PAPER *Special Issue on Dependable Computing*

# A Test Plan Grouping Method to Shorten Test Length for RTL Data Paths under a Test Controller Area Constraint

**Toshinori HOSOKAWA**[†], **Hiroshi DATE**[††], **Masahide MIYAZAKI**[†††], **Michiaki MURAOKA**[†††], ***Regular Members, and*** **Hideo FUJIWARA**[††††], ***Fellow***

**SUMMARY** This paper proposes a test generation method using several partly compacted test plan tables for RTL data paths. Combinational modules in data paths are tested using several partly compacted test plan tables. Each partly compacted test plan table is generated from each grouped test plan set and is used to test combinational modules corresponding to the grouped test plans. The values of control signals in a partly compacted test plan table are supplied by a test controller. This paper also proposes the architecture of a test controller which can be synthesized in a reasonable amount of time, and proposes a test plan grouping method to shorten test length for data paths under a test controller area constraint. Experimental results for benchmarks show that the test lengths are shortened by 4 to 36% with −9 to 8% additional test controller area compared with the test generation method using test plans.

***key words:*** *test plan grouping, test controllers, partly compacted test plan tables, RTL data paths, hierarchical test generation*

## 1. Introduction

A design for testability (DFT) method [1], [2] is important for the design of reliable VLSI circuits. The objectives of a DFT method are the following: (1) high fault efficiency, (2) short test application time [3], and (3) at-speed-testing [4] under area and power consumption constraints. Recently, non-scan DFT methods [5], [6] for RTL (Register Transfer Level) design circuits were proposed to attain the above-mentioned objectives. RTL design circuits consist of a data path part and a controller part. The former is represented by hardware elements (e.g. registers, multiplexers, and operational modules) and signals, and the latter is represented by a finite state machine (FSM). A controller and a data path are connected with internal signals: control signals and status signals. A control signal comes from a controller, and a status signal comes from a data path. DFT methods [5]–[10] for data paths are based on a hierarchical test generation approach [11] and are classified into two major approaches. One is a DFT approach based on the normal function of a controller [7], [8], [10], and the other is a DFT approach without the normal function of a controller [5], [6], [9].

In the former approach, test plans [9] for combinational modules (multiplexers and operational modules) in a data path are generated using the normal function of a controller. If test plans cannot be generated using the normal function of a controller, DFT elements are added into a data path to generate test plans. The values of the original control signals in a test plan are supplied by the original controller and the values of control signals added for DFT in a test plan are supplied by the test registers [7]. In this DFT approach, the area overhead for DFT is small, but the information of normal data flow is required to generate test plans. The length of each test plan depends on the normal function of a controller. Thus, the test application time also depends on the normal function.

In the latter approach, test plans are generated to minimize their lengths using only the structures of a data path, and DFT elements are added into a data path to generate test plans with minimum lengths. The values of the control signals in a test plan are supplied by a test controller [5], [6]. In this DFT approach, the test application time is short and the information of normal data flow is not required to generate test plans. However, the area overhead for DFT is large because a test controller is required to supply the test plans. In this paper, it is considered that test application time is the most important problem and the discussion focuses on the latter DFT approach. In [5], a data path is strongly testable [5], [9] and a test controller is a sequential circuit. In [6], a data path is fixed-control testable and a test controller is a combinational circuit. The area for test controllers was improved compared with that of [5], but the area overhead for data paths was increased because fixed-control testability is covered by strong testability. In this paper, it is considered that the area overhead for data paths is more critical than area for test controllers. Thus, the architecture of a test controller proposed in [5] is discussed.

Recently, a test generation method using a compacted test plan table [12] was proposed. All test plans in a data path are compacted into a compacted test plan table such that the length is minimized. In other

Manuscript received April 4, 2003.
Manuscript revised July 7, 2003.

[†] The author is with the College of Industrial Technology, Nihon University, Narashino-shi, 275–8575 Japan.

[††] The author is with System JD Co., Ltd., Fukuoka-shi, 814–0001 Japan.

[†††] The authors are with the Design Technology Development Department, Semiconductor Technology Academic Research Center (STARC), Yokohama-shi, 222–0033 Japan.

[††††] The author is with the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Ikoma-shi, 630–0101 Japan.

words, the method tests as many combinational modules as possible at the same time in order to reduce the test application time. In [12], it was assumed that control signals of a data path are controllable.

This paper will first discuss the test lengths by the test generation using test plans, and by the test generation using a compacted test plan table (CTPT) in Sect. 2. Their test controllers will also be discussed. Then, their problems are revealed. In Sect. 3, in order to solve these problems, a test generation method using several partly compacted test plan tables and the architecture of a test controller are proposed. The optimization problem for a test plan grouping is formulated using the integer linear programming (ILP) to shorten the test length under a test controller area constraint. In Sect. 4, experimental results are shown. Finally, Sect. 5 concludes this paper.

## 2. Test Controller for Data Path Testing and its Test Length

### 2.1 Data Path with Strong Testability

The DFT method [9] for data paths is based on hierarchical test generation [11] and strong testability [9].

Hierarchical test generation is an efficient technique for generating test patterns of very large data paths. In hierarchical test generation, there are two steps. In the first step, it extracts a combinational module M from the data path and generates test patterns of $M$ at gate level using a combinational test generation tool. In the second step, it generates a test plan [9] at RTL. A test plan is defined as follows. Thus, the test patterns and the responses for $M$ are propagated using original data path flows of the data path.

**Definition 1:** Test plan [9]
A test plan for a combinational module $M$ is the test sequence at primary inputs and control signals that propagates a test pattern to $M$ from primary inputs and propagates the response to primary outputs. The values in a test plan are 0's, 1's, $b$'s ($b \in \{0,1\}$) and/or $X'$s ($X$ means don't care). $b$ is the value of a primary input or a control signal which constitutes a test pattern to detect a fault in $M$.

**Example 1:** Table 1 shows the four test plans $T_1$, $T_2$, $T_3$ and $T_4$ for the combinational modules 1, 2, 3, and 4 in a data path, respectively. $P_1$ shows the primary input of a data path and $c_1$, $c_2$, $c_3$, and $c_4$ show the control signals of a data path.

**Definition 2:** Strong testability [9]
A data path is strongly testable if there exists a test plan for each combinational module $M$. A data path which is strongly testable is referred to as a data path with strong testability.

The DFT method [9] is applied to data paths to be strongly testable.

