IEICE TRANS. INF. & SYST., VOL.E94–D, NO.1 JANUARY 2011

PAPER

F-Scan: A DFT Method for Functional Scan at RTL

Marie Engelene J. OBIEN[†a)], *Nonmember*, **Satoshi OHTAKE**[†], *Member*, *and* **Hideo FUJIWARA**[†], *Fellow*

SUMMARY Due to the difficulty of test pattern generation for sequential circuits, several design-for-testability (DFT) approaches have been proposed. An improvement to these current approaches is needed to cater to the requirements of today's more complicated chips. This paper introduces a new DFT method applicable to high-level description of circuits, which optimally utilizes existing functional elements and paths for test. This technique, called F-scan, effectively reduces the hardware overhead due to test without compromising fault coverage. Test application time is also kept at the minimum. The comparison of F-scan with the performance of gate-level full scan design is shown through the experimental results.

key words: *scan-based DFT, functional RTL circuits, high-level testing, assignment decision diagrams*

1. Introduction

As VLSI Design becomes more complicated due to the trends of minimizing chip size and maximizing speed, the importance of testing has increased to ensure the quality of electronic consumer products. A concern is the difficulty of sequential test generation. To reduce the exponential complexity of sequential automatic test pattern generation (ATPG), various design for testability (DFT) approaches have been proposed. Accordingly, the circuit structure and functionality change during test mode to allow easier testing. The most popular approach is scan design, which increases the testability of sequential circuits considerably [1]. Full Scan Design is widely used because it effectively reduces the sequential circuit ATPG problem into a combinational one.

The trade-offs of DFT, as demonstrated by full scan at gate-level, are: 1) very large test hardware overhead and 2) significantly long test application time. These penalties prove DFT, particularly full scan, to be very costly, especially for high-volume, low-cost applications. While the disadvantages of DFT hold true, our proposed DFT technique reduces chip area overhead and test application time as much as possible so that for high-density circuits, such overhead can be negligible. Moreover, we apply DFT to register-transfer level (RTL) circuits wherein the number of primitive elements in the circuit is reduced.

There have been several DFT techniques proposed, both scan and non-scan based. Historically, Gupta et al. [3] introduced an approach to RTL DFT, which is a structured partial scan design that converts only the selected flip-flops into scan flip-flops. However, full-scan-based approaches ensure stronger testability of circuits. Cost-free scan design [2] was first proposed for gate-level circuits to improve the area overhead of full scan design. H-Scan [4], [6] is a full-scan-based technique that utilizes paths between registers, but only through multiplexers. Although it can achieve the same fault coverage as full scan, further area overhead reduction can still be achieved. An improvement is orthogonal scan [5], which uses data path flow as scan path. This method though requires multiple test configurations because it uses hold functions through load enable. *Hold function* is a logic that causes a register to hold the same value when the function is activated. This is necessary when a functional logic is shared by two scan paths because it allows scanning-in and -out of vectors from these paths one at a time, thus allowing the shared element to be used for testing. Our method does not employ this kind of function (with the exception of handling some state registers) because of the disadvantages of adding extra pins for controlling multiple paths during test and the expected longer test application time because simultaneous scan-in and -out cannot take place. Although we use some sort of initialization to scan-in the state register value first, which is a kind of hold function, we do not use *hold* whenever a functional operation is shared by candidate F-scan-paths.

The following works improved the previous methods mentioned. Huang et al. [7] proposed the arrangement of registers in scan chains through cost rules to ensure the lowest possible area overhead for the circuit. Though this method tries to exploit available functional logic as much as possible without the use of hold functions, mask function is not considered. A *mask function* can be applied to operation logic, wherein the value from one input can be passed through the output by masking the other inputs. This function further reduces area overhead, which is a DFT element widely used by our method. D-scan [8], on the other hand, uses *thru functions* (logic that allow values to pass through hardware modules) with predetermined control signals for the scan paths in the circuit. This work, however, utilizes hold functions to handle multiple paths that share the same thru function.

Techniques that utilize available circuity for test were also proposed in non-scan DFT techniques [10], [12], [15]. However, these approaches require a test controller and a means to isolate the controller part from the data path part,

Manuscript received March 23, 2010.
Manuscript revised August 3, 2010.

[†]The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Ikomashi, 630–0192 Japan.
a) E-mail: obienj@is.naist.jp

DOI: 10.1587/transinf.E94.D.104

thus increasing area overhead. Moreover, these methods are applicable to structural description of circuits, while our method handles functional RTL. Most of the designers are increasingly using functional description of circuits, which makes our proposed technique more relevant. Furthermore, our method deals with the circuit in assignment decision diagrams (ADD), which represent both the controller and data path parts similarly. Thus, the application of both the DFT method and test is consistent for the entire circuit. The use of ADD also allows for easier manipulation of the circuit for DFT.

Our new approach to functional scan, F-scan, improves all of the mentioned previous works in terms of area overhead. F-scan organizes every register in the circuit in an F-scan-path by maximizing the use of available functional logic and paths to be used during scan, hence keeping hardware overhead due to test at the minimum. F-scan approach improves the other previously proposed methods (e.g., H-Scan, non-scan DFT) due to the following characteristics: (a) F-path instead of identity path or I-path. F-scan has the ability to use paths between registers with operations that can be masked, not just paths with multiplexers. (b) F-scan is applied on functional RTL circuits, unlike the others that are done on structural RTL circuits. F-scan has the ability to be applied on the entire circuit uniformly, unlike the different approaches proposed before for data path and controller parts. Moreover, F-scan is a scan approach that can run-under-test using system clock, similar to at-speed testing of non-scan based methods. Furthermore, F-scan prioritizes candidates to create F-scan-paths with the least possible scan time. Single F-scan-paths automatically allow parallel and simultaneous scan (dependent on the bit width), thus test application time is minimized. For further reduction, we also prioritize the use of multiple F-scan-paths, whenever readily available (dependent on the available primary inputs and outputs). The new concepts and methodology to create *F-scannable* circuits are provided in this paper.

