Engineering Science and Technology, an International Journal 19 (2016) 1473–1487

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

A Discrete Event System approach to On-line Testing of digital circuits with measurement limitation

P.K. Biswal, H.P. Sambho, S. Biswas *

Department of Computer Science and Engineering, IIT Guwahati, India

ARTICLE INFO

Article history:
Received 6 January 2016
Accepted 21 April 2016
Available online 11 May 2016

Keywords:
On-line Testing
Measurement limitation
Fault coverage
Area overhead
Fault detection latency
Discrete event systems

ABSTRACT

In the present era of complex systems like avionics, industrial processes, electronic circuits, etc., on-the-fly or on-line fault detection is becoming necessary to provide uninterrupted services. Measurement limitation based fault detection schemes are applied to a wide range of systems because sensors cannot be deployed in all the locations from which measurements are required. This paper focuses towards On-Line Testing (OLT) of faults in digital electronic circuits under measurement limitation using the theory of discrete event systems. Most of the techniques presented in the literature on OLT of digital circuits have emphasized on keeping the scheme non-intrusive, low area overhead, high fault coverage, low detection latency etc. However, minimizing tap points (i.e., measurement limitation) of the circuit under test (CUT) by the on-line tester was not considered. Minimizing tap points reduces load on the CUT and this reduces the area overhead of the tester. However, reduction in tap points compromises fault coverage and detection latency. This work studies the effect of minimizing tap points on fault coverage, detection latency and area overhead. Results on ISCAS89 benchmark circuits illustrate that measurement limitation have minimal impact on fault coverage and detection latency but reduces the area overhead of the tester. Further, it was also found that for a given detection latency and fault coverage, area overhead of the proposed scheme is lower compared to other similar schemes reported in the literature.

1. Introduction

With the advancement of technology and larger scales of production, intelligent automation systems are increasingly making their presence felt in all aspects of engineering such as avionics, industrial processes, manufacturing systems, transportation systems, electronic systems, etc.[1,2]. Due to the increase in complexity of these systems, there is also a rise in the number of faults occurring in them. Such systems are required to be always available when needed and maintained on the basis of their current condition monitoring, rather than scheduled or breakdown maintenance [3,4]. In other words, the classical philosophy of performing burn-in tests after production and deploying the system with the assumption of fault free behavior thereafter may not be valid. So, On-line Testing (OLT) i.e., on-line fault detection is becoming an indispensable part of testing [5–8].

Several approaches to on-line fault detection have been reported in the literature and can be broadly classified as fault-tree based analysis [9], expert system based methods [10], machine learning techniques [11,12] and model based methods [13–15]. Any kind of automated reasoning, ranging from fault detection to stability analysis of complex systems, can be achieved efficiently through model based representations. In a model based approach, a detailed process model is constructed first. The system states are estimated from this model and the corresponding fault condition is determined based on the values of the measurable system parameters. The commonly used model based techniques are analytical redundancy based methods [16], Discrete Event System (DES) model based methods [13–15], Hybrid System (HS) model based methods [17,18], etc. DES model based methods are used for fault detection for a wide range of applications because of simplicity of both the model and the associated algorithms. Further, most systems, even with continuous dynamics, can be viewed as DESs at some level of abstraction. A DES is characterized by a discrete state space and some event driven dynamics. The core idea is to develop the normal and faulty DES models corresponding to normal and abnormal scenarios of the system. Subsequently, a detector which is a state estimator is built that determines whether the system is operating under normal, fault or uncertain conditions.

* Corresponding author.
E-mail address: santoshbiswas402@yahoo.com (S. Biswas).
Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2016.04.005
2215-0986/© 2016 Karabuk University. Publishing services by Elsevier B.V.

One of the major infrastructure involved in DES based fault detection is sensors that measure the system parameters continuously, which in turn are used by the detector [19–21]. The more the number of parameters that can be measured by the sensors and given as input to the detector, the higher is the number of faults that can be detected. However, many factors like cost, physical conditions involving extremely high temperature, etc. limit the feasibility of deploying sensors in all the required points in the system. So, instead of measuring all parameters, a subset of parameters are measured and fault detection is performed by the detector only based on such measurements [20,21]. Thus, fault detection under limitation on measuring system parameters i.e., measurement limitation is an important area of research [20,21].

