

An Approach to RTL-GL Path Mapping Based on Functional Equivalence

Hiroshi Iwata , Satoshi Ohtake and Hideo Fujiwara

Graduate School of Information Science, Nara Institute of Science and Technology

8916-5 Takayama, Ikoma, Nara 630-0192, Japan

{ hiroshi-i, ohtake, fujiwara }@is.naist.jp

Abstract

Information on false paths in a circuit is useful for design and test. The use of this information may contribute not only in reducing the time required for logic synthesis, circuit area, test generation time and test application time of the circuit, but also in alleviating over-testing. Since identification of false paths at gate level (GL) is hard for large circuits with a tremendous number of paths, several methods using high-level design information have been proposed. These methods are effective only if the correspondence between paths at register transfer level (RTL) and paths at gate level can be established. Until now, the correspondence has been established only by some restricted logic synthesis. In this paper, we propose a method of mapping an RTL path to its corresponding gate level paths without any specific logic synthesis. The method first maps each bit-sliced RTL signal line of an RTL path to a gate level net by considering the functional equivalence of those signal lines. The RTL path is then mapped to gate level paths using these correspondences.

Key words: path mapping, register transfer level, gate-level, functional equivalence, fault diagnosis

1 Introduction

For design and test of circuits, false path information is very valuable since it can be used for reducing logic synthesis time, circuit area, test generation time, and test application time, while also minimizing over-testing. From the perspective of design, since design constraints on false paths can be ignored, designers can replace gates on the false paths by smaller gates with longer delay. Therefore optimizing paths longer than the critical path can be skipped if they are identified as false paths since they don't have to meet design constraints. From the testing point of view, since no test pattern can be generated for path delay faults on false paths, prior false path identification can greatly reduce ATPG time. Furthermore, since some path delay faults on false paths can become testable due to application of design for testability (DFT) and result in over-testing, this can be alleviated by false path identification.

Several false path identification methods at gate level for combinational circuits[1, 3, 7] and for sequential circuits[4, 8] have been proposed. However, since it is difficult to apply false path identification methods at gate level for large circuits containing a tremendous number of paths, some methods using register transfer level (RTL) design information, instead of gate level, have been proposed[5, 9]. While not specifically targeting false paths, Nourani et al.[5] proposed a method using timing analysis and RTL design information to determine the actual critical path and avoid false paths longer than the true critical path. Yoshikawa et al.[9] defined RTL false path and proposed a method to identify them. Furthermore, Ikeda et al.[2] proposed an RTL false path identification method using high-level synthesis information. However, these methods would be useful only if the correspondence between paths at RTL and paths at gate level can be established. Until now, the correspondence has been established through module interface preserving-logic synthesis (MIP-LS)[9]. Currently, using MIP-LS is the only way to guarantee to have the correspondence. However, it is not practical to restrict synthesis only to MIP-LS. In this paper, we propose a method of mapping an RTL path to its corresponding gate level paths without assuming MIP-LS.

The proposed method first maps each bit-sliced RTL signal line of an RTL path to a gate level net by considering the functional equivalence of those signal lines (this is called signal line mapping). Then, the RTL path is mapped to gate level paths using the functional equivalence relation of the signal lines (this is called path mapping). A method which finds gate level nets suspected to correspond to a given RTL signal line is proposed in [6]. In this paper, we propose a method to identify functionally equivalent gate level nets of the RTL signal line among the nets obtained by the method in [6]. Our RTL path mapping involves mapping each RTL signal line of the path to gate level nets until we obtain a set of gate level paths which passes through these nets.