The rest of the paper is as follows. In Sect. 2, F-scan and other preliminary concepts such as ADD are introduced. We describe the details of F-scan design methodology in Sect. 3. We also explain the procedure for test environment generation in Sect. 4. The experimental results are provided in Sect. 5 and the conclusion in Sect. 6.

2. F-Scan

In order to define *functional scan*, we first give a brief introduction about assignment decision diagrams and other preliminary concepts.

2.1 ADD and the Nine Symbol Algebra

Assignment Decision Diagram or ADD shown in Fig. 1 is a representation developed for high-level synthesis that is complete, efficient, and partially unique. It can be used to describe functional RTL circuits in which the controller and the data path are consistently represented.

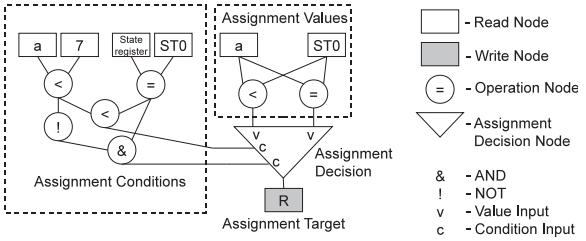


Fig. 1 The assignment decision diagram.

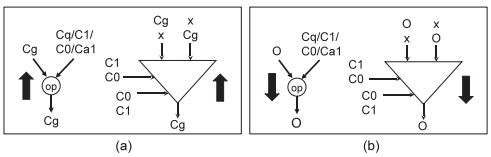


Fig. 2 (a) General controllability and (b) observability of operation nodes and ADNs.

ADD consists of four types of nodes: a) read nodes and b) write nodes (primary inputs or PI and outputs or PO, registers, or constants), c) operation nodes (arithmetic and logic), and d) assignment decision nodes or ADN (multiplexers) [16].

The concept of functional scan uses the following nine-symbol algebra used by Ghosh [17] for automatic test pattern generation (ATPG) of ADD circuits.

1. Cg (general controllability) of a register means it can be controlled to any arbitrary value.
2. Cq (controllability to a constant) of a register means it is controllable to any fixed constant value. This subsumes $C0$(controllability to zero), $C1$(controllability to one), and $Ca1$(controllability to all one).
3. O (observability) of an RTL variable is the ability to observe fault at a variable.
4. Cs (controllability to a state) is similar to Cq but is applied to state registers to control to a particular state.
5. Other symbols are Cz (controllability to the Z value) and O' (complement observability), but these are not used for our study.

In Fig. 2, controllability and observability in functional scan are illustrated with the use of these symbols. In Fig. 2 (a), we see that a value can be passed through an operation node as long as the other inputs to the node (side inputs) are constants such as Cq, $C0$, $C1$, and $Ca1$. Any arbitrary value can also pass through an available ADN by manipulating its control inputs to $C0$ and $C1$. Similarly, we can observe through operation nodes and ADNs as shown in Fig. 2 (b).

2.2 Functional Scan

We introduce the new concepts of functional scan by describing the means of justification and propagation in an ADD circuit. In Fig. 2 (a), we see that any arbitrary value can pass through available operation nodes and ADN, given

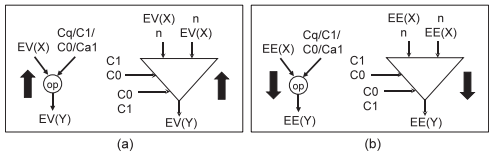


Fig. 3 (a) Essential value justification and (b) Essential error propagation.

by the symbol Cg, as long as the other inputs (side inputs) to the nodes are constants. Similarly, we can observe through operation nodes and ADNs as shown in Fig. 2 (b). Given these, we have the following definitions. Refer to Fig. 5 for clarification of variables used in these definitions.

Assume that $p(X, Y)$ is a path from a read node X to a write node Y such that the side inputs of operational nodes and control inputs of ADNs along the path p can be controlled to fixed constants in an ADD circuit A.

Definition 1: $EV(Y)$ is the ***essential value set*** of Y such that it is a set of values that can be essentially assigned to Y, which means the set of all values assignable to Y according to the functionality of A.

Definition 2: $EE(X)$ is the ***essential error set*** of X such that it is a set of errors that can be essentially detected from X, which means the set of all errors detectable from X. An error can be detected from the difference between a faulty and a fault-free value.

Definition 3: ***Essential Value Justification*** (EVJ) for $p(X, Y)$. Any value in $EV(Y)$ can be justified at Y by $p(X, Y)$ provided that any value in $EV(X)$ is justified at X.

Definition 4: ***Essential Error Propagation*** (EEP) for $p(X, Y)$. Any error in $EE(X)$ can be propagated to Y by $p(X, Y)$.

Figure 3 illustrates how available functional logic is exploited for testing. Each register node can be both ***essentially justifiable*** and ***essentially propagable*** by controlling the side inputs along the involved path to $Cq/C0/C1/Ca1$. This means that for operation nodes in path p, since we know the constant value of the other input(s), we can compute for the value of the input (X) such that any arbitrary value within $EV(Y)$ can be passed to Y to make it *essentially justifiable*. On the other hand, in order for an error in $EE(X)$ to *essentially propagate* through an operation node in path p, the difference between faulty and fault-free values should be detectable from X through Y. For ADNs in path p, any value/error can be retrieved from the ADN by controlling which input of the ADN connected to an essentially justifiable/propagable read node will pass its value/error to the essentially justifiable/propagable write node.

Definition 5: ***Functional scan*** (abbrev. F-scan) is satisfied when all registers are made essentially justifiable and essentially propagable to be used for F-scan function. F-scan is a concept that uses available functional elements and paths to create scan chains for testing.

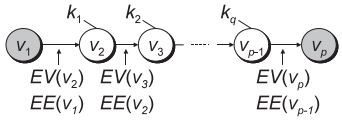


Fig. 4 General representation of F-path.