In VLSI circuits, fault detection is one of the major challenges to achieve acceptable quality of service [22]. Moore had predicted in his seminal paper [23] that transistor density of VLSI circuits would double every 18 months. The prediction has proved to be correct and at present we have reached the Deep Submicron (DSM) era, where single chip implementation of complex systems, network of processor cores, etc. have been successfully fabricated. DSM VLSI circuits involve millions of transistors on a single chip and the operation speed is at the level of GHz.; such high complexity of fabrication also increases the possibility of faults in the fabricated chips [22,24]. Traditional testing schemes like Automatic Test Equipment (ATE) based testing, Built in Self Test (BIST), etc. cannot detect many of these faults. ATE based testing, also called manufacturing test involves connecting the fabricated chip to the tester, applying test patterns and comparing with golden response [25,22]. In BIST, every time before powering up the circuit it is tested using on-chip pattern generator and response analyzer [26]. However, in DSM circuits probability of development of faults during operation is also being observed frequently [27–29]. To elaborate, such faults were not present during manufacturing or during powering up of the circuit, but developed on-the fly during its operation. So classical manufacturing test and BIST cannot detect faults that occur during operation of the circuit. To detect these faults that develop on the fly, a new test methodology called On-Line Testing (OLT) is required. *OLT of circuits can be defined as the procedure to enable integrated circuits to verify the correctness of their functionality during normal operation by checking whether the response of the circuit conforms to its desired dynamic behavior.* OLT for digital circuits is being studied for the last two decades and can be broadly classified into the following main categories, (i) Signature monitoring in Finite State Machines (FSMs) [30,31], (ii) Self-checking design[28,32,33], (iii) Partial replication [34–36]. (iv)On-line Built-in-Self-Test (BIST) [37–39]. OLT techniques have emphasized on keeping the schemes as non-intrusive as possible (i.e., minimal change to the original circuit), totally self-checking, low area overhead, high fault coverage (mainly single stuck at faults), low detection latency etc. However, in the DSM era, several other factors need to considered for OLT namely, coverage for advanced fault models (e.g., bridging faults, delay faults etc.), scalability, flexibility of area overhead of the tester versus fault coverage and detection latency etc. Some important contributions in this area are [27,40,35,36].

The on-line tester circuitry executes concurrently with the CUT and needs to tap certain lines of the CUT. These tap points can be considered as sensors for the tester. As the on-line tester is fabricated on the same chip with the CUT, any point of the CUT can be tapped. This enables the measurement of any required digital parameter of the CUT by the tester. So, most the above mentioned OLT techniques have ignored the issue of tap points or measurement limitation. However, tapping of lines of any circuit results in increase of load (fanouts) on the gates which drive the tap points [41]. To handle the increased load extra buffers are required, which increase the area of the circuit. So if the on-line tester is designed with high number of tapings in the CUT, it results in huge area overhead. So, in this work we aim at design of on-line testers for digital circuits, targeting minimization of tap points using the concept of DES based fault detection with measurement limitation. However, reduction in tap points also compromises fault coverage and detection latency. Therefore, "number of tap points" can be used as parameter to tradeoff area overhead versus fault coverage and detection latency.

In this paper, we propose a DES based scheme for design of online testers of digital circuits, targeting minimization of the number of tap points subject to fault coverage and detection latency. Most of the operations for construction of the on-line tester are based on Ordered Binary Decision Diagrams (OBDDs) [42]. OBDDs represent Boolean functions in a compressed form. Further in OBDDs, operations can be performed directly on the compressed representation, i.e., without decompression. So complexity of the on-line tester design is greatly improved by using OBDDs compared to traditional data structures like Binary tree, FSM etc. The time complexity of designing the on-line tester is also discussed in this paper. Experimental results on ISCAS89 benchmarks have been presented, which illustrate that measurement limitation can be used as a tradeoff parameter to minimize area overhead to a great extent, with minimal compromise in detection latency and coverage. It was also found that for a given detection latency and coverage, area overhead of the proposed scheme is lower compared to other similar schemes reported in the literature.

The paper is organized as follows. In Section 2 we start with literature review on OLT followed by motivations and contributions of the present work. Section 3 presents state based DES modeling and *FN*-detector construction under measurement limitation. Section 4 illustrates use of OBDDs to generate *FN*-detector efficiently. The complexity of construction of the *FN*-detector is also discussed in this section. Section 5 presents experiential results regarding area overhead, fault coverage and detection latency versus measurement limitation (i.e., tap point reduction) of the *FN*-detector. Finally we conclude in Section 6.

2. Literature review and motivation of the work

In this section we briefly discuss the major contributions related to OLT of digital circuits and then build the motivation of the present work. As already mentioned, OLT techniques for digital circuits can be broadly classified into (i) signature monitoring in FSMs, (ii) self-checking design, (iii) partial replication and (iv) on-line BIST.

Signature monitoring in FSMs: Signature monitoring techniques for OLT (of sequential circuits) work by modeling a circuit as FSM and studying the state sequences during its operation. The basic assumption is signature invariant property i.e., signature of a circuit, obtained based on state sequence, is different under normal and faulty conditions. The feasibility of signature invariance is proved in [43] and several authors have proposed implementations based on this approach [44,30,31]. The hybrid signature monitoring scheme reported in [30] detects control flow errors caused by transient and intermittent faults. It is shown that the scheme has offered very high fault coverage with low detection latency and area overhead. In [31], a concurrent control flow error detection and recovery mechanism has been proposed using encoded signature monitoring technique. The scheme recovers from most of the control flow errors with relatively low performance overhead.

However, many times by default, signature invariance is not present. In that case, redundant states need to be inserted and state encoding is modified. So these schemes require re-synthesis and re-design, which lead to a change in the original structure of the circuit; they are accordingly termed as "intrusive OLT methodologies". Further, the state explosion problem in FSMs makes the application of these techniques difficult for practical circuits.

Self Checking Design: The technique of OLT using *self Checking Design* based on error detecting codes was motivated from the error detection and correction techniques used in communication. The circuit output bits are augmented with some additional bits, called check bits (using an additional logic), such that only under normal condition the output augmented with the check bits is a code word of the error detecting code chosen. Some error detecting code based OLT schemes are Parity codes, Berger codes, m-out-of-n codes [28], etc. The on-line tester (checker) verifies whether the circuit output augmented with the check bits is a code word of the error detecting code chosen. As a practical application, the residue code technique is used in [32] for design of self checking modulo multiplier, which is used in various cryptographic systems. In another work [33], D. P. Vasudevan and P. K. Lala have proposed a new approach for designing of self checking carry select adder of arbitrary size and the adder can detect both permanent and transient stuck at faults online.