**Table 1** Test plans.

$T_1$

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | b | 0 | X | X | X |
| 1 | b | X | X | b | X |
| 2 | X | X | X | 0 | X |

$T_2$

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | b | X | X | X | X |
| 1 | X | X | b | X | 0 |

$T_3$

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | b | b | 0 | 1 | X |
| 1 | X | 0 | 1 | X | X |
| 2 | X | X | X | X | X |

$T_4$

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | b | 0 | 0 | 1 | b |
| 1 | X | X | X | X | X |

### 2.2 Compacted Test Plan Table

A compacted test plan table [12] generation is summarized as follows.

**Definition 3:** Test plan scheduling [12]
Let $T_i$ be a test plan for a combinational module $i$ ($i = 1, 2, \ldots, n$) in a data path with strong testability and primary inputs including control signals ($P_0, P_1, \ldots, P_{w-1}$). The value of $P_k$ at time $t$ in $T_i$ is denoted by $T_i(t,k)$. Let $L_i$ be the length of $T_i$. Let $w$ be the number of primary inputs. Let $n$ be the number of combinational modules. Let TPST be a test plan scheduling table. The row of a TPST represents time and the total number of rows is $\sum_{i=1}^{n} L_i$. The column of a TPST represents a test plan for a combinational module and the total number of columns is $n$. Each column is sub-divided into sub-columns. The sub-column represents a primary input and the total number of sub-columns in each column is $w$. The value of $P_k$ at time $t$ and a column for $T_i$ in a TPST is denoted by $TPST(t,i,k)$. It is called that $T_i$ is scheduled at time $t_1$ in a $TPST$ if $TPST(t,i,k) = T_i(t - t_1, k)$ for any $t$, $k$, such that $t_1 \le t < t_1 + L_i$, and $0 \le k < w$. When $T_i$ is scheduled at time $t_1$ in a TPST, the period of time from $t_1$ to $t_1 + L_i - 1$ in a TPST is defined as the life time of $T_i$. $TPST(t,i,k) = X$ for any $t$, $k$, such that $0 \le t < t_1$, $t_1 + L_i \le t < \sum_{i=1}^{n} L_i$ and $0 \le k < w$. The degree of a life time at time $t$ is defined as the number of test plans whose life time contains time $t$.

$$\bigcap_{i=1}^{n} TPST(t,i,k) \quad \text{for any } t, k, \text{ such that}$$
$$0 \le t < \sum_{i=1}^{n} L_i \quad \text{and } 0 \le k < w \qquad (1)$$

Operation (1) means that a compaction operation $\bigcap_f$ [12] shown in Table 2 is applied to the scheduling result of a TPST.

If the result of the operation (1) does not include $\varphi$, it is called that the scheduling result is compatible.

**Table 2** Operation $\cap_f$.

| $\cap_f$ | b | X | 0 | 1 |
|---|---|---|---|---|
| b | φ | b | φ | φ |
| X | b | X | 0 | 1 |
| 0 | φ | 0 | 0 | φ |
| 1 | φ | 1 | φ | 1 |

**Table 3** Test plan scheduling table.

| Test plan | $T_1$ | | | | | $T_2$ | | | | | $T_3$ | | | | | $T_4$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
| 0 | | | | | | | | | | | | | | | | b | 0 | 0 | 1 | b |
| 1 | | | | | | | | | | | b | b | 0 | 1 | X | X | X | X | X | X |
| 2 | | | | | | b | X | X | X | X | X | 0 | 1 | X | X | | | | | |
| 3 | b | 0 | X | X | X | X | X | b | X | 0 | X | X | X | X | X | | | | | |
| 4 | b | X | X | b | X | | | | | | | | | | | | | | | |
| 5 | X | X | X | 0 | X | | | | | | | | | | | | | | | |
| … | | | | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | | | | |

**Table 4** CTPT.

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | b | 0 | 0 | 1 | b |
| 1 | b | b | 0 | 1 | X |
| 2 | b | 0 | 1 | X | X |
| 3 | b | 0 | b | X | 0 |
| 4 | b | X | X | b | X |
| 5 | X | X | X | 0 | X |

If the result of operation (1) includes $\varphi$, it is called that the scheduling result is incompatible. If the scheduling result is compatible, the result of operation (1) is called a compacted test plan table (CTPT). If the degree of life time at time $t$ in a TPST is 0, the row at time $t$ is deleted from a CTPT.

A CTPT is generated such that its length is minimized. The algorithm for CTPT generation was proposed in [12].

**Example 2:** Table 3 shows the example of the test plan scheduling table. In this table, $T_1$ is scheduled at time 3, $T_2$ is scheduled at time 2, $T_3$ is scheduled at time 1, and $T_4$ is scheduled at time 0. The CTPT shown in Table 4 is generated from the scheduling result shown in Table 3.

### 2.3 Supply of Test Plans by a Test Controller

The architecture of a test controller proposed in [5] is summarized as follows. Figure 1 shows the test controller which supplies test plans to control signals of the data path with strong testability. The test controller consists of a test plan generator (TPG), a test pattern register (TPR), and a target module register (TMR) as shown in Fig. 1. Consider the test of a combinational module $M$, which has data inputs and control inputs, in the data path. The TMR is used to store the index of $M$. The bit width of the TMR is $\lceil \log_2 n \rceil$, where $n$ is the number of combinational modules in the data path. The TPG generates the test plan of $M$ from the index stored in the TMR. Thus, the TPG supplies the values in $n$ test plans to control signals. The number of states in the TPG is $\max(L_i)$, where $L_i$ is the length of a test plan for a combinational module $i$. When the data input value of a test pattern of $M$ is justified, if some primary inputs of the data path are not used, the control input value is applied from such primary inputs by way of the TPG. Otherwise, the control input value is pre-stored in the TPR and is applied to the control inputs by way of the TPG. If the Reset is applied, the TPR and the TMR load values from some primary inputs of the data path, otherwise, they hold their values. The mode switching signal t1 is used to disable DFT elements of the data path in normal operation mode. In [5], the detailed architecture of the TPG was not described. If a data path has many test plans and many control signals, the TPG cannot be synthesized in a reasonable time.