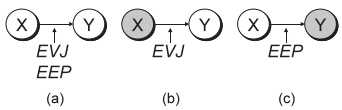


Fig. 5 Essential F-path Illustrated. (a) Case 1, (b) Case 2, and (c) Case 3.

The value ranges are obtained from the description of the ADD circuit and the error set depends on the fault model. This means that the ADD circuit may be augmented differently for various fault models for F-scan to be satisfied. To handle all errors, *complete error propagation* may be used instead. Similarly, if it is difficult to augment the circuit in order to obtain EVJ, *complete value justification* will be considered.

Definition 6: ***Complete Value Justification*** (CVJ) for $p(X, Y)$. Any value can be justified at Y by $p(X, Y)$ provided that any value is justified at X.

Definition 7: ***Complete Error Propagation*** (CEP) for $p(X, Y)$. Any error at X can be propagated to Y by $p(X, Y)$.

Both CVJ and CEP are strong conditions that are equivalent to justification and propagation conditions in full scan design.

2.3 F-Paths and F-Scan-Paths

The difference between gate-level full scan and F-scan is the method of building scan paths. Gate-level full scan arranges all flip-flops in single or multiple chains to shift test vectors while F-scan includes all registers in one or more scan chains called F-scan-paths, wherein the least possible scan time is achieved. While full scan augments multiplexers to connect flip-flops, F-scan exploits available functional elements and paths. F-scan-paths also allow scan-in and -out test vectors simultaneously, thus, similar to full scan, only one test pin is needed to activate scan to handle all registers. Another test pin will be needed to handle the state register, which is further discussed in Sect. 3.

We defined ***F-path*** in [19], which represents the topology of a path in an ADD circuit from a read node to a write node as shown in Fig. 4. Between the read node (PI or register), v_1, and the write node (PO or register), v_p, there may be operation nodes (v_2 to v_{p-1}) and ADN where value or error can *pass* through the path. Side inputs along the path should be made constant (k_1 to k_q).

Considering the path $p(X, Y)$ in the previous subsection and by referring to Fig. 5, we have the following definitions.

Definition 8: $p(X, Y)$ is an ***Essential F-path*** if:

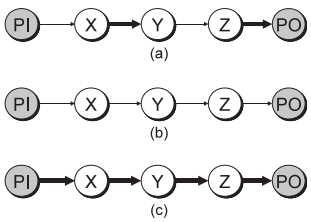


Fig. 6 F-scan-path Illustrated. (a) Single F-scan-path, (b) Essential F-scan-path, and (c) Complete F-scan-path.

- Case 1. X and Y are both registers: $p(X, Y)$ satisfies both essential value justification and essential error propagation for $p(X, Y)$.
- Case 2. X is PI: $p(X, Y)$ satisfies essential value justification for $p(X, Y)$.
- Case 3. Y is PO: $p(X, Y)$ satisfies essential error propagation for $p(X, Y)$.

Definition 9: $p(X, Y)$ is a ***Complete F-path*** if $p(X, Y)$ satisfies both complete value justification and complete error propagation for $p(X, Y)$.

Definition 10: ***Single F-scan-path*** A concatenation of F-paths wherein the head is a PI and the tail is a PO. There are two special cases of F-scan-path.

1. ***Essential F-scan path***. A concatenation of all essential F-paths.
2. ***Complete F-scan path***. A concatenation of all complete F-paths.

Definition 11: ***Multiple F-scan-path*** is a set of mutually compatible (disjoint) F-scan-paths.

Definition 12: An ADD circuit is said to be an ***F-scannable circuit*** if every register in the circuit is included in an F-scan-path, wherein it appears once and only once.

3. DFT Selection Method

We introduce a new functional RTL scan approach called *F-Scan design*, which makes any ADD circuit F-scannable. The preliminary concepts and the DFT algorithm are presented in this section.

3.1 Problem Formulation

In order to test an ADD circuit, we control and observe all read and write nodes by organizing all registers in F-scan-paths. Whenever there is no direct connection from a read node to a write node, the functional logic and path in between can be utilized by augmenting DFT elements that will allow these functional elements to be used for scan. A direct connection may also be augmented.

Definition 13: The DFT for F-scannable ADD circuits is formalized as the following optimization problem.

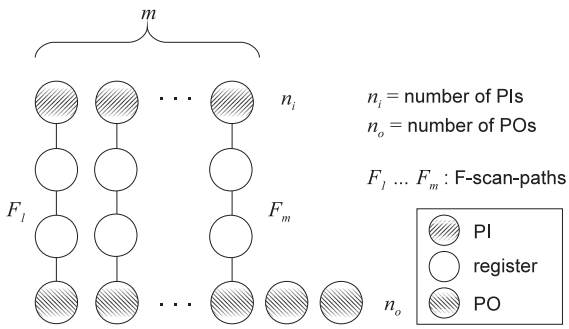


Fig. 7 Determining the number of F-scan-paths.

- **Input:** an ADD circuit
- **Output:** an F-scannable ADD circuit such that there are m F-scan-paths defined as

$$m = \min\{n_i, n_o\} \tag{1}$$

where n_i is the number of PIs and n_o is the number of POs. We can also solve the scan-length using m. Given the number of register nodes, k, we have:

$$\text{scan-length} = \left\lceil \frac{k}{m} \right\rceil. \tag{2}$$

Shown in Fig. 7 is an illustration of an ADD circuit with more POs than PIs. We choose m, which is the number of F-scan-paths $F_1, \ldots, F_m$, to be the minimum between the number of PIs (excluding reset and clock pins) and POs.
- **Optimization:** Minimize area overhead (i.e., hardware of augmented DFT elements)
After determining the fixed number of F-scan-paths, we organize the registers to fit the computed scan-length per F-scan-path. There will be cases wherein the connection of registers in the circuit cannot satisfy the creation of F-scan-paths with length k/m. This means that other F-scan-paths may be longer. Since scan time is a condition, even if a longer F-scan-path can potentially reduce area overhead further, F-scan-path slicing is still considered.

