16th IEEE International Test Synthesis Workshop (ITSW 2009), Mar. 2009.

— Extended Abstract —
Unsensitizable Path Identification at RTL Using High-Level Synthesis Information*

Satoshi Ohtake (Speaker) Naotsugu Ikeda Michiko Inoue Hideo Fujiwara
Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan E-mail: {ohtake, kounoe, fujiwara}@is.naist.jp

1. Introduction

Unsensitizable path information available prior to test generation is very valuable for reducing cost and improving accuracy and quality of delay testing of digital circuits. Since there exists no test pattern that activate delay faults on an unsensitizable path along the path, prior identification of them can greatly reduce test generation time. Furthermore, although some originally unsensitizable paths can become sensitizable due to application of design for testability (DFT) and result in over-testing, this can be alleviated by using the information. Several untestable path delay fault (PDF) identification methods at gate level have been proposed. However, it is difficult to get satisfactory results by these identification methods for a large circuit containing a tremendous number of paths. To avoid this difficulty, a methods utilizing register transfer level (RTL) design information have been proposed [1]. The method can identify untestable paths based on non-robust testability of PDFs in a reasonable amount of time by handling a number of gate level paths as an RTL path. However, for a circuit in RTL description, all the possible combinations of state transitions of the controller are needed to be examined and paths go through status signal lines cannot be handled in the method.

For a circuit designed at behavioral level, an RTL design of the circuit is obtained by high-level synthesis (HLS). By analyzing behavioral level description of a circuit, unusable/unnecessary data transfers and/or control flows can be extracted (e.g., the method in [2]). In this work, we focus on such knowledge obtained from HLS and propose a method to identify unsensitizable paths by utilizing the knowledge in addition to RTL design information for further improvement from the previous method. In this work, (1) all the RTL paths in a circuit including paths go through status signal lines, which are missed in [1], are dealt with, (2) transitions on control signal lines are considered when paths go through the control signal lines are targeted, and (3) time required for identification is improved from the previous methods. We show the effectiveness of use of information obtained from HLS through our experiments using RTL circuits synthesized from behavioral benchmark circuits.

*This work was supported in part by Semiconductor Technology Academic Research Center (STARC) under the Research Project.

2. Preliminary

RTL circuit model Our target RTL design is structural RTL circuit model. A structural RTL description of a circuit consists of a controller and a data path represented by RTL modules and RTL signal lines, where an RTL module is a combinational operation module or a multiplexer (MUX) or a register and an RTL signal line has bit width and connects between two RTL modules. They are connected with each other by control signal lines and status signal lines. Notice that, since we assume that circuits are designed at behavior level and synthesized by a high-level synthesis tool, RTL structural information is affordable.

RTL paths An RTL path is an ordered set of RTL modules $\{m_0, m_1, \ldots, m_n\}$, where m_0, m_n and $m_i (0 < i < n)$ are a primary input or a register, a primary output or a register, and a combinational module, respectively, and the output of $m_i (0 \leq i < n)$ is an input of m_{i+1}. Unsensitiability of RTL paths is defined as follows.

Definition 1: *(RTL unsensitizable path)* Given a sensitization criterion, an RTL path p in an RTL circuit is *RTL unsensitizable under the sensitization criterion* if any gate-level path corresponding to p in its gate-level circuit is gate-level unsensitizable under the criterion for any logic synthesis. □

In this paper, the *non-robust sensitization criterion* [3], which is most widely used in ATPG systems, is used and a path is referred to as an unsensitizable path if both of the PDFs on the path are non-robust untestable.

HLS information In HLS systems, a circuit is generally represented by a *control data flow graph (CDFG)* [4] at behavioral level. During HLS, each operation is assigned to a control step (*scheduling*) and operations and variables are mapped to operational modules and registers, respectively (*binding*). In this paper, we assume that, for a circuit, a *scheduled CDFG (S-CDFG)* and *binding information* are given in addition to RTL structural information. These are referred to as *high-level synthesis (HLS) information*. An example of HLS information of a circuit is shown in Figure 1 (a) and (b). In figure 1 (a), the triangular nodes and edges between them denote conditional branches or merges and control flows, respectively. The circular nodes and edges between them in rectangles denote operations and variables, respec-

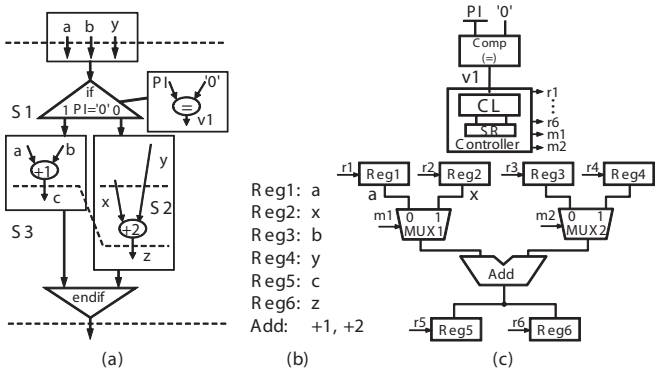

Figure 1. An S-CDFG (a), binding information (b) and a synthesized RTL circuit (c).

tively. The horizontal dotted lines denote boundaries between control steps. In Figure 1 (b), the left of a colon denotes a operational module or a register and the right denotes operations or variables bound to.