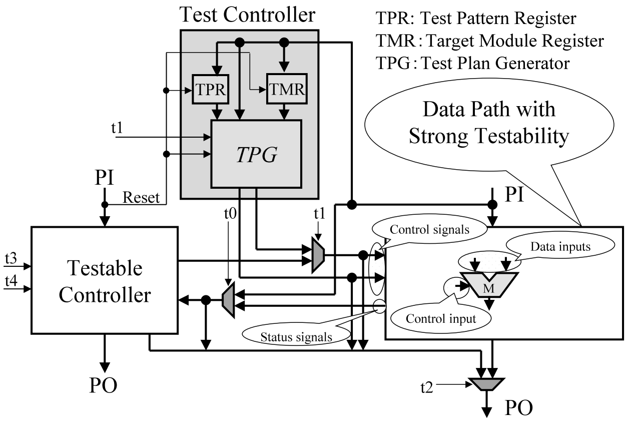

**Fig. 1** Architecture of a test controller.

In [5], testing is sequentially performed for a single combinational module in data paths. The test length for data paths with strong testability using this test controller is, then, given by

$$L = \sum_{i=1}^{n} \left( (L_i + 1) \times N_i \right), \tag{2}$$

where $L$ is the test length for data paths, $n$ is the number of combinational modules, $L_i$ is the length of a test plan for a combinational module $i$ $(i = 1, 2, \ldots, n)$, and $N_i$ is the number of test patterns for a combinational module $i$. $L_i + 1$ cycles are required to apply one test pattern to a combinational module because one cycle is required to load values into the TPR and the TMR. Equation (2) shows that the test length for data paths with strong testability becomes drastically longer as the number of combinational modules and the number of gates in a combinational module increase.

**Example 3:** Consider the test controller which supplies four test plans shown in Table 1 to control signals. The bit width of the TMR is 2 because the number of test plans is 4, the bit width of the TPR is 1 because the number of $b$'s of control signals in a test plan is 1, and the number of states in the TPG is 3 because the maximum length of test plans is 3. Assuming that the numbers of test patterns for the combinational module 1, 2, 3, and 4 are 8, 3, 7, and 2, respectively, the test length for the data path is, according to equation (2), $(3+1)\times 8+(2+1)\times 3+(3+1)\times 7+(2+1)\times 2=75$.

### 2.4 Supply of a CTPT by a Test Controller

All test plans are compacted using the algorithm proposed in [12], and a resultant CTPT is generated. Combinational modules are tested using a CTPT.

Consider a test controller for supplying a CTPT based on the test controller shown in Fig. 1. In the test generation described in subsection 2.3, because a combinational module is tested by a test plan, the TMR is necessary to distinguish n test plans. In the test generation using a CTPT, there is only one test plan table because all the test plans of the n combinational modules are compacted into one table. Because the n combinational modules are tested by using the same CTPT, there is no need to distinguish combinational modules. Thus, the TMR is unnecessary. The TPG generates the values of control signals in a CTPT. The number of states in the TPG is $L_{CTPT}$, where $L_{CTPT}$ is the length of a CTPT. While testing combinational modules, if the primary inputs which drive the TPR are not used, the control input values can be reloaded into the TPR from them using the test controller shown in Fig. 2. In Fig. 2, if the control input values are reloaded into the TPR, the value of the reload signal is 1, otherwise is 0. If a test controller does not have the reload function, the bit width of the TPR is the number of $b$'s of control signals in a CTPT. Thus, the reload function is necessary to reduce the bit width of the TPR.

The test length for data paths with strong testability using this test controller is, then, given by

$$L=\max\left(N_i\right)\times\left(L_{CTPT}+1\right), \tag{3}$$

where $L$ is test length for data paths and $N_i$ is the number of test patterns for a combinational module $i$.

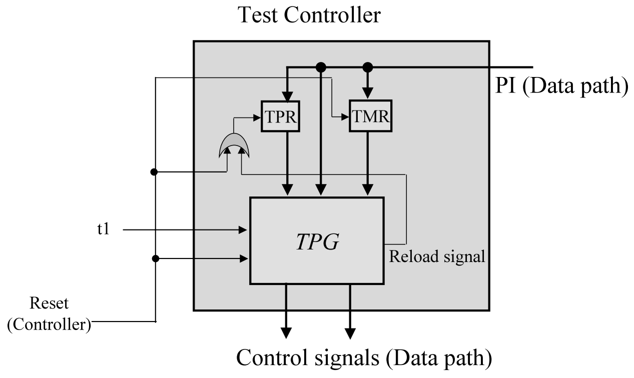

**Fig. 2** Test controller with reload function.

The TPG cannot be synthesized in a reasonable time when $L_{CTPT}$ is large and the number of control signals is large. Thus, it is considered that the test controller to generate the values of control signals in a CTPT is not practical.

It is also predicted that the test length by a test generation using a CTPT is long for data paths with the following characteristics.

Characteristic 1.
The maximum number of test patterns for a combinational module and the minimum number of test patterns for the other combinational modules are very different.

Characteristic 2.
The number of combinational modules with the maximum number of test patterns is small and the number of combinational modules with the minimum number of test patterns is large.

**Example 4:** In the CTPT shown in Table 4, the bit width of the TPR is 4. The values of control signals can be loaded into the TPR at time −1 which is earlier than time 0 in the CTPT by one cycle. The number of states in the TPG is 6 because the length of the CTPT is 6. The test length for the data path is, according to equation (3), $8\times(6+1)=56$.

## 3. Test Generation Method Using Several Partly Compacted Test Plan Tables

In this section, a test generation method using several partly compacted test plan tables is proposed to shorten test length compared with the conventional methods described in Sect. 2.

### 3.1 Preliminaries

**Definition 4:** Partly compacted test plan table
A subset of a test plan set is compacted and the resultant one is referred to as a partly compacted test plan [12]. Especially, when a partly compacted test plan is used for test generation, it is referred to as a partly compacted test plan table (PCTPT).

**Example 5:** Table 5 shows the two PCTPTs

**Table 5** Two PCTPT.

$\mathrm{PCTPT}_1$

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | $b_3$ | $b_3$ | 0 | 1 | X |
| 1 | $b_1$ | 0 | 1 | X | X |
| 2 | $b_1$ | X | X | $b_1$ | X |
| 3 | X | X | X | 0 | X |

$\mathrm{PCTPT}_2$

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | $b_4$ | 0 | 0 | 1 | $b_4$ |
| 1 | $b_2$ | X | X | X | X |
| 2 | X | X | $b_2$ | X | 0 |

**Table 6** Drive control signal table.

$DC_1$

| $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|
| 1 | 0 | 1 | 0 |

$DC_2$

| $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|
| 0 | 1 | 0 | 1 |

$DC_3$

| $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |

$DC_4$

| $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |

(PCTPT$_1$ and PCTPT$_2$). Four test plans shown in Table 1 are partitioned into two groups ($G_1$ and $G_2$). $T_1$ and $T_3$ belong to $G_1$, and $T_2$ and $T_4$ belong to $G_2$. PCTPT$_1$ is generated from $G_1$, and PCTPT$_2$ is generated from $G_2$ by applying the algorithm shown in [12]. In Table 5, $b_i$'s correspond to a test pattern for each combinational module $i$ ($i = 1, 2, \ldots, n$). $b_i$'s are replaced with the test pattern (0's and/or 1's).