The rest of this paper is organized as follows. The proposed signal line mapping methodology, which identifies the gate level functionally equivalent nets of RTL signal lines, is presented in Section 2. Section 3 presents the proposed path mapping methodology, which uses correspon-

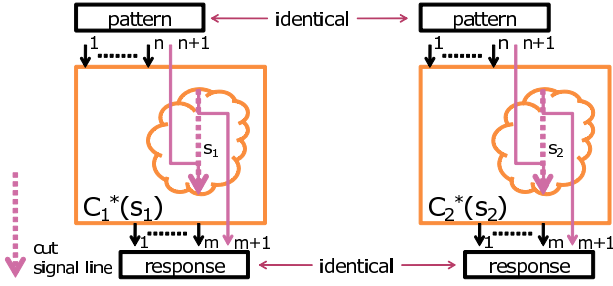


Figure 1. Functionally equivalent signal lines s_1 and s_2 .

dence between gate level nets and RTL signal lines established by the previous signal line mapping methodology. Section 4 discusses the implementation of the signal line mapping proposed in Section 2. Section 5 shows case studies. Section 6 concludes the paper.

2 RTL-GL signal line mapping problem

In this section, we define functional equivalence relation between two signal lines and our target circuits, and also formulate the problem searching functionally equivalent nets. Furthermore, we will also show an algorithm for solving the problem.

2.1 Functional equivalence

Here, we first define *signal line cutting*, which is an operation needed for defining functionally equivalent signal lines.

Definition 1: (*Signal line cutting*) For a combinational circuit C with n inputs, m outputs and an internal signal line s , the following operation is referred to as *cutting C on s* .

1. Create the $(n+1)$ -th new input port and the $(m+1)$ -th new output port.
2. Remove the signal line s .
3. Create connections between $(n+1)$ -th input port and the end point of s and between the start point of s and the $(m+1)$ -th output port.

In the following discussion, we represent the combinational circuit resulting from the above operations as $C^*(s)$.

For two functionally equivalent combinational circuits, we define a functional equivalence of signal lines as follows.

Definition 2: (*Functionally equivalent signal line*) For two functionally equivalent combinational circuits C_1 and C_2 with internal signal lines s_1 and s_2 , respectively, s_1 and s_2 are *functionally equivalent* if $C_1^*(s_1)$ and $C_2^*(s_2)$ are functionally equivalent. \square

Figure 1 illustrates functionally equivalent signal lines. The signal lines s_1 and s_2 are functionally equivalent if the responses from $C_1^*(s_1)$ and $C_2^*(s_2)$ are identical for any input pattern. If there is only a limited number of input patterns, then limited functional equivalence can be considered. Let X be a set of patterns with n bits. Let X^* be the

set of patterns with $n+1$ bits formed by expanding each $x \in X$ into $x&0$ and $x&1$ where ' $a&b$ ' denotes concatenation of vectors a and b . The pattern translation from set X to X^* is *bit expansion of X* .

Definition 3: (*Functionally equivalent signal line w.r.t. X*) For two n -input combinational circuits C_1 and C_2 with internal signal lines s_1 and s_2 , respectively, we generate $(n+1)$ -input combinational circuits $C_1^*(s_1)$, $C_2^*(s_2)$ by cutting s_1 , s_2 , respectively. Given a set of n bit input patterns, X , and its expanded set X^* , s_1 and s_2 are *functionally equivalent w.r.t. X* if any pattern of X^* is applied to $C_1^*(s_1)$ and $C_2^*(s_2)$, their output responses are identical. \square

Lemma 1 Let C_1 , C_2 and s_1 be functionally equivalent circuits and a signal line in C_1 , respectively. If there exists at least one signal line that is functionally equivalent to s_1 in C_2 and if there exists only one signal line s_2 that is functionally equivalent to s_1 w.r.t. a set of input patterns, then s_1 and s_2 are functionally equivalent. \square

[Proof] The set of signal lines in C_2 that are functionally equivalent to s_1 in C_1 w.r.t. a set of input patterns, X , properly includes all the signals in C_2 that are functionally equivalent to s_1 . Therefore, if there exists functionally equivalent signal lines to s_1 in C_2 and s_2 is only the signal line that is functionally equivalent to s_1 w.r.t. X , then s_1 and s_2 are functionally equivalent. \square

2.2 Circuit model

In this paper, we only consider structural RTL designs. A structural RTL design consists of a controller represented by combinational modules and state registers, and a datapath represented by RTL modules and signal lines connecting them where an RTL module is a MUX or an operational module or a register. An RTL signal line consists of one-bit signal lines as follows.