The area overhead for making circuits self checkable is usually not high. However, this technique is also intrusive because some special structure is required in the circuit to limit the scope of fault propagation, which guarantees the self checking property. These structures can be achieved by re-synthesis and re-design of the original circuit and they may affect the critical paths in the circuit. The above schemes mainly targeted stuck at faults, however, in DSM era coverage for advanced fault models (e.g., bridging faults, delay faults etc.) are required. OLT for Bridging Faults (BFs) have been attempted in [45,27] using checkers based on coding theory.

Partial replication: One of the most simple way of OLT is duplicating the circuit itself and cross checking for similarity of output responses. Even if the scheme is non-intrusive, the area overhead is more than 100%. An interesting alternative to full hardware duplication is presented by Drineas et al. in [34]. The scheme is based on *partial replication*, where the replicated module is replaced by a minimized version of the CUT. In the first step, a complete set of test vectors for all the stuck at faults possible are generated using Automatic Test Pattern Generation (ATPG) algorithms. Following that, a subset of faults are selected and a subset of test vectors for the selected faults are taken and synthesized into a circuit that is used for OLT. It may be noted that ATPG algorithms are optimized to generate the minimum number of test vectors that detect all faults. As the scheme applies ATPG algorithms in a reverse philosophy, so it becomes prohibitively complex for large circuits. In [35,36,46], Biswas et al. have developed partial replication based OLT schemes using fault detection theory of DES. The on-line testers designed using the DES based scheme are non-intrusive, scalable, provide coverage for advanced fault models (bridging, delay etc.), facilitate tradeoffs in area/power overheads versus coverage/detection latency, etc.

On-line BIST: The technique of designing circuits with additional on-chip logic, which can be used to test the circuit before it starts, is called off-line BIST. Off-line BIST resources are now being used for On-line Testing [37–39]. This technique utilizes the idle time of the various parts of the circuit during operation to perform on-line BIST. Thus, this is the only technique that provides both on-line and off-line test facilities utilizing the same on-chip hardware resources. There are three issues that are related to the performance of on-line BIST namely, availability of idle time, minimizing test length so as to fit within the idle time available and test schedule. So the feasibility and efficiency depend on the amount of idle time available. Therefore, this technique cannot be effective for many circuits because for design efficiency the utilization of circuit modules are kept high (there by having low idle times).

The on-line tester circuit is placed on-chip with the CUT. The tester taps certain lines of the CUT, whose values are used to determine whether any fault has developed in the CUT. Such tap points are analogous to sensors used in physical systems. Unlike physical systems, where sensors cannot be placed at all desired locations, in case of circuits the on-line tester can tap any point of the CUT. So, all the above mentioned OLT techniques have emphasized on keeping the scheme as non-intrusive (i.e., minimal changes to the original circuit) as possible, totally self-checking, low area overhead, high fault coverage, low detection latency, etc. but ignored the issue of tap points. However, tapping of lines of a CUT results in increase of load (fanouts) on the gates which drive the tap points [41]. Such increase in load requires use of additional buffers that increase area overhead [41]. So, if the concept of fault detection under measurement (i.e., sensor placement limitation in case of physical system) is applied for OLT of circuits, we can minimize the tap points of the CUT and reduce the number of driving buffers. This will minimize the area overhead of the tester. However, minimization in tap points also compromises fault coverage and detection latency. So, "number of tap point" can be considered as a new design parameter to provide trade-offs between area overhead versus fault coverage and detection latency.

In this paper, we propose a DES based approach for OLT of digital circuits with measurement limitation. Specifically, the contributions of this paper vis-a-vis the motivations are as follows:

- *DES based design technique for On-line tester of digital circuits with measurement limitation.* A novel measurement limitation DES based technique for design of on-line testers of digital circuits, targeting minimization of tap points has been proposed. All the previous works on on-line tester design ignored the issue of measurement limitation because the tester is fabricated on the same chip with the CUT and any point of the CUT can be tapped (measured) easily. However, we have shown that higher the number of lines of the CUT that are measured by the tester, higher is the area overhead. So the proposed scheme applies measurement limited DES to reduce the tap points and minimize the area overhead. However, reduction in tap points also compromises fault coverage and detection latency. Therefore, in the proposed scheme "number of tap points" is provided as a parameter to tradeoff area overhead versus fault coverage and detection latency.
- *Scalable technique for on-line tester design using OBDD based operations.* The steps involved in the construction of the on-line tester which are computationally complex (namely, (i) determining the exhaustive set of fault detecting transitions, (ii) determining the fault detecting transitions that retain their capability to detect faults under the given measurement limitation) are implemented using Ordered Binary Decision Diagrams (OBDDs). OBDDs represent Boolean functions in a compressed form and operations can be performed directly on the compressed representation. So complexity of the on-line tester design is greatly improved by using OBDDs compared to traditional data structures like Binary tree, FSM etc. The time complexity of designing the on-line tester is also formally analyzed.
- *Illustration of the fact that "lowering tap points by on-line tester facilitates lower overhead at slight compromise in latency and fault coverage*". The proposed on-line tester has been validated on ISCAS89 benchmarks. The experimental results illustrated that measurement limitation can be used as a tradeoff parameter to the minimize area overhead to a great extent, with minimal compromise in detection latency and fault coverage. To the best of our knowledge, such observations regarding measurement limitation, area overhead, detection latency and fault coverage for on-line testers in digital circuits have been reported for the first time in the literature. Also, the results revealed that

given a detection latency and coverage, area overhead of the proposed scheme is lower compared to other similar schemes reported in the literature.