**Definition 5:** Density degree

The density degree $DD_{T_i}$ for a test plan $T_i$ shows the number of 0's, 1's, and $b$'s in $T_i$ and is given by the following equation.

$$DD_{T_i} = \sum_{k=1}^{u} \{(c0_k + c1_k + cb_k) \times \delta_{ki}\},$$

where $u$ is the number of control signals, $c0_k$ is the number of 0's of the control signal $c_k$ in $T_i$, $c1_k$ is the number of 1's of the control signal $c_k$ in $T_i$, $cb_k$ is the number of $b$'s of the control signal $c_k$ in $T_i$, and $\delta_{ki}$ is the 0-1 variable. If one of the following conditions is at least satisfied, $\delta_{ki}$ is 0. Otherwise, $\delta_{ki}$ is 1.

(C1) $c0_k$ and $cb_k$ are 0.

(C2) $c1_k$ and $cb_k$ are 0.

(C3) $c0_k$ and $c1_k$ are 0, and $cb_k$ is 1.

**Example 6:** $DD_{T_1}$, $DD_{T_2}$, $DD_{T_3}$, and $DD_{T_4}$ are the density degrees for the test plans $T_1$, $T_2$, $T_3$, and $T_4$ shown in Table 1, respectively. $DD_{T_1}$, $DD_{T_2}$, $DD_{T_3}$, and $DD_{T_4}$ are 2, 0, 4, and 0, respectively.

**Definition 6:** Drive control signal table

The drive control signal table $DC_i$ for a combinational module $i$ shows the control signals where a test plan $T_i$ is supplied. The column of a $DC_i$ represents a control signal $c_k$ ($k = 1, 2, \ldots, u$), where $u$ is the number of control signals of a data path. The row of a $DC_i$ represents flags to show whether $T_i$ is supplied to control signals or not. The value of a flag for $c_k$ in $DC_i$ is denoted by $DC_i(c_k)$. Thus, when there exists 0, 1, or $b$ at any time for a control signal $c_k$ in $T_i$, $DC_i(c_k)$ is 1. Otherwise, $DC_i(c_k)$ is 0.

**Example 7:** Table 6 shows the drive control signal tables $DC_1$, $DC_2$, $DC_3$, and $DC_4$ for the test plans $T_1$, $T_2$, $T_3$, and $T_4$ shown in Table 1, respectively.

## 3.2 PCTPT Generation and Architecture of TPG

A test generation method using several PCTPTs is proposed to shorten test length. A test plan set is partitioned into m groups $G_j$ ($j = 1, 2, \ldots, m$, and $m$ is the number of groups). A PCTPT for each group is generated by compacting test plans in each group.

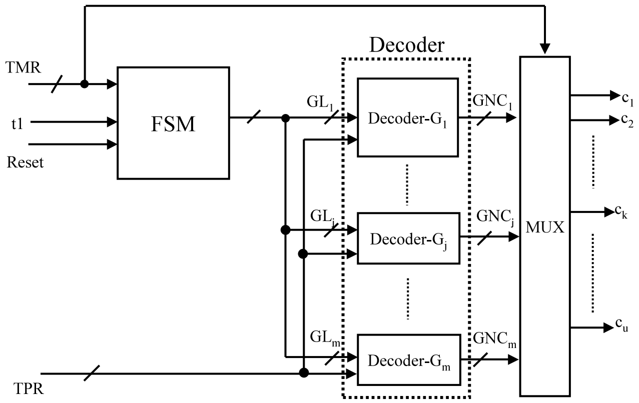

**Fig. 3** Architecture of TPG.

Consider a data path with Characteristic 1 and Characteristic 2. In the test generation using a CTPT, the test length is long for such a data path because the length of unnecessary test sequence is long. In the test generation using several PCTPTs, if the test plans with the maximum number of test patterns and the test plans with minimum number of test patterns do not belong to the same group, the test length can be drastically improved compared with that by the test generation using a CTPT. The test plan grouping method is proposed to shorten the test length in subsection 3.3.

Figure 3 shows the proposed architecture of the TPG. The TPG consists of the FSM, the Decoder, and the MUX. As shown in Fig. 3, the Decoder is divided into m decoders for each PCTPT to synthesize the TPG in a reasonable time. Thus, TPG can be synthesized in a reasonable time by setting the appropriate values of the constraints as described in subsection 3.3.

In Fig. 3, Let $GL_j$ be the length of PCTPT$_j$, Decoder-$G_j$ be a decoder for PCTPT$_j$ and $GNC_j$ be the number of control signals where the values in PCTPT$_j$ are supplied. The MUX is an array of multiplexers. The Decoder consists of m decoders Decoder-$G_j$. The maximum value of the length of the PCTPT$_j$ is the number of states in the FSM, and affects the area of the FSM. The density degree of PCTPT$_j$ affects the area of the Dcoder-$G_j$. $\sum_{j=1}^{m} GNC_j$ affects the area of the MUX.

## 3.3 Test Plan Grouping Method

The test length for data paths can be shortened under a test controller area constraint by considering a test plan grouping. In this subsection, the optimization problem for a test plan grouping is formulated using ILP as follows.

(1) **Input**

(a) $n$ test plans $T_i$ and the number of test patterns $N_i$ $(1 \leq i \leq n, n$ is the number of test plans)
(b) The number of groups: $m$ $(1 \leq m \leq n)$
(c) Constraint $q$
$q$ means $\max\limits_{j}(GNC_j)$. $\left(\max\limits_{i}\left(\sum\limits_{k=1}^{u} DC_i(c_k)\right) \leq q \leq u,\right.$
$u$ is the number of control signals in a data path$\left.\right)$
$GNC_j$ is given by the following equation.

