values of $e'_{i,j}$

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$i=1$	0	1	0	0	0
$i=2$	0	1	0	0	1
$i=3$	0	1	1	0	1
$i=4$	1	1	1	0	1
$i=5$	1	1	1	0	1

From inequality (1), it can be seen that r_j can be set to 1 freely. In the second term of the above equation, to prevent r_j from being 1 freely, the term is multiplied by $|O_f|$. Therefore, in the final solution, $r_j = 1$ if and only if the error of f never reaches at the j-th primary output of the circuit.

The values of the primary inputs in the circuits whose u_i take 1 form a minimum test set for f.

3. 3 Example

To clarify our ILP formulation, we give an example here. We use a combinational circuit C with five primary outputs as an example circuit. To generate a minimum test set for a fault f in C, five copies $C_1, C_2, \ldots, C_5$ of C need to be prepared. Now, let us consider a situation where ILP constraints for the test generation was provided for an ILP solver, and, during solving the ILP problem, the temporal feasible assignment shown in Tables 2–5 was obtained.

Table 2 represents that the errors of f reach at the 2nd primary output of C_1, at the 2nd and 4th primary outputs of C_2, at the 3rd and 5th primary outputs of C_3, at the 1st and 5th primary outputs of C_4, and at the 2nd and 3rd primary outputs of C_5, respectively. From Table 4, we can see that C_1, C_2, C_3 and C_4 are used for fault detection at the 2nd, 5th, 3rd and 1st primary outputs, respectively. Note that, it is possible for u_5 to take 1. However, in the final solution after solving the ILP problem, such an assignment will be rejected.

3. 4 Sizes of variables and constraints

Here, we estimate the sizes of variables and constraints in our test generation problem. Let n be the number of signal lines in a combinational circuit. It is enough to prepare $3n$ variables for fault detection (Figure 3). As mentioned in the previous subsection, since $|O_f|$ copies of the original circuit are produced, totally $3n \cdot |O_f|$ variables are required for fault detection. The additional variables of r_j, $e'_{i,j}$ and u_i, where $1 \leqq i \leqq |O_f|$ and $1 \leqq j \leqq |O_f|$, are used to derive a minimum test set, totally $|O_f|^2 + 2|O_f|$ variables are also needed. Thus, we need to prepare at most $3n \cdot |O_f| + |O_f|^2 + 2|O_f|$ variables. The number of constraints for fault detection and for test set minimization can roughly be estimated as $O(n \cdot |O_f|)$ and $O(|O_f|^2)$, respectively.

Table 4: Values of u_i

$i=1$	1
$i=2$	1
$i=3$	1
$i=4$	1
$i=5$	0

Table 5: Values of r_j

$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
0	0	0	1	0

4. Conclusions and Future Work

In this paper, we presented an integer programming formulation to generate high quality transition tests for combinational circuits. When a combinational circuit and a transition fault in the circuit are given, our method always generates a minimum test set that propagates the errors of the fault to all the primary outputs reachable from the fault site. In the future, we should evaluate the proposed method for several benchmark circuits.

Acknowledgments

This work was supported in part by the research promoting expenses for assistant professors of JAIST.

References

[1] S. C. Ma, P. Franco and E. J. McCluskey, "An experimental chip to evaluate test techniques: experimental results," *Proc. International Test Conference*, pp. 663–672, 1995.

[2] C.-W. Tseng and E. J. McCluskey, "Multiple-output propagation transition fault test," *Proc. International Test Conference*, pp. 358–366, 2001.

[3] B. Benware, C. Schuermyer, N. Tamarapalli, K.-H. Tsai, S. Ranganathan, R. Madge, J. Rajski and P. Krishnamurthy, "Impact of multiple-detect test patterns on product quality," *Proc. International Test Conference*, pp. 1031–1040, 2003.

[4] H. Tang, G. Chen, S. M. Reddy, C. Wang, J. Rajski and I. Pomeranz, "Devect aware test patterns," *Proc. Design, Automation and Test in Europe*, pp. 450–455, 2005.

[5] T. Hosokawa and K. Yamazaki, "An n-detection test generation method to increase fault sensitization coverage," *IE-*

ICE Transactions on Information Systems (Japanese Edition), Vol. J90-D, No. 6, pp. 1474–1482, June 2007.

[6] B. Vaidya and M. B. Tahoori, "Delay testing based on transition faults propagated to all reachable outputs," *Proc. International Workshop on Defect Based Testing*, pp. 67–75 2004.

[7] I. Park, A. Al-Yamani and E. J. McCluskey, "Effective TARO pattern generation," *Proc. VLSI Test Symposium*, pp. 161–166 2005.

[8] N. K. Jha and S. Gupta, *Testing of digital systems*, Cambridge University Press, 2003.

[9] J. P. M. Silva, "Integer programming models for optimization problems in test generation," *Proc. Asia and South Pacific Design Automation Conference*, pp. 481–487, 1998.

[10] P. F. Flores, H. C. Neto and J. P. M. Silva, "An exact solution to the minimum size test pattern problem," *ACM Transactions on Design Automation of Electronic Systems*, Vol. 6, Issue 4, pp. 629–644, Oct. 2001.