3. DES framework under measurement limitation: Circuit modeling and *FN*-detector design

In this section, we model a digital sequential circuit having a single clock using DES modeling framework.[1] Fig. 1 illustrates the basic block diagram of a sequential circuit with on-line tester. In this paper we consider only the Next State Function (NSF) block and flip-flops for OLT. The mechanism can be easily extended for the output function block, which is a combinational circuit. The NSF block and the flip-flops are extracted from the circuit and would be considered for OLT (similar to the case in [35]). In other words, two sub-parts of the circuit i.e., NSF block and the flip-flops are considered as the CUT.

A sequential circuit without the OF block is modeled as a DES $G = \langle V, X, X_0, \Sigma, \Im \rangle$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of Boolean variables, X is a finite set of states, $X_0 \subseteq X$ is the set of initial states, Σ is a finite set of input symbols and $\Im$ is a finite set of transitions. The set V of variables can be partitioned into two subsets, namely (i) $S = \{v_1, v_2, \ldots, v_k\}$ representing the state variables and (ii) $I = \{v_{k+1}, v_{k+2}, \ldots, v_n\}$ representing the input variables. A state $x \in X$ is a mapping $x : S \rightarrow \{0, 1\}$. Similarly, any input symbol $\sigma \in \Sigma$ is a mapping $\sigma : I \rightarrow \{0, 1\}$. Thus, a state is represented by a binary k-tuple, where $k = \lceil \log_2 |X| \rceil$. Similarly, any input symbol can be represented as a binary i-tuple, where $i = n - k$ and $|\Sigma| = 2^{n-k}$.

All input and state variables are not measurable. Let $I_m \subset I$ and $S_m \subset S$ be the subsets of measurable input and state variables, respectively. The unmeasurable state (input) variables correspond to register-outputs (primary input lines) which are not tapped by the on-line tester. The measurable input alphabet $\Sigma_m = \{\sigma|_{I_m}$ such that $\sigma \in \Sigma\}$ and the measurable set of states $X_m = \{x|_{S_m}$ such that $x \in X\}$.[2]

A *transition* $\tau \in \Im$ from a state x_τ to another state x^+_τ is an ordered three-tuple $\tau = \langle x_\tau, \sigma_\tau, x^+_\tau \rangle$ where, x_τ (*initial*(τ)) is the initial state of the transition, x^+_τ (*final*(τ)) is the final state of the transition and $\sigma_\tau \in \Sigma$ (*input*(τ)) is the input symbol of the transition.

3.1. Circuit modelling under single stuck-at (s-a) fault

Single s-a faults are represented in an DES model of a circuit using the following steps:

- The variable set V is extended to include a subset C of $k = \lceil lg(p + 1) \rceil$ *status variables*. Thus, $V = S \cup I \cup C$, where S and I are the sets of state and input variables respectively, as given before.
- The state mappings are extended so that each becomes a mapping form $S \cup C$ to $\{0, 1\}$. $\bigcup_{x \in X} x(C) = \{N, F_1, F_2, \ldots, F_p\}$, where N stands for normal status and $F_i, 1 \leqslant i \leqslant p$, stands for the i^{th} fault status. The image $x(C)$ of C under x is called the fault label of the state x.

The set of status variables are unmeasurable. The DES model of a circuit (capturing both normal and fault status) can be conveniently conceived as a collection of submachines, one for the normal condition and one each for the faults $F_1, F_2, \ldots, F_p$. The onset of a fault F_i is captured by a transition from a state x with $x(C) = N$ to a state x' with $x'(C) = F_i$; such transitions are termed as s_i-transitions (for the "start" of fault F_i) and are represented as $s_i = \langle x_1, T, x_2 \rangle$, where $x_1(C) = N, x_2(C) = F_i$ and T stands for "true". Due to the occurrence of an s_i-transition, only the status variables change their values from N to F_i. All other state variables remain unchanged since the state register changes only at the triggering edges of the clock depending upon the inputs. Thus, s_i-transitions are unmeasurable. These transitions are asynchronous, i.e., they need not occur at the triggering edges of the clock. Their enabling conditions do not depend on any input variable combinations as they are always true.

Under one notation, we use a flat indexing like $x_1, x_2, \ldots, x_l$ when no distinction needs to be made among the submachine boundaries (normal or faulty). According to the second notation, the states for the normal submachine are designated as $x_{0j}, 1 \leqslant j$, and those of the F_i-submachine are designated as $x_{ij}, 1 \leqslant j$; likewise for the transitions.

Definition 1 (*N-state). A G-state is called normal (i.e., N-state), denoted as* $x_{0j}, 1 \leqslant j$, *if* $x_{0j}(C) = N$. *The set of all normal states is denoted as* X_N.