$$GNC_j = \sum_{k=1}^{u} \bigvee_{i=1}^{n} (X_{ij} \times DC_i(c_k))$$

The following 0-1 variable $X_{ij}$ is defined as an ILP variable. $X_{ij} = 1$ ($T_i$ belongs in $G_j$), $X_{ij} = 0$ (Otherwise)
(d) Constraint $p$
$p$ means $\max\limits_{j}\left(\sum\limits_{i=1}^{n}(X_{ij} \times L_i)\right)$. $\left(\max\limits_{j}(L_i) \leq p \leq \sum\limits_{i=1}^{n} L_i\right)$
(e) Constraint $r$
All test plans in $G_j$ are concatenated, the resultant one is referred to as a concatenated test plan of $G_j$, and it is denoted by $CT_j$. All test plans in a data path are concatenated, the resultant one is referred to as a concatenated test plan of a data path, and it is denoted by $CT_{all}$.
$r$ means $\max\limits_{j}(DD_{CT_j})$. $\left(\max\limits_{i}(DD_{T_i}) \leq r \leq DD_{CT_{all}}\right)$

(2) **Output**
$m$ test plan sets $G_j$ $(1 \leq j \leq m)$
(3) **Optimization:** minimize the following cost function $F$

$$F = \sum_{j=1}^{m}\sum_{i=1}^{n}((MAXTP_j - N_i) \times L_i \times X_{ij})$$

$$MAXTP_j = \max_i (X_{ij} \times N_i)$$

**Constraints**

(c1) $\max\limits_{j}(GNC_j) \leq q$

(c2) $\max\limits_{j}\left(\sum\limits_{i=1}^{n}(X_{ij} \times L_i) \leq p\right)$

(c3) $\max\limits_{j}\left(DD_{CT_j}\right) \leq r$

(c4) $\sum\limits_{j=1}^{m} X_{ij} = 1$

(c5) $\sum\limits_{i=1}^{n} X_{ij} \geq 1$

The cost function $F$ is the total sum of the length of unnecessary test sequence for each combinational module and it is expected that the test length is reduced by minimizing $F$. (c1) means that the maximum output number of Decoder-$G_j$ in the TPG is less than or equal to $q$. The area of the MUX in the TPG is reduced by adjusting $q$. (c2) means that the maximum value of the total sum of each test plan length in $G_j$ is less than or equal to $p$. The area of the FSM in the TPG is reduced by adjusting $p$. (c3) means that the maximum value of the density degree of a concatenated test plan $CT_j$ is less than or equal to $r$. The area of the Decoder in the TPG is reduced by adjusting $r$. (c4) means that a test plan $T_i$ belongs to only one group. (c5) means that $G_j$ is not empty.

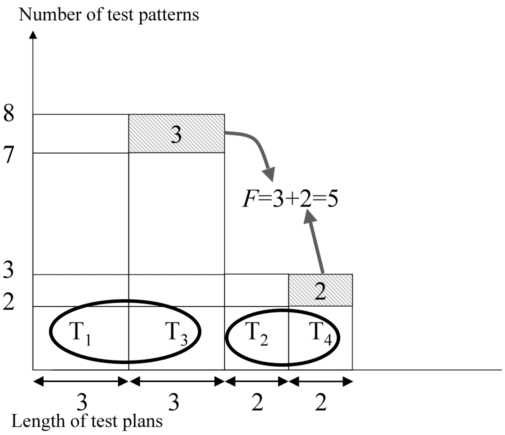

**Fig. 4** Example of test plan grouping $(m = 2)$.

After grouping, test plans in each group are compacted, and a PCTPT for each group is generates from those results.

**Example 8:** Figure 4 shows the examples of test plan grouping $(m = 2)$. The vertical axis of the graph shows the number of test patterns for combinational modules used in Example 3. The rectangles in the graph show the test plans, and the width of the rectangles shows the length of the test plans shown in Table 1. The circles in the graph show the groups of the test plans. In the graph, the area of parts with shadows shows the total sum of the lengths of the unnecessary test sequence for combinational modules. Thus, the area shows the value of the cost function $F$. The minimum value of $F$ is 5. Assuming that the values of $p$, $q$, and $r$ are 6, 4, and 10, respectively, the following constraints are satisfied in this grouping.

$$\max_j\left(\sum_{I=1}^{n}(X_{ij} \times L_i)\right) = \max(6,4) = 6 \leq p(=6)$$

$$\max_j(GNC_j) = \max(3,4) = 4 \leq q(=4)$$

$$\max_j\left(DD_{CT_j}\right) = \max(8,4) = 8 \leq r(=10)$$

After grouping, $T_1$ and $T_3$ are compacted, and PCTPT$_1$ shown in Table 5 is generated. Likewise, $T_2$ and $T_4$ are compacted, and PCTPT$_2$ shown in Table 5 is generated.

**Example 9:** Let us now assume the value of $p$ is 5. The values of $q$ and $r$ are the same as those in Example 8. $T_1$ and $T_3$ are grouped, and $T_2$ and $T_4$ are grouped. The following constraint is not satisfied in this grouping.

$$\max_j \left( \sum_{i=1}^{n} (X_{ij} \times L_i) \right) = \max(6, 4) = 6 \leq p(= 5)$$

Thus, this grouping is invalid. Another grouping is tried. $T_1$ and $T_2$ are grouped, and $T_3$ and $T_4$ are grouped as shown in Fig. 5. The value of $F$ is 20. Thus, the total sum of the lengths of the unnecessary test sequence becomes larger than that of Example 8. The following constraints are satisfied in this new grouping.

$$\max_j \left( \sum_{i=1}^{n} (X_{ij} \times L_i) \right) = \max(5, 5) = 5 \leq p(= 5)$$
$$\max_j (GNC_j) = \max(4, 4) = 4 \leq q(= 4)$$
$$\max_j \left( DD_{CT_j} \right) = \max(2, 6) = 6 \leq r(= 10)$$

After grouping, $T_1$ and $T_2$ are compacted, and PCTPT$_1$' shown in Table 7 is generated. Likewise, $T_3$ and $T_4$ are compacted, and PCTPT$_2$' shown in Table 7 is generated. In Table 7, $b_i$'s correspond to a test pattern for each combinational module $i$ $(i = 1, 2, \ldots, n)$. $b_i$'s are replaced with the test pattern (0's and/or 1's). In this example, the value of cost function $F$ increases from 5 to 20. Thus, the test length is affected by the values of the parameters.