To assure that the least scan time is achieved without adding extra PI and PO as much as possible, we consider the condition of having m F-scan-paths. However, there may be situations when the bit widths of the registers in the circuit do not match the bit widths of the available PIs and POs. For different cases, we do the following:

1. If a circuit has no PI and/or PO for data transfer and the registers have the same bit width, we augment PI or PO with bit width equal to that of the registers in the ADD circuit. If the registers do not have the same bit width, we determine which bit width is common to most number of registers and consider that for the bit width of PI and PO to be augmented. If there are other registers with higher bit width to that, we slice those registers.

2. If a circuit has a one-bit PI and PO, we do not augment any PI or PO even if the registers in the circuit have higher bit widths. In this case, one test cycle of F-scan will be equivalent to that of full scan design, but the available functional elements and paths will be utilized.
3. If a circuit has a one-bit PI (resp. PO) and PO (resp. PI) with higher bit width, and the registers in the circuit have bit widths equal or less than that of the PO (resp. PI), we augment PI (resp. PO) such that the bit width will be equal to that of the register with the highest bitwidth. If there are registers with higher bit width compared with the PO (resp. PI), then these registers will be sliced.
4. If the bit widths of the PI and PO in the circuit do not match, we choose the bit width of the F-scan-path according to the bit width that is common to most of the registers in the circuit. We then augment PI or PO or both to handle the F-scan-path during scan. For registers that have higher bit widths, slicing is done. For registers that have lower bit widths, we combine them to produce a group of registers with the same bit width as that of the F-scan-path.

Slicing is done by dividing a register according to the desired bitwidth and then, by connecting them in parallel through multiplexers. For example, if there is a 16-bit register to be sliced to two, the first eight bits will be connected to the other eight bits of the same register in parallel such that it will take two clock cycles to scan-in/out test vectors to the whole register. On the other hand, combining is done by scheduling the registers along the F-scan-path at the same time frame. This scheduling is explained more by the *test environment*, which will be discussed in the next section.

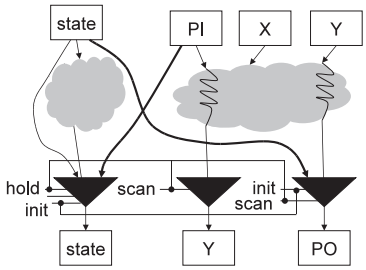
There is also a possible impact of the proposed DFT on logic synthesis. F-scan may introduce extra data flow at the ADD level. Some operations in the circuit may be involved in the extra data flow, which may prevent sharing of one operational module (at structural RTL) with other several operations (at functional RTL). If this happens, the resulting gate-level circuit may be large because the reduction of area during synthesis is not maximized. This is evaluated through our experiments.

3.2 Overview of the DFT Algorithm

The DFT algorithm consists of the following stages.

- **Stage 1.** Create a weighted connectivity graph (WCG) based on the information given by the ADD circuit. Here, all possible F-paths between each read/write node are exhaustively determined.
- **Stage 2.** Construct the F-scan-paths to make the circuit F- scannable.

Considering the number of possibilities, determining the F-scan-paths that are disjoint for an ADD circuit is regarded as an NP-hard problem. Thus, we employ a heuristic algorithm to simplify it. The details are described in the next subsection.



hold/scan	initialize	Mode
0	0	Normal
0	1	Initialize
1	0	Hold/Scan
1	1	Forbidden combination

Fig. 8 Augmentation to handle state registers.

3.3 DFT Algorithm Specifics

3.3.1 Handling State Registers

The state register is not readily accessible from PIs and POs and usually has a different bit width with the other registers, hence it cannot be readily included in the F-scan-paths. We augment the circuit to handle the state register as shown in Fig. 8. The bold lines indicate the added connection from PI to state registers and from state registers to PO. To test the state registers, we first initialize by scanning-in state value from the PI to the state register. Then, we set the test control inputs in scan mode for the entire circuit, while the state value is being held. When normal mode is done, new state value is scanned out during initialize, then we hold this value again to scan out the register values. Simultaneously, scan-in can occur while scanning-out. However, if there is a PI/PO pair available in the circuit that is not used by any F-scan-path or if there is an available F-scan-path that can include the state register, there is no need for the hold function.

3.3.2 New ADD Elements for Masking

Since there is no available ADD node that describes the ***mask function*** to keep an input to an operation node constant during scan, we have proposed the following new ADD elements in [19]. These elements are used as DFT elements for F-scan. Figure 9 illustrates the new ADD elements and their corresponding gate-level representation, which are saved in the library.

Definition 14: ***C0 mask.*** This mask is used for addition and subtraction operation nodes when the side input is not readily a constant. When the scan pin is set to 0, the output of this element is equal to the normal value of the line. If the scan pin is set to 1, the output of this element is 0.

Definition 15: ***C1 mask.*** This is used for multiplication and division operation nodes for them to pass any value from

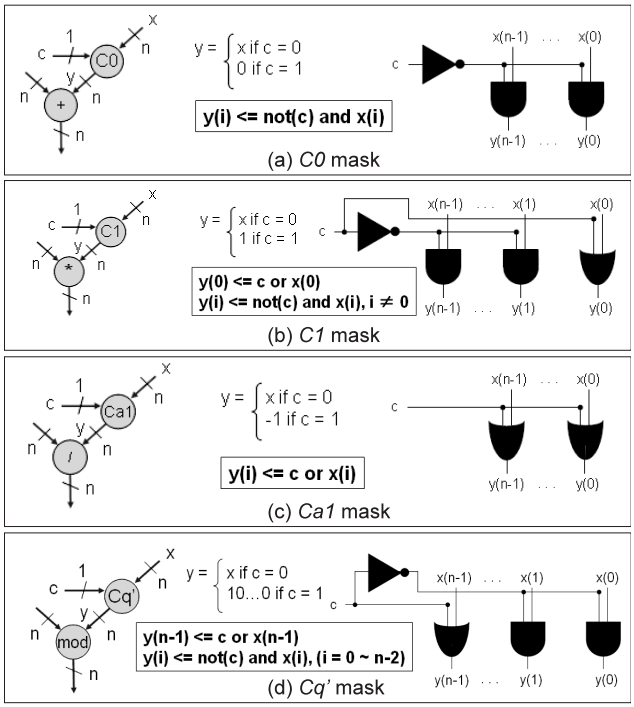


Fig. 9 New mask functions for ADD illustrated.

one input to the output without any changes. The output of this node is equal to the normal value of the line when the scan pin is set to 0. If it is set to 1, the output of this element is 1.