Definition 2. F_i-state A G-state is called an F_i-state, denoted as $x_{ij}, 1 \leqslant j$, if $x_{ij}(C) = F_i$. The set of all F_i states is denoted as X_{F_i}.

In the sequel, a G-transition $\tau_{0j} = \langle x_{\tau_{0j}}, \sigma_{\tau_{0j}}, x^+_{\tau_{0j}} \rangle$ is called a *normal G-transition* if $x_{\tau_{0j}}, x^+_{\tau_{0j}} \in X_N$. Similarly, a G-transition $\tau_{ij} = \langle x_{\tau_{ij}}, \sigma_{\tau_{ij}}, x^+_{\tau_{ij}} \rangle$ is called a *Fault F_i G-transition* if $x_{\tau_{ij}}, x^+_{\tau_{ij}} \in X_{F_i}$. Since faults are assumed to be *permanent*, there is no transition from any X_N state to any X_{F_i} state.

Definition 3 (*Measurement equivalent states).* Two states x_1 and x_2 are measurement equivalent, denoted as $x_1 E x_2$, if $x_1|_{S_m} = x_2|_{S_m}$.

Definition 4 (*Measurement equivalent transitions).* Two transitions $\tau_1 = \langle x_{\tau_1}, \sigma_{\tau_1}, x^+_{\tau_1} \rangle$ and $\tau_2 = \langle x_{\tau_2}, \sigma_{\tau_2}, x^+_{\tau_2} \rangle$ are measurement equivalent, denoted as $\tau_1 E \tau_2$, if $x_{\tau_1}|_{S_m} = x_{\tau_2}|_{S_m}$, $x^+_{\tau_1}|_{S_m} = x^+_{\tau_2}|_{S_m}$ and $\sigma_{\tau_1}|_{I_m} = \sigma_{\tau_2}|_{I_m}$.

In other words, two transitions are measurement equivalent if their source states are measurement equivalent, destination states are measurement equivalent and so are the inputs.

Throughout this paper the simple example circuit given in Fig. 2 is used to illustrate the theory. A single s-a-1 fault F_1 is assumed at the fanout branch marked *A*. Fig. 3 shows the DES model for the circuit corresponding to the normal and faulty behaviour.

3.2. FN-detector for DES model of a circuit

Let us first consider the circuit of Fig. 2 under the case of full measurement i.e., $S_m = S$ and $I_m = I$. Comparing the transitions under normal condition with the corresponding ones after the s-a-1 fault in the DES model given in Fig. 3, it is noted that there is one transition namely, $\langle x_{11}, 1, x_{13} \rangle : \tau_{16}$, that reflects a change in behaviour after fault because the corresponding transition in the normal condition is $\langle x_{01}, 1, x_{02} \rangle : \tau_{06}$, where $x_{01}|_{S_m} = x_{11}|_{S_m} = 10$, $\sigma_{\tau_{06}} = \sigma_{\tau_{16}} = 1$ but $x_{02}|_{S_m} = 00 \neq x_{13}|_{S_m} = 01$. All other F_1-transitions have an equivalent N-transition e.g., $\tau_{01} E \tau_{11}$; so they cannot detect the fault.

Hence, a finite state machine can be designed to detect whether the following happens: The circuit is in state x_{01} or equivalently, in x_{11} (i.e., measured state variables are 10) following which at input

[1] In this paper we consider sequential circuits only. The technique can be easily adapted for combinational circuits because "combinational circuit can be modeled equivalently as sequential circuit with a single state".

[2] $x|_{S_m}$ is the restriction of values x to set S_m. For example, if $x = \langle v_1 v_2 v_3 \rangle = \langle 1 1 0 \rangle$ and $S_m = \{v_1, v_2\}$, then $x|_{S_m}$ returns $\langle 1 1 \rangle$.

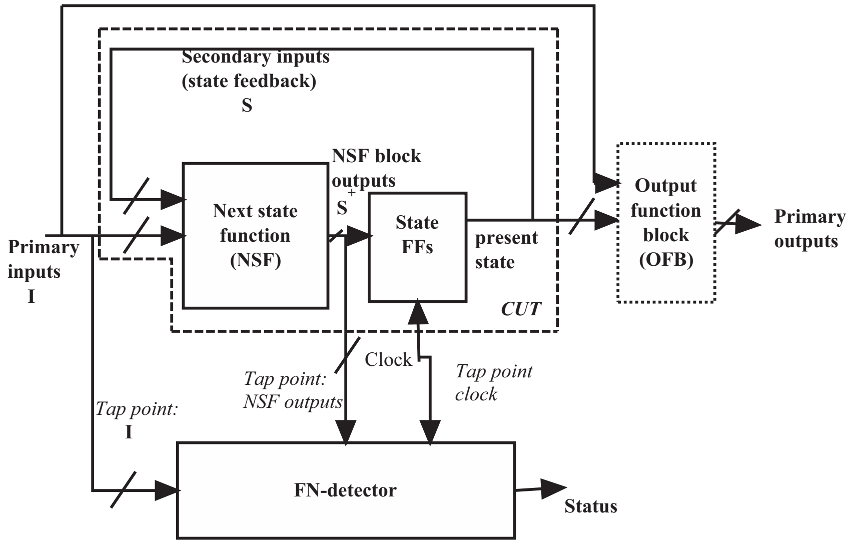

Fig. 1. Basic architecture of a sequential circuit with on-line tester (*FN*-detector).