In this work, we assume HLS and logic synthesis to be the following. (1) Every register allocated during HLS has a load enable control and is controlled so that it does not load if its input has no valid data. (2) For any combinational RTL module, no logic optimization beyond its boundary is done during logic synthesis.

3. Proposed Method

For a circuit at RTL, existence of an RTL path does not always mean that the path can be used to propagate meaningful data. For example, in Figure 1 (a) and (b), RTL module "Add" is shared by two operations "+1" and "+2". Figure 1 (c) shows an example of its synthesized RTL circuit, There are eight RTL paths which pass through "Add" but only four paths of them are actually used path. Paths that start at "Reg2" or "Reg4" and end at "Reg5" never propagate meaningful data because any operation corresponding to such paths is not specified at behavior level.

In order to deal with RTL structural information and HLS information for a circuit efficiently, we construct *module input output dependence graphs (IODGs)*. An IODG represents relation between inputs (resp. an output) of an RTL module and variables on the inputs (resp. output) of operations bound to the RTL module and dependence of input and output variables. For an RTL path, our untestable path identification method employs IODGs for modules on the path to examine whether some meaningful data flow along the path can be made or not.

IODG The IODG of an RTL module M is a directed graph $G = (V,E)$, where V is the set of nodes representing input and output variables of operations corresponding to M, and $E \subset V \times V$ is the set of edges representing input/output (IO) dependence between a couple of input and output variables. IO dependence between two variables is defined as a relation that (1) these two variables are an input variable and an output variable of an operation and (2) the edge corresponding to the input variable does not cross boundary of control steps more than once in the interested S-CDFG. Notice that, (2) is a condition for sensitizing the path to test delay fault along the path.

RTL paths can start and/or end at a state register and also can contain control signals and/or status signals. For an RTL module M_C with a control input, in addition to the IODG of M_C, a partial IODG is created for each control value of the control input. However the partial IODG is used for identifying paths, which go through the control input, with transitions on the control input, the discussion about this is omitted due to the limitation of the space.

Since operations corresponding to MUXs in a given RTL schematic are not described explicitly in the corresponding S-CDFG, we cannot extract IO dependences for variables corresponding to MUXs. Furthermore, IO dependences for variables corresponding to the control logic (CL) in the controller cannot be derived because operations corresponding to the CL are not appear in the S-CDFG.

To deal with MUXs and CL, *thru* operations corresponding to MUXs and *dummy* operations to explicitly represents the IO variables of the CL are added to the S-CDFG and dummy modules corresponding to the dummy operations are added to the RTL schematic. Figure 2 shows the modified HLS information and RTL schematic of Figure 1. In this example, thru operations $t1,t2,t3,t4$ are added to the S-CDFG. Variables $v2,v3,v6,v7$ are created by adding dummy operations $d1,d2,d3,d4$. According to addition of the dummy operations, dummy modules $D1,D2,D3,D4$ are added to the RTL schematic.

From the modified HLS information and RTL schematic, we can create IODGs for the RTL modules including the MUXs and CL. Example of IODGs for modules in the RTL circuit of Figure 1(c) is showin in Figure 3. In the IODG for the CL, a special variable S is used to represent the state register.

Unsensitizable path identification Given an RTL circuit and HLS information. For an RTL path $P = \{M_0, M_1, \ldots, M_{n-1}, M_n\}$, there exists a data flow corresponding to P if the value of M_0 changes when a state transition occurs, the effect is propagated along P and the effect is captured at M_n when the next state transition occurs. Thus there exist variables $v_0, v_1, \ldots, v_{n-1}$ and operations $o_1, o_2, \ldots, o_{n-1}$ which satisfy the following property. (1) v_0 and v_{n-1} are assined to M_0 and M_n, respectively. (2) The edge corresponding to v_0 does not cross boundaries of control steps more than once on the S-CDFG. (3) For each $i(1 \leq i \leq n-1)$, o_i is assigned to M_i and v_{i-1} and v_i are the input variable and output variable of o_i, respectively. From the definition