Definition 4: (*Bit-sliced RTL signal line*) For an RTL signal line s , each one bit signal line separated from s is referred to as a *bit-sliced RTL signal line of s* . The i -th bit of s is represented as $s[i]$. \square

For solving signal line mapping problem, it is sufficient to consider only the RTL combinational circuit C^R which is the combinational part of a given structural RTL design S^R and the gate level combinational circuit C^G which is the combinational part of a gate level design synthesized from S^R . We assume that there is a one-to-one correspondence between the input/output signal lines of C^R and C^G . This relation is called *I/O mapping information*. The I/O mapping information of C^R and C^G can be obtained by preserving all the bits of the registers in S^R during logic synthesis.

2.3 RTL-GL signal line mapping problem

In this subsection, we formulate the *RTL-GL signal line mapping problem* to find a set of nets, which is functionally

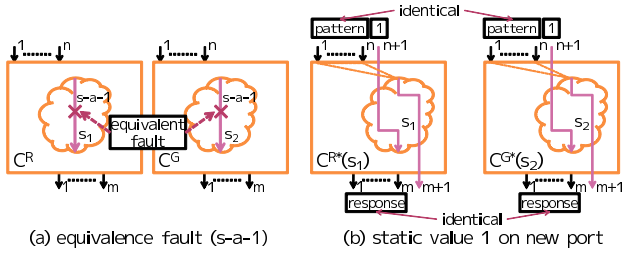


Figure 2. Relation between equivalent faults and functionality of respective signal lines.

equivalent to a bit-sliced RTL signal line in an RTL combinational circuit, in a gate level combinational circuit.

Definition 5: (RTL-GL signal line mapping problem)

Input

- C^R : an RTL combinational circuit design
- C^G : a gate level circuit that is functionally equivalent to C^R
- The I/O mapping information between C^R and C^G
- $e^R[k]$: the k -th bit-sliced RTL signal line of an RTL signal line e^R in C^R

Output $E^G = \{e^G | e^G \equiv_l e^R[k]\}$ where e^G is a net in C^G and \equiv_l is a symbol representing relation of functional equivalence of signal lines \square

2.4 Proposed method

Given an RTL combinational circuit C^R and a gate level combinational circuit C^G , checking functional equivalence between a bit-sliced RTL signal line e^R in C^R and a gate level net e^G in C^G can be performed by applying all the possible input patterns to both circuits $C^{R*}(e^R)$ and $C^{G*}(e^G)$, and comparing their output responses. For each bit-sliced RTL signal line, by examining every gate level net, we can identify functionally equivalent gate level nets. However, it is not practical to explicitly check all the possible input patterns and the functional equivalence for all the combinations of nets in gate level circuit and bit-sliced RTL signal lines. Ravi et al.[6] proposed a method finding candidates for functionally equivalent nets of a given bit-sliced RTL signal line using fault diagnosis techniques using test patterns (i.e., limited input patterns). In this paper, their method is utilized to solve the RTL-GL signal line mapping problem. More specifically, their method assumes that the bit-sliced RTL signal line has a stuck-at fault and finds the stuck-at faults, which have the identical behavior of the fault under the test patterns, in the gate level circuit. The faults in the gate level circuit and the fault in the RTL circuit are said to be *equivalent*. It is a necessary condition for functional equivalence that the responses of the RTL circuit and the gate level circuit are identical when a value v is fixed to e^R and e^G (see Fig.2 (b)), when s-a- v is assumed to be presented on e^G and on e^R (see Fig.2 (a)).

Moreover, we propose an algorithm to find a set of functionally equivalent gate level nets w.r.t. a set of limited input

patterns of the bit-sliced RTL signal line. Note that the set of the gate level nets obtained by the proposed algorithm may include not functionally equivalent nets. The solution of RTL-GL signal line mapping problem can be used for solving path mapping problem only if the solution does not include not functionally equivalent nets. We show that the algorithm can identify the functionally equivalent net of the bit-sliced RTL signal line under some assumption in Theorem 1. We also evaluate the effectiveness through case studies in Section 5. The algorithm to solve the RTL-GL signal line mapping problem is as follows.