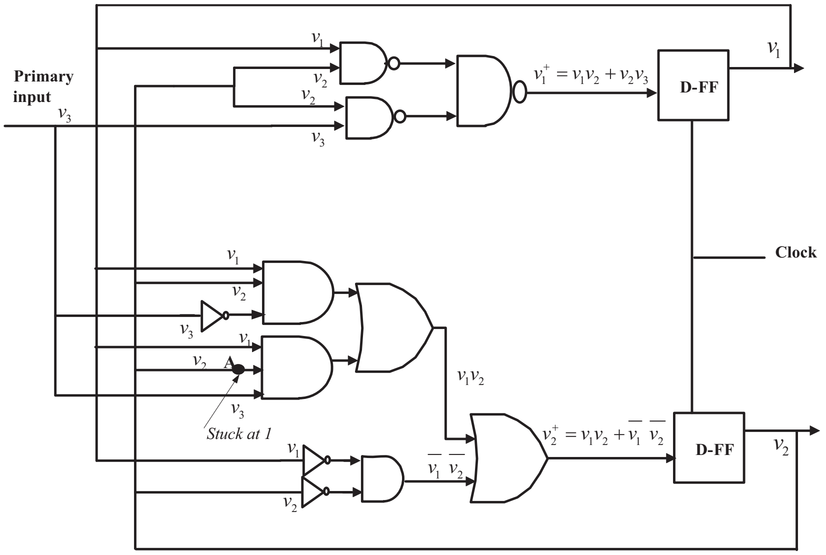

Fig. 2. A simple sequential circuit with a s-a-1 fault.

1 the next state is x_{13} (i.e., the measured state variables in the next clock period become 01); this indicates the s-a-1 fault. We refer to this machine as an *FN-detector* (fault versus normal condition detector) and the transition(s), whose occurrences are used to detect the faults, as *FD*-transitions (fault detecting transitions).

From the above observation, it appears that the *FN*-detector needs to measure not only the NSF block outputs but also the state flip-flop outputs. For example, for the CUT of Fig. 2, the *FN*-detector may monitor v_1, v_2 in addition to the NSF block outputs v_1^+, v_2^+ and the input v_3. If the input vector $\langle v_1, v_2, v_3, v_1^+, v_2^+\rangle = \langle 1\,0\,1\,0\,1\rangle$, then the *FN*-detector can move to a fault indicating state; else, on encountering $\langle 1\,0\,1\,0\,0\rangle$ (or any other *don't care* pattern), it can loop back to the same state.

In a typical VLSI circuit, however, the number of state variables can be quite high. It will, therefore, be worthwhile to contain the number of inputs to the *FN*-detector circuit. We can indeed do so by letting the *FN*-detector measure (monitor) only the NSF outputs and permitting the detection of an *FD*-transition to be a two step process needing two clock periods. In the first clock period, the *FN*-detector checks whether the CUT *is going to be* in the initial state of the *FD*-transition and if it happens, then in the next clock period, the *FN*-detector examines whether the primary input and the NSF block output match, respectively the input and the final state of the *FD*-transition. Obviously, both the steps can be accomplished by monitoring only the NSF block outputs. It is to be noted that the mechanism sill permits the *FN*-detector to proceed in step with the CUT, both being driven by the same clock edge. The basic schematic of the *FN*-detector vis-a-vis the CUT is shown in Fig. 1. The process of *FN*-detector construction from *FD*-transition is first illustrated for the CUT of Fig. 2 and then generalized.

The state transition diagram of the *FN*-detector of the circuit's DES model, shown in Fig. 3, is given in Fig. 4. This *FN*-detector is designed to detect the occurrence of the *FD*-transition $\langle x_{11}, 1, x_{13}\rangle : \tau_{16}$. The *FN*-detector starts from the initial state z_0. It reaches state z_1 by the transition labelled t_2 when the CUT moves to the state $x_{11} = initial(\tau_{16})$, that is, when the measured NSF block

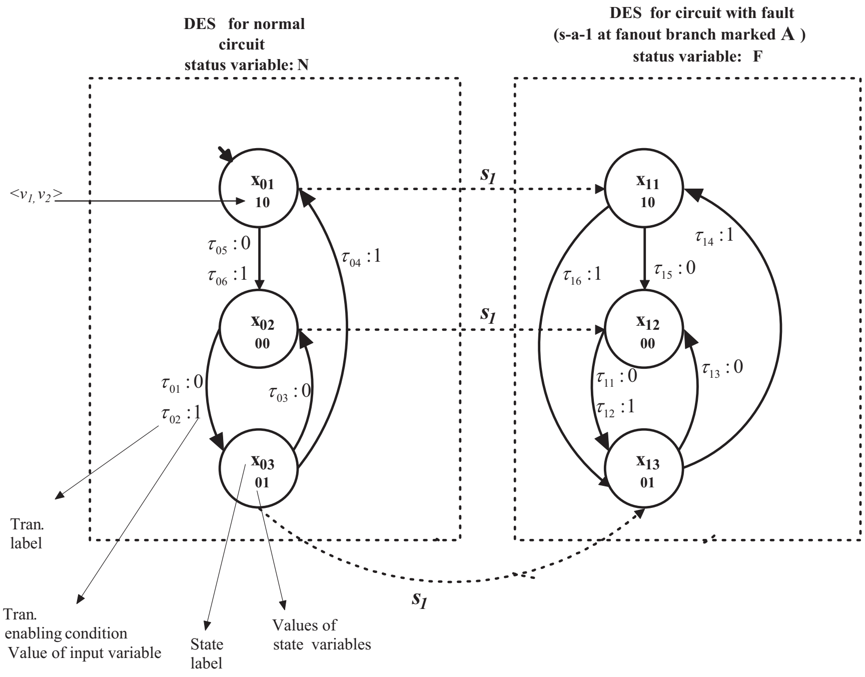

Fig. 3. A DES model for the circuit (of Fig. 2).