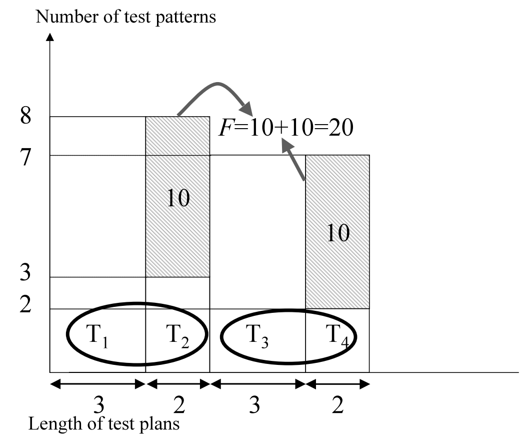

**Fig. 5** Another example of test plan grouping ($m = 2$).

**Table 7** Two PCTPTs (Example 9).

PCTPT$_1$'

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | $b_1$ | 0 | X | X | X |
| 1 | $b_1$ | X | X | $b_1$ | X |
| 2 | $b_2$ | X | X | 0 | X |
| 3 | X | X | $b_2$ | X | 0 |

PCTPT$_2$'

| Time | $P_1$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ |
|---|---|---|---|---|---|
| 0 | $b_3$ | $b_3$ | 0 | 1 | X |
| 1 | X | 0 | 1 | X | X |
| 2 | $b_4$ | 0 | 0 | 1 | $b_4$ |
| 3 | X | X | X | X | X |

### 3.4 Test Generation

After a gate level circuit for a combinational module is synthesized, test generation is performed for a single stuck-at-fault in a combinational module. As a result, test patterns for a combinational module are generated. Next, $b_i$'s corresponding to a test pattern for each combinational module $i$ $(i = 1, 2, \ldots, n)$ are replaced with the test pattern (0's and/or 1's). The above-mentioned processing is iterated for all test patterns for each combinational module $i$. The test length for a data path with strong testability is given by

$$L = \sum_{j=1}^{m} \left( MAXTP_j \times (L_{PCTPT_j} + 1) \right), \qquad (4)$$

where $MAXTP_j = \max_i (X_{ij} \times N_i)$, $L_{PCTPT_j}$ is the length of $PCTPT_j$, $N_i$ is the number of test patterns for a combinational module $i$, and $L$ is the test length for a data path.

**Example 10:** In the PCTPTs shown in Table 5, the bit width of the TPR is 2. The values of control signals are loaded into the TPR at time $-1$ which is earlier than time 0 in the PCTPT by one cycle. The number of states in the FSM is 4 because the length of $\max_j (L_{PCTPT})$ is 4. The test length for the data path is, according to equation (4), $8\times(4+1)+3\times(3+1) = 52$.

**Example 11:** In the PCTPTs shown in Table 7, the bit width of the TPR is 2. The values of control signals are loaded into the TPR at time $-1$ which is earlier than time 0 in the PCTPT by one cycle. The number of states in the FSM is 4 because the length of $\max_j \left( L_{PCTPT_j} \right)$ is 4. The test length for the data path is, according to equation (4), $8\times(4+1)+7\times(4+1) = 75$.

## 4. Experimental Results

In this section, the experimental results of the test generation method using several PCTPTs are described by applying it to some practical RTL data paths.

The platform of the preliminary experiments is as follows.

CPU: Pentium III, Frequency: 1 GHz,
and Memory: 512 Mbyte.

The characteristics of the practical RTL data paths with strong testability are shown in Table 8. Circuit, #PI, #PO, #CS, #ST, #R, #M, and |bit| denote the circuit name, the number of primary inputs, the

**Table 8** Characteristics of RTL data paths.

| Circuit | #PI | #PO | #CS | #ST | #R | #M | \|bit\| |
|---|---|---|---|---|---|---|---|
| RISC | 32 | 96 | 177 | 5 | 47 | 115 | 32 |
| DCT-C | 96 | 224 | 112 | 2 | 22 | 312 | 32 |
| IDCT-C | 96 | 224 | 135 | 6 | 27 | 349 | 32 |
| MPEG | 56 | 128 | 589 | 0 | 241 | 368 | 8 |

**Table 9** Experimental results.

| Circuit | PCTPT | | | | | PCTPT-TA10 | | | | | | | | PCTPC-TA30 | | | | | | | | TP | | CTPT | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | m | TL | TA | $R_{TL}$(%) | $R_{TA}$(%) | m | TL | TA | $R_{TL}$(%) | $R_{TA}$(%) | p | q | r | m | TL | TA | $R_{TL}$(%) | $R_{TA}$(%) | p | q | r | TL | TA | TL | TA |
| RISC | 8 | 3216 | 1637 | 37.8 | 75.5 | 97 | 4924 | 1003 | 4.8 | 7.5 | 8 | 32 | 14 | 54 | 4219 | 1188 | 18.4 | 27.3 | 10 | 38 | 16 | 5172 | 933 | 41940 | 1535 |
| DCT-C | 7 | 10392 | 4552 | 54.8 | 10.8 | 124 | 14487 | 4469 | 36.3 | 8.7 | 24 | 195 | 36 | 7 | 10392 | 4552 | 54.8 | 10.8 | | | | 22744 | 4110 | 23630 | NA |
| IDCT-C | 8 | 15316 | 4427 | 44.9 | 105.3 | 160 | 17818 | 1950 | 36.0 | -9.6 | 24 | 205 | 36 | 40 | 15901 | 2752 | 42.8 | 27.6 | 96 | 287 | 144 | 27819 | 2157 | 64912 | NA |
| MPEG | 7 | 77011 | 9019 | 30.9 | 122.3 | | | | | | | | | 32 | 104457 | 4925 | 6.3 | 21.5 | 384 | 412 | 576 | 111495 | 4052 | 96269 | NA |

number of primary outputs, the number of control signals, the number of status signals, the number of registers, the number of combinational modules, and the bit widths of data path signals, respectively. The logic synthesis was performed using the Design Compiler® of Synopsys and the test generation for each combinational module was performed using the TetraMax® ATPG of Synopsys.

The proposed method was compared with two conventional methods: the test generation method using test plans [5], and the test generation method using a CTPT [12]. In the test generation method using a CTPT, the test controller area could not be synthesized for three data paths in a reasonable time (less than 24 hours) because each $L_{CTPT}$ is large and the number of control signals is large as described in subsection 2.4. Therefore, we concluded that it is difficult to apply the test generation method using a CTPT to practical data paths. Thus, we will not refer to the test generation method using a CTPT from now on.