1. Generate a complete test set T for all the testable stuck-at faults in C^G .
2. For each $v \in \{0, 1\}$, the following two steps are performed.
 - (a) Obtain a set of faulty output responses R_{fv} by applying T to the RTL circuit C^R with an injected s-a- v fault on the given bit-sliced RTL signal line $e^R[k]$.
 - (b) Find all the single s-a- v faults of C^G such that all the faulty circuits induced by the faults respond the same output responses R_{fv} when T is applied to these circuits. Here we refer the faults the equivalent faults. A set of the nets is referred to as E^{Gv} such that the nets have equivalent faults.
3. Obtain $E^G = E^{G0} \cap E^{G1}$.
4. Extract the input cone circuit $C^R(e^R[k])$ of the additional primary output appeared by cutting $e^R[k]$ where the input cone circuit of a signal line s in a circuit C is referred to as $C(s)$.
5. For each $e^G \in E^G$, make an input cone circuit $C^G(e^G)$ of the additional primary output added by cutting e^G .
6. For each $C^G(e^G)$, remove e^G from E^G if the set of the output responses of $C^R(e^R[k])$ and $C^G(e^G)$ are not identical when T is applied.

Steps 1-3 are the same as the procedure for searching functionally equivalent signal line by using fault diagnosis technique in [6]. In [6], the complete test set T for the detectable faults in a gate level circuit is used as the fault diagnosis input patterns. The procedure first finds the s-a-0 (resp. 1) faults in C^G that are equivalent to the s-a-0 (resp. 1) fault injected on $e^R[k]$. Then the procedure selects gate level nets that have both s-a-0 and s-a-1 faults as the equivalent faults. Moreover, in this paper, for the set of gate level nets obtained by steps 1-3, we add steps 4-6 to exclude the gate level nets that have different values from the bit-sliced RTL signal line for same pattern of T . Lemma 2 shows that, for a bit-sliced RTL signal line $e^R[k]$, the algorithm outputs the functionally equivalent nets of $e^R[k]$ w.r.t. a set of input patterns T as a solution to the problem.

Lemma 2 Given an RTL combinational circuit C^R , its synthesized gate level circuit C^G and a bit-sliced RTL signal

line $e^R[k]$ in C^R . Any $e^G \in E^G$ are functionally equivalent to $e^R[k]$ w.r.t. a set of input pattern T such that E^G is the set of gate level nets in C^G obtained by the RTL-GL signal line mapping algorithm. \square

[Proof] The RTL-GL signal line mapping algorithm mainly consists of two steps:

- (i) Select gate level nets that have both s-a-0 and s-a-1 faults as the equivalent faults of s-a-0 and s-a-1 on $e^R[k]$ under T , respectively.
- (ii) For E^G obtained by (i), $e^G \in E^G$ is removed if the input cone circuits of e^G in C^G and $e^R[k]$ in C^R , respectively, are not functionally equivalent.

We show that (i) and (ii) satisfy the condition of Definition 3 where primary outputs of C^G and C^R have the same output response when e^G and $e^R[k]$ have the same value for any input pattern of T .

First, we show that (i) guarantees that the primary outputs of C^G and C^R have the same response for any input pattern in T when e^G and $e^R[k]$ have the same value. Let C^R has n inputs ($x^R[i](i = 1, \dots, n)$) and m outputs ($z^R[i](i = 1, \dots, m)$). $C^{R*}(e^R[k])$ has $n + 1$ inputs ($x^{R*}[i](i = 1, \dots, n + 1)$) and $m + 1$ outputs ($z^{R*}[i](i = 1, \dots, m + 1)$). We inject a s-a- v fault on $e^R[k]$ in C^R where $v \in \{0, 1\}$. For any $t \in T$, the output response from $z^R[1], \dots, z^R[m]$ of C^R with the s-a- v obtained by applying t to C^R with the fault and that from $z^{R*}[1], \dots, z^{R*}[m]$ of $C^{R*}(e^R[k])$ obtained by applying $t \& v$ to C^{R*} .