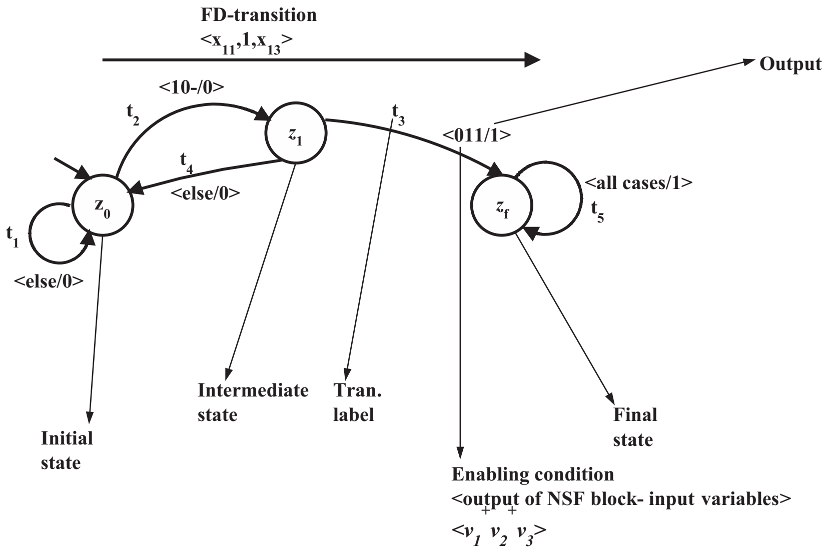

Fig. 4. The *FN*-detector for the DES model of the circuit with a s-a-1 fault (Fig. 3).

outputs v_1^+ and v_2^+ are 1 and 0, respectively. The value of the input variable v_3 is a don't care because the *FN*-detector depends only on the next state of the CUT to reach z_1. If the CUT is going to be in a state other than x_{11}, (or x_{01}), then the self-loop transition t_1 occurs. Thus, the *FN*-detector assumes the state z_1 simultaneously with the CUT assuming the state x_{11} (or x_{01}). The transition t_3 from the state z_1 corresponds to the fact that the *FD*-transition τ_{16} is going to occur in the CUT in the next clock edge because the NSF block outputs $v_1^+ = 0$ and $v_2^+ = 1$ and the input variable $v_3 = 1$, as given by the 3-tuple $\langle v_1^+ v_2^+ v_3 \rangle = \langle 0\,1\,1 \rangle$. Thus, the transition t_3 leads the *FN*-detector to the final state z_f yielding output 1 indicating that the s-a-1 fault has occurred at the fanout branch marked *A*. If such an input-next state combination is not found in the state z_1, then the *FN*-detector moves back to the initial state by the transition t_4. Once the final state z_f is reached, the *FN*-detector remains in that state forever maintaining the output as 1 since the faults are assumed to be permanent.

Now, we formally characterize the *FD*-transitions and the *FN*-detector.

Definition 5 (*FD-transition*). *An F_i-G-transition $\tau_{ik} = \langle x_{\tau_{ik}}, \sigma_{\tau_{ik}}, x^+_{\tau_{ik}} \rangle$ is an FD-transition, for fault F_i (denoted as FD_i), if there is a normal-G-transition $\tau_{0l} = \langle x_{\tau_{0l}}, \sigma_{\tau_{0l}}, x^+_{\tau_{0l}} \rangle$ such that $x_{\tau_{0l}} E x_{\tau_{ik}}, \sigma_{\tau_{0l}}|_{I_m} = \sigma_{\tau_{ik}}|_{I_m}$ and not $x^+_{\tau_{0l}} E x^+_{\tau_{ik}}$. The set of all FD_i-transitions is denoted as $\Im_{FD_i}$. The set of all FD-transitions for all faults is denoted as $\Im_{FD}$.*