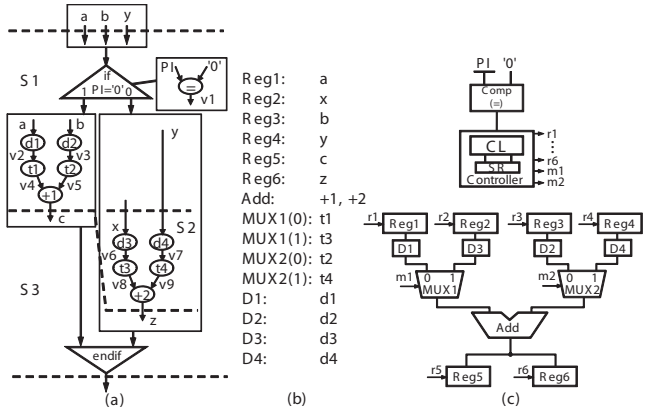

Figure 2. An augmented S-CDFG (a), augmented binding information (b) and its corresponding RTL circuit (c).

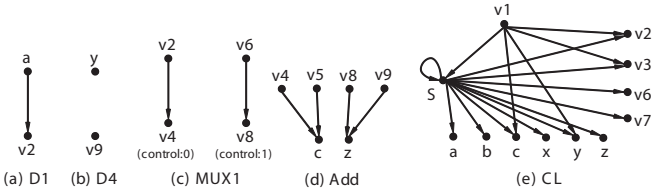

Figure 3. IODGs for modules in the circuit of Figure 2 (c).

of IODG, for each $M_i (1 \leq i \leq n-1)$, there exists an edge (v_{i-1}, v_i) of the IODG of M_i. This means that there exists a path $v_0, v_1, \ldots, v_{n-1}$ referred to as an *IODG path* on the union of IODGs. In the proposed method of unsensitizable path identification, an IODG path corresponding to a given RTL path is sought and the RTL path is judged to be unsensitizable if there does not exist any IODG path. In this work, we also prove the correctness of the conditions for unsensitizability of path

4. Experimental Results

In this experiments, we evaluate the effectiveness of the method. The identification results for RTL circuits synthesized from behavioral level benchmark circuits by the HLS method of [5] are shown in Table 1. In the table and tables discussed below, columns "#Total" show the total number of paths in the circuits, columns "#US" denote the number of unsensitizable paths and columns "#US w/ trans." or "#USwt" denote the number of unsensitizable paths with transition on control signals. The time required for the identification is less than 1 second for every circuit. Since GCD is the only circuit which has status signals, we examined in detail. Table 2 and 3 shows the break down for GCD at RTL and the classification of corresponding gate-level paths, respectively. Although the previous method cannot find any of unsensitizable paths, the proposed method can identify 800 unsensitizable gate-level paths (1/3 of the total number of paths).

Table 1. The number of RTL unsensitizable paths identified by our method.

Circuit	Through no control signal		Through control signals		
	#US	#Total	#US	#US w/ trans.	#Total
JWF	124	155	78	144	516
LWF	8	21	0	12	46
Tseng	6	22	0	0	40
Paulin	13	31	2	12	78
GCD	0	10	4	9	70

Table 2. Classification of RTL paths in GCD wrt going through status signals.

Through no control signal				Through control signals					
No status		Th. status		No status			Th. status		
#US	#Total	#US	#Total	#US	#USwt	#Total	#US	#USwt	#Total
0	12	0	8	0	0	22	4	9	44

5. Conclusions

This paper proposed a method for identifying unsensitizable paths based on non-robust sensitization criterion in sequential circuits synthesized through high-level synthesis (HLS) and logic-synthesis. The method uses RTL structural information and HLS information of the circuits. Since a bunch of paths, called an RTL path, is dealt with at RTL, large number of unsensitizable gate-level paths can be efficiently identified. By using HLS information, the method can efficiently handle paths pass through status signals which cannot be dealt with in the previous work. Our experimental results proved that the proposed method can improve the quantity of identified unsensitizable paths compared to the previous method.

References

[1] Y. Yoshikawa, S. Ohtake and H. Fujiwara: "False path identification using rtl information and its application to over-testing reduction for delay faults," in *Proc. of Asian Test Symposium*, pp. 65–68, 2007.

[2] R. Bergamaschi: "The effects of false paths in high-level synthesis," in *Proc. of International Conf. on Computer-Aided Design*, pp. 80–83, 1991.

[3] A. Krstić and K.-T. T. Cheng: *Delay fault testing for VLSI circuits*, Kluwer Academic Publishers, 1998.

[4] D. D. Gajski: *High-Level Synthesis: Introduction to Chip and System Design*, Kluwer Academic Publishers, 1992.

[5] T. Takasaki, T. Inoue and H. Fujiwara: "A high-level synthesis approach to partial scan design," in *Proc. of Asian Test Symp.*, pp. 309–314, 1999.

Table 3. Classification of gate-level paths in GCD.

#Structural paths in the circuit				# US	
Through no control	Through control: Register load	Through control: MUX select	Total	Without transition	With transition
280	744	1400	2424	0	800