Definition 16: ***Ca1 mask.*** This is an alternative to $C1$ masking applicable to multiplication and division operation nodes as well. When the scan pin is set to 0, normal value of the line applies. If it is set to 1, all bits become 1.

Definition 17: ***Cq' masking for modulo.*** Since this constant is specific for modulo masking, we indicate it as Cq'. Being the highest $2n$ value within the range of the line, bitwise, the highest bit is 1 while the rest are zeros. This value $(10\ldots0)$ is the output of this node if the scan pin is set to 1. If it is set to 0, normal values of the line apply. This type of mask limits the range of a line, which is why using it is subject to the requirements of the essential ranges.

3.3.3 Weighted Connectivity Graph

The ***weighted connectivity graph*** (WGC) represents the topology of an ADD circuit, which includes the read/write nodes and the cost information derived from F-path candidates. The cost rules are defined in [19], which depend on the amount of circuitry to be augmented to realize the F-path.

Determining all possible paths from a read node to a write node is a problem that grows exponentially with the circuit size. Thus, essential F-path candidates for each read-write node pair are limited to a number of possible paths in the circuit, which is chosen by the designer depending

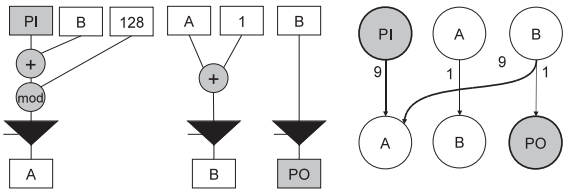


Fig. 10 Sample WCG extracted from an ADD Circuit.

on the size of the circuit. If incompatibilities exist for all the current F-path candidates, another path is determined and the cost is compared with the full scan cost. The path with the least cost is to be chosen. Another candidate is a complete F-path, of which the cost is similar to full scan. This is chosen only when no available essential F-path is compatible to be used in the F-scan-path or if there is no essential F-path obtained at all.

In Fig. 10, there are two candidate essential F-paths involving the read node B. The weights are indicated for each essential F-path. From A to B, for example, the cost 1 corresponds to the gate needed to augment the control input to activate this path during scan. Such information can be represented in WCG, where all paths merely indicate connectivity and weights.

3.3.4 Local Optimum Heuristic Approach

This ensures that in every local location (i.e., read-write node pairs in one scan time frame) the least area overhead due to test possible is achieved by choosing the candidate essential (or for the worst case, complete) F-paths that has the least cost. In the following, we only use the term F-path, which can be *either essential or complete*, depending on which has the least cost and is usable to create the F-scan-path.

1. ***PI/PO Priority.*** Once the number of F-scan-paths is determined, the primary inputs having F-paths to write nodes (registers) with the least cost are chosen. These F-paths are automatically the first in the F-scan-paths. Once chosen, backtrack is not applicable to change these F-paths (locked). Similarly, the F-paths with the least cost that connect read nodes (register) to primary outputs are chosen and locked in the F-scan-paths.
2. ***Controllability.*** Starting from the first F-path in each of the F-scan-paths, the next F-path is chosen from the rest of the unconnected F-path candidates such that it is the least cost. The process continues until the registers are arranged in F-scan-paths to guarantee EVJ that includes all registers. If in the process incompatibility is detected, backtrack is done until all F-scan-paths are mutually compatible.
3. ***Slicing.*** When the registers are arranged for control, it may occur such that one or more F-scan-paths are longer than the others. Since the length of the F-scan-paths is determined, we slice the long F-scan-paths and move the register or set of registers to shorter F-scan-paths to balance the lengths of all F-scan-paths.

4. ***Observability.*** To make all F-paths *EEP*, we finally connect all F-scan-paths to the F-paths connected to POs. We choose the connection such that it is the cheapest one.

4. Test Environment Generation Procedure

The testability of the circuit-under-test (CUT) is guaranteed if at least both *essential value justification* and *essential error propagation* are satisfied for all registers. After applying F-Scan DFT for the circuit, all the registers are guaranteed to be essentially justifiable and propagable. The ***test environment*** of the CUT therefore consists of the scheduled signal assignment values and scheduled output response needed to perform a complete F-scan cycle. It involves the F- scan-in phase, test phase, and F-scan-out phase, wherein F-scan-in and -out are overlapped.

Test patterns, on the other hand, are generated through an available ATPG tool after synthesizing the circuit to gate-level. The *test sequence* is then derived by embedding the test patterns to the test environment. This includes the input test vectors and the test response. We use the generated test sequence to test the F-scannable ADD circuit. This means that though the application of test sequence is done at ADD level, the generation of patterns is done at the gate-level. This means that the fault coverage obtained after applying ATPG on the gate-level combinational circuit is not real. This is because the circuit may change from ADD level to gate-level. Thus, in the case where the synthesized circuit is different from the ADD-level circuit, fault simulation has to be performed in order to determine the true fault coverage of the test patterns.

F-scan-in Phase. To do *F-scan-in*, all read nodes used for data transfer in the F-scan-paths must contain their respective test patterns in order to justify these patterns to the write nodes. The necessary read nodes that will activate the F-scan-paths should also be controlled to their activating values. The F-scan-in environment therefore includes the schedule of signal assignments that completes the F-scan-in phase. This schedule depends on the order of the registers in the F-scan-paths. Direct value assignments are scheduled according to which F-scan-paths are to be activated, e.g. 1 or 0 for scan/hold pin and initialize pin. One cycle in this phase ends when all registers satisfy *essential value justification*.