In Table 9, Circuit denotes circuit name, "PCTPT" shows the experimental results of the test generation method using several PCTPTs, "PCTPT-TA10" shows the experimental results of the test generation method using several PCTPTs with less than 10% additional test controller area constraint, "PCTPT-TA30" shows the experimental results of the test generation method using several PCTPTs with less than 30% additional test controller area constraint, "TP" shows the experimental results of the test generation method using test plans [5], and "CTPT" shows the experimental results of the test generation method using a CTPT [12]. In Table 9, $m$ denotes the number of groups, TL denotes the test length for data paths, and TA denotes the area of a test controller. $R_{TL}$ and $R_{TA}$ of "PCTPT" are defined as follows. $R_{TL}$ and $R_{TA}$ of "PCTPT-TA10" ("PCTPT-TA30") are also defined in the same way as those of "PCTPT".

$$R_{TL} = (\text{TL of "TP"} - \text{TL of "PCTPT"}) \,/\, \text{TL of "TP"}$$
$$R_{TA} = (\text{TA of "PCTPT"} - \text{TA of "TP"}) \,/\, \text{TA of "TP"}$$

In "PCTPT", the experiments were iterated changing the values of the parameters manually until the test controller was synthesized in about 10 hours. As a result, parameters $p$, $q$, and $r$ were set to 2000, infinity, and 3000. Given $m$, test plans were partitioned into $m$ groups to shorten test length. "PCTPT" shortened the test length by 30 to 54% compared with "$TP$". However, the test controller area increased by 10 to 122% compared with "TP". As for test controller area, the area of "TP" was minimum except for IDCT-C.

**Table 10** Characteristics of test controllers.

| Circuit | TG | TA | TMR | TPR | #State | FSM | Decoder | MUX |
|---|---|---|---|---|---|---|---|---|
| RISC | TP | 933 | 49 | 7 | 5 | 98 | 119 | 660 |
| | PCTPT | 1637 | 21 | 175 | 80 | 450 | 739 | 252 |
| | PCTPT-TA10 | 1003 | 49 | 28 | 7 | 120 | 164 | 642 |
| | PCTPT-TA30 | 1188 | 42 | 28 | 9 | 133 | 386 | 599 |
| DCT-C | TP | 4110 | 63 | 0 | 10 | 264 | 307 | 3476 |
| | PCTPT | 4552 | 21 | 0 | 236 | 885 | 2984 | 662 |
| | PCTPT-TA10 | 4469 | 49 | 0 | 15 | 299 | 1397 | 2724 |
| | PCTPT-TA30 | 4552 | 21 | 0 | 236 | 885 | 2984 | 662 |
| IDCT-C | TP | 2157 | 63 | 0 | 12 | 309 | 80 | 1705 |
| | PCTPT | 4427 | 21 | 0 | 290 | 953 | 2969 | 484 |
| | PCTPT-TA10 | 1950 | 56 | 0 | 14 | 301 | 252 | 1341 |
| | PCTPT-TA30 | 2752 | 42 | 0 | 56 | 408 | 1265 | 1037 |
| MPEG | TP | 4052 | 63 | 0 | 65 | 1101 | 30 | 2858 |
| | PCTPT | 9019 | 21 | 0 | 1488 | 4381 | 3376 | 1241 |
| | PCTPT-TA30 | 4925 | 35 | 0 | 229 | 1359 | 2139 | 1392 |

In "PCTPT-TA10" ("PCTPT-TA30"), the experiments were iterated changing the values of the parameters manually until the test controller area constraint was satisfied and the test length was shortened. As a result, $p$, $q$, and $r$ were set to appropriate values to shorten test length with less than 10% (30%) additional test controller area compared with "TP".

Thus, the following (1), (2), and (3) are taken into account.

(1) $p$ affects the area of the FSM in the TPG.
(2) $q$ affects the area of the MUX in the TPG.
(3) $r$ affects the area of the Decoder in the TPG.

"PCTPT-TA10" shortened the test length by 4 to 36% with less than 10% additional test controller area compared with "TP". "PCTPT-TA10" could not find the values of the parameters to shorten the test length with less than 10% additional test controller area for MPEG. "PCTPT-TA30" shortened the test length by 6 to 54% with less than 30% additional test controller area compared with "TP". As for IDCT-C, "PCTPT-TA10" reduced the test controller area by 9% compared with "TP".

Table 10 shows the detailed area of the test controllers. In Table 10, "TG" denotes a test generation method, "TA" denotes the test controller area, "TMR" denotes the area of the TMR, "TPR" denotes the area of the TPR, "#State" denotes the number of states in the FSM, "FSM" denotes the area of the FSM in the TPG, "Decoder" denotes the area of the Decoder in the TPG, and "MUX" denotes the area of the MUX in the TPG. As for "PCTPT", "TMR", and "MUX" were reduced, and "TPR", "FSM", and "Decoder" were increased compared with "TP". As for "PCTPT-TA10" and "PCTPT-TA30", by setting the appropriate value to $p$, the maximum value of the lengths of PCTPTs was shortened, and "FSM" was reduced compared with "PCTPT". By setting the appropriate value to $r$, the maximum value of the density degree of PCTPTs was reduced, and "Decoder" was reduced compared with "PCTPT". By setting the appropriate value to $q$, the maximum number of control signals where the values in PCTPTs were supplied was reduced, "MUX" was reduced compared with "TP", and "MUX" was increased compared with "PCTPT".

## 5. Conclusion

This paper proposed a test generation method using several PCTPTs for RTL data paths with strong testability. The optimization problem for test plan grouping is also formulated using ILP to shorten test length under a test controller area constraint. Experimental results for practical RTL data paths show that the test lengths are shortened by 4 to 36% with less than 10% additional test controller area and the test lengths are shortened by 6 to 54% with less than 30% additional test controller area compared with the test generation method using test plans.

The algorithm to find the optimum values of the parameters is under development. In this paper, the values of the parameters are manually set and test plans are partitioned into some groups. We have shown that it is possible to find the values of the parameters that minimize the test length under a test controller area constraint. Future work includes proposing an effective algorithm for finding the optimum values of the parameters.

## Acknowledgements

This work was sponsored by NEDO (New Energy and Industrial Technology Development Organization) as VCDS Project (SoC advanced design technology development project). The Authors would like to thank Professor Michiko Inoue, Professor Satoshi Ohtake, and Professor Tomokazu Yoneda of Nara Institute of Science and Technology for their valuable discussion and comments. The Authors would like to thank Professor Tomoo Inoue and Professor Ichihara of Hiroshima City University for his valuable discussion and comments. The Authors would like to thank Dr. Rafael K. Morizawa of Semiconductor Technology Academic Research Center for his valuable comments.

### References

[1] H. Fujiwara, Logic Testing and Design for Testability, The MIT Press, 1985.

[2] M. Abramovici, M.A. Breuer, and A.D. Friedman, Digital systems testing and testable design, IEEE Press, 1995.