Let C^G has n inputs ($x^G[i](i = 1, \dots, n)$) and m outputs ($z^G[i](i = 1, \dots, m)$). $C^{G*}(e^G)$ has $n + 1$ inputs ($x^{G*}[i](i = 1, \dots, n + 1)$) and $m + 1$ outputs ($z^{G*}[i](i = 1, \dots, m + 1)$). We inject a s-a- v fault on signal line e^G in C^G . For any $t \in T$, the output response from $z^G[1], \dots, z^G[m]$ of C^G with the s-a- v obtained by applying t to C^G with the fault and that from $z^{G*}[1], \dots, z^{G*}[m]$ of $C^{G*}(e^G[k])$ obtained by applying $t \& v$ to C^{G*} .

Consequently, for any input pattern of T , the output responses from $z^{R*}[1], \dots, z^{R*}[m]$ and $z^{G*}[1], \dots, z^{G*}[m]$ of $C^{R*}(e^R[k])$ and $C^{G*}(e^G)$, respectively, are the same because C^R and C^G are functionally equivalent and s-a- v faults on $e^R[k]$ and e^G are equivalent under T .

Next we show that (ii) guarantees that $e^G \in E^G$ and $e^R[k]$ have the same value. For the procedure 4-6, the RTL-GL signal line mapping algorithm excludes all the signal lines having different output responses by applying some input pattern of T . Therefore $e^G \in E^G$ and $e^R[k]$ have the same value for any input pattern of T .

From the above discussion, every element of E^G and $e^R[k]$ are functionally equivalent w.r.t. T . \square

Theorem 1 Given an RTL combinational circuit C^R , its synthesized gate level circuit C^G and a bit-sliced RTL signal line $e^R[k]$ in C^R . For the RTL-GL signal line mapping algorithm, $e^G \in E^G$ is the functionally equivalent net in C^G of $e^R[k]$ if there exists at least one functionally equiv-

alent net of $e^R[k]$ and the solution of the algorithm is with $|E^G| = 1$. \square

[Proof] From Lemma 2, every $e^G \in E^G$ obtained by RTL-GL signal line mapping algorithm is functionally equivalent to $e^R[k]$ w.r.t. T . Since $|E^G| = 1$, from Lemma 1, the element of E^G is functionally equivalent to $e^R[k]$. \square

3 RTL-GL path mapping problem

In this section, we define a gate level path, an RTL path and functional equivalence between them. Furthermore, we formulate the path mapping problem and present a solution of the problem.

3.1 Gate level and RTL path models

Definition 6: (*Gate level path*) An ordered set of gate level nets $\{e_1^G, \dots, e_n^G\}$ is called a *gate level path* if the following conditions are satisfied.

- (1) e_1^G is the net adjacent to a primary input or the output of an FF
- (2) e_n^G is the net adjacent to a primary output or the input of an FF
- (3) $e_i^G (i = 2, \dots, n - 1)$ is the net connecting between the gates having e_{i-1}^G as an input and e_{i+1}^G as an output, respectively \square

Definition 7: (*Sub gate level path*) A subset of a gate level path p^G is called a *sub gate level path* of p^G . \square

Definition 8: (*RTL path*) An ordered set of RTL signal lines $\{e_1^R, \dots, e_n^R\}$ is called an *RTL path* if the following conditions are satisfied.

- (1) e_1^R is the line adjacent to a primary input or the output of a register
- (2) e_n^R is the line adjacent to a primary output or the input of a register
- (3) $e_i^R (i = 2, \dots, n - 1)$ is the signal line connecting between the modules having e_{i-1}^R as an input and e_{i+1}^R as an output, respectively \square

Definition 9: (*Sub RTL path*) A subset of an RTL path p^R is called a *sub RTL path* of p^R . \square

Definition 10: (*Bit-sliced RTL path*) An ordered set of bit-sliced RTL signal lines $\{e_1^R[k_1], \dots, e_n^R[k_n]\}$ is called a *bit-sliced RTL path* if the following conditions are satisfied.