Test Phase. The *test phase* happens by setting the circuit to normal mode where all read nodes (registers) are used as input-registers and the same registers (also write nodes) are used as output-registers for testing the circuit. Here, the test-mode environment includes the PI values (if needed), the output response, and the scan/hold/initialization pins assignment that will turn the circuit to normal mode, i.e. zero value.

F-scan-out Phase. To complete the test environment, *F-scan-out* is done. F-scan-in phase and F-scan-out-phase are overlapped, i.e. pipelined, after the first scan-in cycle. Thus, the signals that activate F-scan-in also enable F-scan-out at the same time. This is illustrated in Fig. 11 (d).

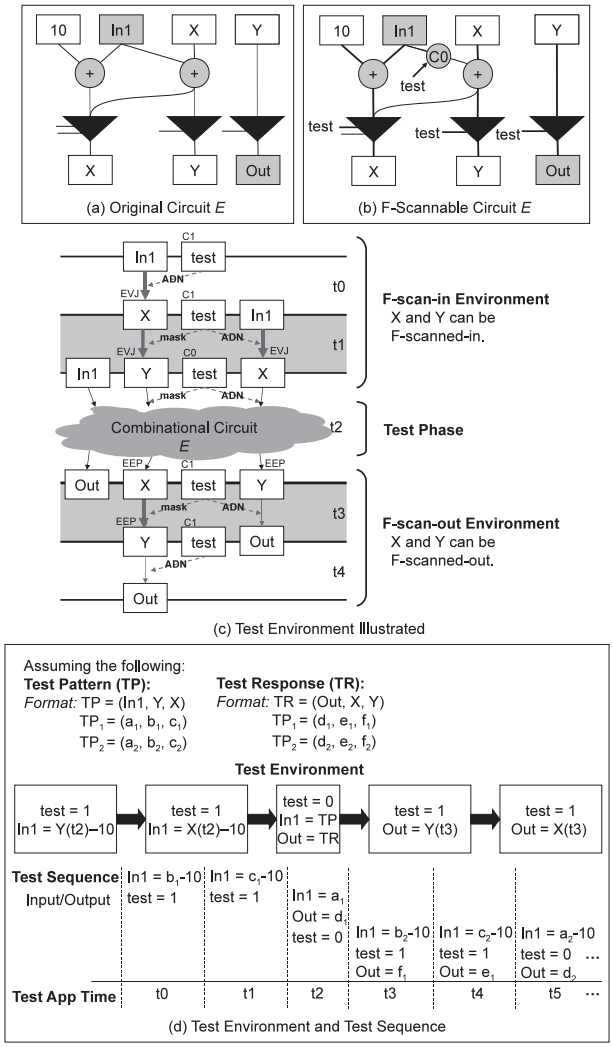


Fig. 11 Sample case to show the test process.

In F-scan-out phase, the values (and errors) in all registers are scanned-out. In order to check all the register values obtained after test phase, the test response, which is the output pattern of the circuit after test, is compared with the generated expected response.

Since the F-paths are not necessarily *I-paths* (or complete F-paths), the exact value of the generated test patterns may not be justified to the registers during F-scan-in. Also, the test response may not be the same with the expected response. Thus, adjustments to the test sequence may be done. However, for simplicity, we did not adjust the patterns in the test sequence for our experiments and the results show that such difference is negligible.

Also, the F-scan-paths may be tested separately to ensure that the scan operation is without faults. This can be done by simply testing the CUT without going into normal mode.

4.1 An Example

A sample case to generate the test environment and test sequence is shown in Fig. 11. The original circuit, E (without augmentation), is shown in Fig. 11 (a). The state register and the control values to the ADNs are not shown, and so these are not included in the examples test environment. Figure 11 (b) shows the F-scannable circuit E. The F-paths are indicated with thicker lines and the mask is presented as a $C0$ element. The test environment is given by Fig. 11 (c), wherein it shows that a complete F-scan cycle for this circuit is equal to five clock cycles, $t0$ to $t4$. Fig. 11 (d) gives the test environment and the resulting test sequence given the test patterns $TP1$ and $TP2$ and test responses $TR1$ and $TR2$. Shown in the test sequence, the first F-scan-in occupies $t0$ and $t1$. Test phase happens in $t2$. From $t3$ to $t4$, F-scan-in and F-scan-out are overlapped. The same goes on until all the test patterns generated by ATPG are embedded to complete the test sequence.

5. Experimental Results

We applied the proposed method to 20 ITC'99 Benchmark Circuits. No experiments are done on b16 and b19 because the ADD representation of these circuits cannot be produced. In [18], we have shown that our method is superior to other scan-based techniques, such as orthogonal scan, without the consideration of the controller part. Thus, for our experiments, we did not compare with other scan-based techniques because such methods have different approaches for the controller and data path parts. We only compare with full scan because this technique, similar with F-scan, can be applied uniformly to the entire functional RTL circuit.

Table 1 presents the area overhead comparison of F-scan against gate-level full scan. For each benchmark, the synthesis was done using DesignCompiler of Synopsys. Column 1 contains the benchmark circuit names. Columns 2 and 3 show the number of flip-flops and the number of PIs and POs for each benchmark, respectively. Column 4 corresponds to the original area of the circuit. Columns 5 and 6 present the number of augmented pins and the resulting area overhead of gate-level full scan design, respectively. Similarly, such results for F-Scan are given in Columns 7 and 8. From these results, we can observe that for all benchmark circuits, F-scan has significantly lesser area overhead compared with gate-level full scan. For smaller circuits like b01 and b02, the area overhead of F-scan can be equal to that of full scan, but not greater. Moreover, for the biggest benchmark b18, the area overhead of full scan is 15.16% of the size of the circuit while for F-scan, the overhead is only 2.69%. This means that our proposed method is most effective for high-density circuits. Also, our results show that situations when there's an increased area overhead due to additional data flow caused by F-scan do not occur.