[3] T. Hosokawa, M. Yoshimura, and M. Ohta, "Design for testability strategies using full/partial scan designs and test point insertions to reduce test application times," Proc. Asia and South Pacific Design Automation Conference, pp.485–491, 2001.

[4] P.C. Maxwell, R.C. Aitken, V. Johansen, and I. Chiang, "The effect of different test sets on quality level prediction: When is 80% better than 90%?," Proc. International Test Conference, pp.358–364, 1991.

[5] S. Ohtake, H. Wada, T. Masuzawa, and H. Fujiwara, "A non-scan DFT method at register-transfer level to achieve complete fault efficiency," Proc. Asia and South Pacific Design Automation Conference, pp.599–604, 2000.

[6] S. Ohtake, S. Nagai, H. Wada, and H. Fujiwara, "A DFT method for RTL circuits to achieve complete fault efficiency based on fixed-control testability," Proc. Asia and South Pacific Design Automation Conference, pp.331–334, 2001.

[7] I. Ghosh, A. Raghunathan, and N.K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.16, no.9, pp.1001–1014, 1997.

[8] I. Ghosh, A. Raghunathan, and N.K. Jha, "Hierarchical test generation and design for testability methods for ASPP's and ASIP's," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.18, no.3, pp.357–370, 1999.

[9] H. Wada, T. Masuzawa, K.K. Saluja, and H. Fujiwara, "Design for strong testability of RTL data paths to provide complete fault efficiency," Proc. 13th Int. Conf. on VLSI Design, pp.300–305, 2000.

[10] S. Nagai, S. Ohtake, and H. Fujiwara, "A design for hierarchical testability for RTL data paths using extended data flow graphs," Proc. Workshop on RTL ATPG & DFT (WRTLT), pp.128–133, 2001.

[11] B.T. Murray and J.P. Hayes, "Hierarchical test generation using pre computed tests for modules," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.9, no.6, pp.594–603, 1990.

[12] T. Hosokawa, H. Date, and M. Muraoka, "A test generation method using a compacted test table and a test generation method using a compacted test plan table for RTL data path circuits," Proc. 20th IEEE VLSI Test Symposium (VTS2002), pp.328–335, 2002.

**Toshinori Hosokawa** received the B.E. degree in Electronics and Communication Engineering from Meiji University, Kawasaki, Japan, in 1987. He also received the D.E. degree from Meiji University in 2001. From 1987 to 2003, he was with Matsushita Electric Industrial Co., Ltd. and worked on logic simulation engine, automatic test pattern generation, fault simulation, design for testability and high level testing. From 2000 to 2003, he joined Semiconductor Technology Academic Research Center (STARC) and worked on testing for system on a chip and hardware/software co-verification. He was also a lecturer at Meiji University in 2001 and 2002. Since 2003, he has been an associate professor of the College of Industrial Technology, Nihon University. His research interests are CAD technologies for system LSI testing, including high-level synthesis for testability, design for testability, built-in self test, and test pattern generation, and hardware/software co-design. He is a member of IEEE (Institute of Electrical & Electronics Engineers) and IPSJ (Information Processing Society of Japan).

**Hiroshi Date** received the B.S. and M.S. degrees in Science of Mathematics from Kyushu University, Fukuoka, Japan, in 1985 and 1987, respectively. He also received the D.E. degree from Kyushu University in 1993. From 1987 to 1996, he was with Hitachi Research Laboratory, Hitachi Ltd. and was engaged in VLSI-CAD and parallel processing. From 1990 to 1993, he joined Institute for New Generation Computer Technology (ICOT) and worked on VLSI-CAD using parallel processing. From 1996 to 2001, he was with Institute of Systems & Information Technologies / KYUSHU. From 1999 to 2001, he was an associate professor of Kyushu University. From 2001 to 2002, he was with Abel systems Inc. Since 2000, he has joined Semiconductor Technology Academic Research Center. Since 2002, he has been a chairman & CEO of System J D Co., Ltd. His research interests include design & test methodology and CAD technologies for system LSI, and information security. He is a member of IPSJ (Information Processing Society of Japan), TTTC (Test Technology Technical Council), ACM (Association for Computing Machinery), and IEEE (Institute of Electrical & Electronics Engineers).

**Masahide Miyazaki** received the B.E. and M.S. degrees in Space Science from Nagoya University, Nagoya, Japan, in 1990 and 1992, respectively. In 1992, he joined Hitachi Ltd. He has worked on sequential redundancy, and design for testability. In 2002, he went to Semiconductor Technology Academic Research Center (STARC) as a Hitachi assignee, where is currently working on testing for system on a chip. He entered the graduate program of the Graduate School of Information Science, Nara Institute of Science and Technology in the Autumn of 2003.

**Michiaki Muraoka** received the B.E. degree in Electrical Engineering from Keio University, Tokyo, Japan in 1974. He worked on the research and development of system and semiconductor EDA system, especially in the logic and RTL design and silicon compilation at OKI Electric from 1974 to 1988. He joined Matsushita Electric Industrial Company where he worked on the research and development of EDA technology and IP based design methodology as the technical leader from 1988 to 2000. He was temporarily assigned to STARC in 2000 as a senior manager of VCDS Development Group. His current research interests are in the areas of the next generation design methodology and EDA technology for the giga scale SoC (System on Chip).

**Hideo Fujiwara** received the B.E., M.E., and Ph.D. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1974, respectively. He was with Osaka University from 1974 to 1985 and Meiji University from 1985 to 1993, and joined Nara Institute of Science and Technology in 1993. In 1981 he was a Visiting Research Assistant Professor at the University of Waterloo, and in 1984 he was a Visiting Associate Professor at McGill University, Canada. Presently he is a Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. His research interests are logic design, digital systems design and test, VLSI CAD and fault tolerant computing, including high-level/logic synthesis for testability, test synthesis, design for testability, built-in self-test, test pattern generation, parallel processing, and computational complexity. He is the author of Logic Testing and Design for Testability (MIT Press, 1985). He received the IECE Young Engineer Award in 1977, IEEE Computer Society Certificate of Appreciation Award in 1991, 2000 and 2001, Okawa Prize for Publication in 1994, IEEE Computer Society Meritorious Service Award in 1996, and IEEE Computer Society Outstanding Contribution Award in 2001. He is an advisory member of IEICE Trans. on Information and Systems and an editor of IEEE Trans. on Computers, J. Electronic Testing, J. Circuits, Systems and Computers, J. VLSI Design and others. Dr. Fujiwara is a fellow of the IEEE, a Golden Core member of the IEEE Computer Society, and a member the Information Processing Society of Japan.