- (1) $e_1^R[k_1]$ is the k_1 -th bit-sliced RTL signal line adjacent to a primary input or the output of a register
- (2) $e_n^R[k_n]$ is the k_n -th bit-sliced RTL signal line adjacent to a primary output or the input of a register
- (3) $e_i^R[k_i] (i = 2, \dots, n - 1)$ is the k_i -th bit-sliced RTL signal line connecting between the modules having $e_{i-1}^R[k_{i-1}]$ as an input and $e_{i+1}^R[k_{i+1}]$ as an output, respectively \square

Definition 11: (*Sub bit-sliced RTL path*) A subset of a bit-sliced RTL path p^R is called a *sub bit-sliced RTL path* of p^R . \square

3.2 Relation between paths

We define the functional equivalence between sub bit-sliced RTL path and sub gate level path as follows.

Definition 12: (*Functionally equivalent path*) Sub bit-sliced RTL paths and sub gate level paths are simply referred to as *sub paths*. Sub paths $q_1 = \{e_{1_1}, \dots, e_{1_n}\}$ and $q_2 = \{e_{2_1}, \dots, e_{2_m}\}$ are *functionally equivalent* if q_1 and q_2 satisfy the following conditions.

- (1) $n = m$
- (2) $e_{1_i} \equiv_l e_{2_i} \ (i = 1, \dots, n)$ \square

Definition 13: (*Identification of path*) A sub RTL path q^R is said to *uniquely identify* an RTL path p^R if p^R is the only path that properly includes q^R . \square

3.3 RTL-GL path mapping problem

We formulate the RTL-GL path mapping problem as a problem to search a set of sub-gate level paths corresponding to an RTL path.

Definition 14: (*RTL-GL path mapping problem*)

Input

- C^R : an RTL combinational circuit
- C^G : a gate level circuit that is functionally equivalent to C^R
- The I/O mapping information between C^R and C^G
- p^R : an RTL path

Output $P^G = \bigcup_{i=0}^n \bigcup_{j=0}^{m_i} P_{ij}^G$

Where P_{ij}^G is defined as follows. Let $q_i^R (i = 1, \dots, n)$ be a sub RTL path that uniquely identifies p^R and $q_{ij}^R (j = 1, \dots, m_i)$ be a sub bit-sliced RTL path of q_i^R where n is the number of the sub RTL paths of p^R . Let q_{ij}^G be a sub gate level path that is functionally equivalent to q_{ij}^R . P_{ij}^G is a set of gate level paths including q_{ij}^G . \square

3.4 RTL-GL path mapping algorithm

We propose an algorithm solving the RTL-GL path mapping problem as follows. The algorithm establishes correspondences between an RTL path p^R and a set of gate level paths, P^G .

- (1) Generate the minimum sub RTL path $q_i^R (i = 1, \dots, n)$ that uniquely identifies p^R .
- (2) Obtain a gate level net e_{ijk}^G that is functionally equivalent to each bit-sliced RTL signal line $q_{ijk}^R (k = 1, \dots, l)$, where q_{ijk}^R is an element of a bit-sliced RTL path $q_{ij}^R (j = 1, \dots, m_i)$ of q_i^R , m_i is the number of combinations of bit-sliced RTL paths obtained by specifying bit portion of every RTL signal line on q_i^R , and l is the number of RTL signal lines on q_i^R .
- (3) For each pair of i and j , find all the gate level paths that include $\{e_{ij_1}^G, \dots, e_{ij_l}^G\}$. The set of the obtained gate level paths is referred to as P_{ij}^G .
- (4) Calculate $P^G = \bigcup_{i=0}^n \bigcup_{j=0}^{m_i} P_{ij}^G$.

Table 1. Circuit characteristics.