Next, we present the ATPG results of F-scan and gate-level full scan in Table 2. Here, we show the fault coverages, test application time, and CPU time (ATPG) for both cases. For all benchmarks, we generated the test patterns using TetraMax of Synopsys using combinational ATPG on synthesized circuits. Moreover, for F-scan, we performed fault simulation in order to obtain the true fault coverage of the test patterns applied on the ADD-level. This is because the synthesized gate-level circuit, where ATPG is done, may be different from the ADD circuit, where the test is applied.

In Table 2, the first column contains the ITC'99 Benchmark Circuits. Columns 2 and 6 show the ATPG fault efficiency and fault coverage for full scan and F-scan, respectively. Column 7 provides the real fault coverage for F-scan through fault simulation. Columns 3, 4, and 5 show the number of test patterns, test application time, and ATPG CPU time (in seconds) for full scan, respectively. The same information for F-scan are given in Columns 8, 10, and 11. Column 9 gives the F-scan length, which is the number of clock cycles per complete scan. The test application time of F-scan is based on this and the number of test patterns. Column 12 shows the CPU time for fault simulation. From these results, we can observe that F-scan is able to achieve high fault efficiency and fault coverage for all of the benchmark circuits. We can also observe from the results of fault simulation that the test patterns produced by doing ATPG on synthesized F-scannable circuit achieve high fault coverage even when applied at the ADD-level circuit. For most circuits, F-scan has better fault coverage compared to full scan. This is because upon augmentation using F-scan, the structure of the F-scannable circuit becomes different from the original circuit used for full scan. The additional circuitry improves the accessibility of most parts of the circuit, thus some faults that are not detectable by full scan are made detectable by F-scan. Furthermore, the increase in the number of faults of F-scannable circuits is due to the augmented cir-

Table 1 Area overhead results.

Ckts	FFs	PI/PO	Orig. Area (Units)	Full Scan		F-Scan	
				+P	AOH (%)	+P	AOH (%)
b01	5	2/2	86	1	23.26	1	23.26
b02	4	1/1	69	1	23.19	1	23.19
b03	30	4/4	360	1	33.33	1	7.22
b04	66	11/8	1014	1	26.04	1	2.66
b05	34	1/36	933	1	14.58	2+9	11.68
b06	9	2/6	135	1	26.67	1	21.48
b07	49	1/8	687	1	28.53	2+8	9.32
b08	21	9/4	299	1	28.09	1	16.05
b09	28	1/1	337	1	33.23	2	9.50
b10	17	11/6	291	1	23.37	1	18.90
b11	31	7/6	697	1	17.79	1+1	9.04
b12	121	5/6	2005	1	24.24	2+5	−52.27
b13	53	10/10	680	1	31.18	2	12.35
b14	245	32/54	11150	1	8.79	2+1	5.76
b15	449	36/70	8493	1	21.15	2+5	9.15
b17	1415	37/97	26336	1	21.49	2+5	3.69
b18	3320	36/23	87508	1	15.16	2+6	2.69
b20	490	32/22	23459	1	8.36	2+1	4.70
b21	490	32/22	23065	1	8.50	2+1	5.86
b22	735	32/22	34856	1	8.43	2+1	0.99

Table 2 Automatic test pattern generation results.

Ckts	Gate-Level Full Scan				F-Scan						
	ATPG Fault Eff. (Fault Cov.)	No. of TP	TAT	ATPG CPU Time (s)	ATPG Fault Eff. (Fault Cov.)	Fault Sim Coverage	No. of TP	F-Scan Length	TAT	ATPG CPU Time (s)	Fault Sim Time (s)
b03	100% (96.87%)	47	1487	0	100% (100%)	100.00%	52	8	476	0	0
b04	100% (91.48%)	91	6163	0.91	100% (92.69%)	92.69%	107	10	1187	0.69	0.01
b05	100% (98.04%)	128	4514	0.01	100 % (98.46%)	98.43%	131	6	923	0.02	0
b06	100% (100%)	26	269	0	100% (100%)	100.00%	34	4	174	0	0
b07	100% (94.63%)	89	4499	0.01	100% (98.24%)	98.15%	88	5	533	0.01	0
b08	100% (98.76%)	62	1385	0	100% (100%)	100.00%	60	4	304	0.01	0
b09	100% (100%)	35	1043	0	100% (99.83%)	99.83%	57	28	1681	0	0
b10	100% (100%)	65	1187	0	100% (99.84%)	99.84%	71	3	287	0	0
b11	100% (100%)	116	3743	0.01	100% (100%)	100.00%	124	4	624	0.01	0.01
b12	100% (99.97%)	251	30743	0.02	100% (99.74%)	99.73%	166	44	7514	0.02	0
b13	100% (99.40%)	73	3995	0.01	100% (100%)	100.00%	79	20	1679	0.01	0
b14	100% (99.45%)	954	234929	0.8	100% (99.89%)	99.62%	985	9	9859	6.92	0.14
b15	100% (99.28%)	830	373949	51.23	100% (99.95%)	99.94%	910	8	8198	169.06	0.06
b17	100% (99.06%)	2132	3020327	62.39	100% (99.93%)	99.92%	2237	24	55949	115.83	0.34
b18	100% (99.12%)	5363	17813843	260.65	100% (99.78%)	99.63%	5622	48	275526	419.79	3.39
b20	100% (99.46%)	1747	858267	238.87	100% (99.69%)	99.58%	1799	18	34199	288.78	0.61
b21	100% (99.46%)	1728	848938	245.09	100% (99.73%)	99.58%	1756	18	33382	297.17	0.63
b22	100% (99.45%)	2414	1777439	253.74	100% (99.70%)	99.57%	2467	27	69103	308.13	0.7

cuitry. All faults in the augmented part are made detectable in F-scan as confirmed in the experiments, thus increasing the fault coverage of F-scan compared to that of full scan. On the other hand, even if the fault efficiency achieved by ATPG is 100% for F-scan, this cannot be always achieved after fault simulation by the current ATPG approach and further improvements in the future are necessary. There is a need for a new ATPG method that guarantees 100% fault efficiency.

Test application time is also significantly reduced for F-scan. We assumed only one scan chain for full scan for simplicity. It is possible to create multiple scan chains for full scan and in that case the test application time will be comparable to F-scan's.

6. Conclusion

A new approach to functional RTL scan called F-scan has been presented in this paper. It maximally utilizes available functional elements and paths in the circuit to insert scan paths for testing. The proposed method reduces area overhead due to test compared to full scan design, as shown by the experimental results. Test application time is also superior against full scan.

Acknowledgments

This work was supported in part by Japan Society for Promotion of Science (JSPS) under Grants-in-Aid for Scientific Research (B) (No. 20300018). The authors thank Prof. Michiko Inoue, Prof. Tomokazu Yoneda, and members of the Computer Design and Test Laboratory in Nara Institute of Science and Technology for their valuable comments and suggestions during discussions regarding our work.

References

[1] H. Fujiwara, Logic Testing and Design for Testability, MIT Press, Cambridge, MA, 1985.
[2] C.-C. Lin, M.M. Sadowska, M.T-C. Lee, and K-C. Chen, "Cost-free scan: A low-overhead scan path design," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.17, no.9, pp.852–861, Sept. 1998.
[3] R. Gupta and M.A. Breuer, "Partial scan design of register transfer level circuits," JETTA, vol.7, pp.25–46, 1995.
[4] S. Bhattacharya and S. Dey, "H-Scan: A high level alternative to full-scan testing with reduced area and test application overheads," Proc. VLSI Test Symposium, pp.74–80, April 1996.
[5] R.B. Norwood and E.J. McCluskey, "Orthogonal scan: Low overhead scan for data paths," Proc. Int. Test Conf., pp.659–668, 1996.
[6] T. Asaka, S. Bhattacharya, S. Dey, and M. Yoshida, "H-Scan+: A practical low-overhead RTL design-for-testability technique for industrial designs," Proc. Int. Test Conf., pp.265–274, 1997.
[7] Y. Huang, C.C. Tsai, N. Mukhejee, O. Samman, D. Devries, W.T. Cheng, and S.M. Reddy, "Synthesis of scan chains for netlist descriptions at RT-level," JETTA, vol.18, pp.189–201, 2002.
[8] C.Y. Ooi and H. Fujiwara, "A new scan design technique based on pre-synthesis thru functions," Proc. 15th IEEE Asian Test Symposium, pp.163–168, 2006.
[9] I. Ghosh, A. Raghunathan, 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.
[10] K. Takabatake, M. Inoue, T. Masuzawa, and H. Fujiwara, "Non-scan design for testable data paths using thru operation," Proc. Asia and South Pacific Des. Autom. Conf., pp.313–318, 1997.
[11] I. Ghosh, A. Raghunathan, N.K. Jha, "A design-for-testability technique for register-transfer level circuits using control/data flow extraction," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.17, no.8, pp.706–723, 1998.
[12] 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. VLSI Des., pp.300–305, 2000.
[13] 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 South Pacific Des. Autom. Conf., pp.599–604, 2000.

[14] 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 Des. Autom. Conf., pp.331–334, Feb. 2001.
[15] H. Fujiwara, H. Iwata, T. Yoneda, and C.Y. Ooi, "A non-scan design-for-testability for register-transfer level circuits to guarantee linear-depth time expansion models," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.27, no.9, pp.1535–1544, 2008.
[16] V. Chaiyakul, D.D. Gajski, and L. Ramachandran, "High-level transformations for minimizing syntatic variances," Proc. Des. Autom. Conf., pp.413–418, 1993.
[17] I. Ghosh and M. Fujita, "Automatic test pattern generation for functional register-transfer level circuits using assignment decision diagrams," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.20, no.3, pp.402–415, 2001.
[18] M.E.J. Obien and H. Fujiwara, "F-Scan: An approach to functional RTL scan for assignment decision diagrams," Proc. IEEE 8th Int. Conf. on ASIC, pp.589–562, Oct. 2009.
[19] M.E.J. Obien and H. Fujiwara, "A DFT method for functional scan at RTL," Proc. 10th IEEE Workshop on RTL and High Level Testing, pp.6–15, Nov. 2009.

Marie Engelene J. Obien received her B.S. degree in electronics and communications engineering from Ateneo de Manila University, Philippines in 2005. She also received her M.S. degree in electronics engineering from the same university in 2008. At present, she is pursuing a doctorate degree in Nara Institute of Science and Technology. Her research interests include high level testing, design for testability, and design for security. She is also a student member of IEEE.

Satoshi Ohtake received the B.E. degree in computer science from the University of Electro-Communication, Tokyo, Japan, in 1995 and the M.E. and Ph.D. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 1997 and 1999, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science from 1998 to 1999. Presently he is an Assistant Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. His research interests are VLSI CAD, design for testability, and test pattern generation. Dr. Ohtake is a member of IEEE and IPSJ.

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, Nara, Japan in 1993. Presently he is a Professor with 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 has published over 350 papers in refereed journals and conferences, and nine books including the book from the MIT Press (1985) entitled Logic Testing and Design for Testability. He received many awards including the Okawa Prize for Publication, three IEEE CS (Computer Society) Certificate of Appreciation Awards, two IEEE CS Meritorious Service Awards, IEEE CS Continuing Service Award, and two IEEE CS Outstanding Contribution Awards. He has served as an editor and associate editors of several journals, including the IEEE Trans. on Computers, and Journal of Electronic Testing: Theory and Application, and as guest editor of several special issues of IEICE Transactions of Information and Systems. Dr. Fujiwara is a fellow of the IEEE, a Golden Core member of the IEEE Computer Society, and a fellow of the IPSJ.