Circuit	#bit	#PI	#PO	#reg	#path ^R	#path ^b
GCD	4	2	1	3	8	692
LWF	4	2	2	5	19	7,600
Tseng	4	2	2	6	20	2,216

4 Implementation of signal line mapping

In this section, we discuss an implementation of the RTL-GL signal line mapping. Using this implementation, proposed method can be used to map an identified RTL false path by the method in [9] into corresponding gate level paths automatically.

4.1 Searching equivalent fault

In steps 1-3 of the algorithm, we search functionally equivalent nets by using equivalent fault relation. The test pattern set created in step 1 which detects all the detectable faults in C^G can be generated by some ATPG tool. The output responses R_{f0} (resp. R_{f1}) of step 2a can be calculated by some RTL simulation tool for C^R with the s-a-0 (resp. s-a-1) fault. For step 2b, we can realize automation by some fault diagnosis tool.

4.2 Checking the values of signal lines

In steps 4-6 of the algorithm, we evaluate that the bit-sliced RTL signal $e^R[k]$ and the gate level net e^G , which is obtained by the former steps, have the same value for any input pattern of T . This is realized with evaluating if output responses from input cone circuits of $e^R[k]$ and e^G in C^R and C^G , respectively, are the same for every pattern of T .

5 Case study

In this section, we show experimental results for evaluating our RTL-GL path mapping method. We used three RTL benchmark circuits, GCD, LWF and Tseng. In this experiments, we used only the datapath part of each circuit and tried to map all the paths in the datapath. Table 1 shows the circuit characteristics of the circuits. Columns #bit, #PI, #PO, #reg, #path^R and #path^b show the bit width, the number of primary inputs, that of primary outputs, that of registers, that of RTL paths and that of bit-sliced RTL paths.

We use the RTL path mapping ratio (Pmr) given by $Pmr = \frac{|P^{RT}|}{|P^R|} \times 100[\%]$ as an evaluation criterion. Where $|P^R|$ is the total number of RTL paths in the datapath and $|P^{RT}|$ is the number of RTL paths mapped. Furthermore, to evaluate more in detail, we consider bit-sliced RTL paths in the datapath. We use the bit-sliced RTL path mapping ratio (Pmr_b) given by $Pmr_b = \frac{|P_b^{RT}|}{|P_b^R|} \times 100[\%]$. Where $|P_b^R|$ is the total number of bit-sliced RTL paths in the datapath and $|P_b^{RT}|$ is the number of bit-sliced RTL paths mapped.

Table 2 shows the above path mapping ratio obtained by our proposed method for the three datapaths of the benchmark circuits. The proposed method achieves the RTL path mapping ratio and the bit-sliced RTL path mapping ratio of average 85.9% and 79.1%, respectively. The reason why

Table 2. Path mapping.

Circuit	GCD	LWF	Tseng	Avg
Pmr	88.9%	73.7%	95.0%	85.9%
Pmr_b	72.3%	65.7%	99.3%	79.1%

(bit-sliced) RTL paths which cannot be mapped to gate level paths exist is that the algorithm cannot find any signal line needed for path mapping, i.e., there exist no functionally equivalent net in the gate level circuits.

For the RTL-GL signal line mapping, the proposed method guarantees to get a correct solution provided that there exists at least one gate-level net that is functionally equivalent to a targeted bit-sliced RTL signal line. Therefore, we need to evaluate the impact of the assumption, i.e., the ratio of signal mapping errors induced by the assumption to the total number of signal lines to be mapped should be evaluated. The error ratio (Err) is given by $Err = \frac{|e_b^{Err}|}{|e_b^R|} \times 100[\%]$. Where $|e_b^R|$ is the total number of bit-sliced RTL signal lines in the datapath and $|e_b^{Err}|$ is the number of bit-sliced RTL signal lines which are incorrectly mapped to gate level nets. In this paper, we calculate Err for the datapath of LWF. If we use the method of Ravi et al. [6] for the purpose of the signal line mapping, the method more likely to make errors than the proposed method. Table 3 shows the mapping error ratio of the proposed method and that of the method of [6].

Compared to the method of [6], the proposed method guarantees to find nets which are functionally equivalent to a bit-sliced RTL signal line w.r.t. T by employing additional procedure 4-6. However, time required for the procedure 1-3 dominates the overall execution time. Therefore the proposed method can improve accuracy of the signal line mapping without a large impact on its execution time. Furthermore, even if our method chooses a net which is not functionally equivalent to the bit-sliced RTL signal line (irrespective of T), it may not directly induce incorrect path mapping. Since, for a bit-sliced RTL path, there are several signal lines must be mapped and the mapped gate level nets must be structurally connected, it is conceivable that the mapping error may be absorbed during path mapping.

6 Conclusions

Establishing correspondence between an RTL circuit and its synthesized gate level circuit is important for high level testing approaches. In this work, we focus on correspondence between a path in the RTL circuit and paths in the gate level circuit. Existence of the correspondence enables the technologies that handle a path at RTL as a bundle of a tremendous number of paths in the gate level circuit. There is a method to identify false paths using RTL design information [9] which is feasible only if RTL paths can be mapped into gate level paths. The method can quickly identify false paths using RTL information under the assumption

Table 3. Error ratio.

[6]	5.0%
ours	0%

that correspondence between RTL paths and gate level paths is available. Until now, it is guaranteed only by a restricted logic synthesis.

In this paper, we propose a method to establish the correspondence between RTL paths and gate level paths without using any restricted logic synthesis. In our case study, for several benchmark circuits, average 85.9% of the RTL paths was able to be mapped into gate level paths. The signal line mapping, unfortunately, may incorrectly select nets which are not functionally equivalent because of restricting search space. In the study, however, only functionally equivalent net was selected.

In our future work, we improve the accuracy of the signal line mapping. Furthermore, to improve the quantity of mapped paths, signal line mapping from an RTL signal line into multiple gate level nets is dealt with.

Acknowledgments

The authors would like to thank Profs. Michiko Inoue and Tomokazu Yoneda of Nara Institute of Science and Technology for valuable discussion and their cooperation. This work was supported in part by Semiconductor Technology Academic Research Center (STARC) under the Research Project and in part by Japan Society for the Promotion of Science (JSPS) under Grants-in-Aid for Scientific Research (B) (No. 20300018).

References

- [1] K. T. Cheng and H. C. Chen. Classification and identification of nonrobust untestable path delay faults. *IEEE Trans. on CAD*, 15(8):845–853, Aug. 1996.
- [2] N. Ikeda, S. Ohtake, M. Inoue, and H. Fujiwara. RTL false path identification using high level synthesis information. Technical Report 482, Feb. 2008.
- [3] S. Kajihara, K. Kinoshita, I. Pomeranz, and S. Reddy. A method for identifying robust dependent and functionally unsensitizable paths. In *International Conf. on VLSI Design*, pages 82–87, 1997.
- [4] A. Krstić, S. T. Chakradhar, and K.-T. T. Cheng. Testable path delay fault cover for sequential circuits. In *European Design and Test Conf.*, pages 220–226, 1996.
- [5] M. Nourani and C. A. Papachristou. False path exclusion in delay analysis of RTL structures. *IEEE Trans. on Very Large Scale Integration Systems*, 10(1):30–43, Feb. 2002.
- [6] S. Ravi, I. Ghosh, V. Boppana, and N. K. Jha. Fault-diagnosis-based technique for establishing RTL and gate-level correspondences. *IEEE Trans. on CAD*, 20(12):1414–1425, 2001.
- [7] S. M. Reddy, S. Kajihara, and I. Pomeranz. An efficient method to identify untestable path delay faults. In *Asian test symposium*, pages 233–238, 2001.
- [8] R. Tekumalla and P. Menon. Identifying redundant path delay faults in sequential circuits. In *International Conf. VLSI Design*, pages 406–411, 1996.
- [9] Y. Yoshikawa, S. Ohtake, and H. Fujiwara. False path identification using RTL information and its application to over-testing reduction for delay faults. In *Asian Test Symposium*, pages 65–68, 2007.