Let $\Im_{FD_i} = \{\tau_{i1}, \tau_{i2}, \ldots, \tau_{il}\}$, where, $1 \leqslant j \leqslant l, \tau_{ij} = \langle x_{\tau_{ij}}, \sigma_{\tau_{ij}}, x^+_{\tau_{ij}} \rangle$. The *FN*-detector is driven by the same clock edge as the CUT. In general, there are three types of states in any *FN*-detector – an initial state z_0, a set of intermediary states $z_1, z_2, \ldots, z_l$ and a single final state z_f. The initial state z_0 keeps track of the next *G*-state by monitoring the outputs of the NSF block $v_1^+, v_2^+, \ldots, v_k^+$. The input variables $v_{k+1}, v_{k+2}, \ldots, v_n$ are don't cares for the transitions emanating from z_0. Whenever the CUT is going to be in any state $x_{\tau_{ij}}|_{S_m}$, for some $\tau_{ij} \in \Im_{FD_i}$, at the next clock edge, the *FN*-detector moves to an intermediary state z_j. Thus, there is a transition from z_0 to z_j (for τ_{ij}), labeled with the values of the measurable state variables corresponding to $x_{\tau_{ij}}|_{S_m}$; the outputs associated with all these transitions from z_0 are 0. In the state z_j, the *FN*-detector keeps track of those outputs of the NSF block $v_1^+, v_2^+, \ldots, v_k^+$ which are in S_m and inputs from $v_{k+1}, v_{k+2}, \ldots, v_n$ which are in I_m. If the input pattern matches with $\sigma_{\tau_{ij}}|_{I_m}$ and the NSF block output pattern matches with $x^+_{\tau_{ij}}|_{S_m}$, then the *FN*-detector moves to the final state z_f yielding an output 1; else it moves back to z_0. Thus, there is a transition from z_j to z_f (for τ_{ij}), labeled with the values of the measurable state variables corresponding to the state $x^+_{\tau_{ij}}$ and the measurable values of the input variables corresponding to $\sigma_{\tau_{ij}}$. The set of *FN*-detector states, therefore, is given by $Z = \{z_0, z_1, z_2, \ldots, z_l, z_f\}$, where $z_1, z_2, \ldots, z_l$ correspond to the initial states of the FD_i-transitions. In a similar way *FD*-transitions of all other faults need to be incorporated in the *FN*-detector by associating intermediary states with each transition. It is possible to merge two intermediary states z_j and z_n into a single one if the corresponding *FD*-transitions $\tau_{ij} = \langle x_{\tau_{ij}}, \sigma_{\tau_{ij}}, x^+_{\tau_{ij}} \rangle$ and $\tau_{pn} = \langle x_{\tau_{pn}}, \sigma_{\tau_{pn}}, x^+_{\tau_{pn}} \rangle$ are such that $x_{\tau_{ij}}|_{S_m} = x_{\tau_{pn}}|_{S_m}$.

Thus, the *FN*-detector is a finite state machine given by the six-tuple $G_Z = \langle Z, z_0, \Sigma_Z, \delta_Z, Y_Z, z_f \rangle$ where Z is the set of states, z_0 is the initial state, $\Sigma_Z = X_m \times \Sigma_m$is the input alphabet, δ_Z is the transition function, Y_Z is the output function and z_f is the final state. Here, $\delta_Z : Z \times \Sigma_Z \rightarrow Z$ and $Y_Z : Z \times \Sigma_Z \rightarrow \{0, 1\}$.

3.2.1. FN-detector under measurement limitation

Let us now examine the feasibility of an *FN*-detector under measurement limitation $S_m = \{v_2\}$ and $I_m = \{v_3\}$. In this case, the transition $\tau_{16} = \langle x_{11}, 1, x_{13} \rangle$ is an FD_1-transition because there is a normal-*G*-transition namely, $\tau_{06} = \langle x_{01}, 1, x_{02} \rangle$ such that $x_{01}|_{S_m} = x_{11}|_{S_m} = 0$, $\sigma_{\tau_{06}}|_{I_m} = \sigma_{\tau_{16}}|_{I_m} = 1$ and $x_{02}|_{S_m} \neq x_{13}|_{S_m}$ because $x_{02}|_{S_m} = 0$ and $x_{13}|_{S_m} = 1$.

Interestingly, however, τ_{16} cannot detect the fault F_1 in an *FN*-detector as explained below. Suppose we proceed to construct an *FN*-detector as follows. Since τ_{16} is measured as $\langle x_{11}|_{S_m}, v_3|_{I_m}, x_{13}|_{S_m} \rangle = \langle v_2, v_3, v_2^+ \rangle = \langle 0, 1, 1 \rangle$, on detecting v_2^+ to be 0, the *FN*-detector would go to an intermediate state. Following that, if the input v_3 is measured to be 1, and v_2^+ is measured as 1, then the final state of the *FN*-detector is visited indicating fault F_1. However, the transition $\tau_{02} = \langle x_{02}, 1, x_{03} \rangle$ will also be measured as $\langle 0, 1, 1 \rangle$; in other words, τ_{02} is measurement equivalent to τ_{16}. Thus, under the measurement limitation being considered, the *FN*-detector cannot detect the fault. So, we may say that τ_{16} no longer remains so under measurement limitation $S_m = \{v_2\}$ and $I_m = \{v_3\}$. Let us now examine the feasibility of an *FN*-detector under another measurement limitation $S_m = \{v_1, v_2\}$ and $I_m = \{\}$, i.e., input v_3 is not tapped. In this case, the transition $\tau_{16} = \langle x_{11}, 1, x_{13} \rangle$ is an FD_1-transition because there is a normal-*G*-transition namely, $\tau_{06} = \langle x_{01}, 1, x_{02} \rangle$ such that $x_{01}|_{S_m} = x_{11}|_{S_m} = 10$, $\sigma_{\tau_{06}}|_{I_m} = \sigma_{\tau_{16}}|_{I_m} = \phi$ and $x_{02}|_{S_m} \neq x_{13}|_{S_m}$ because $x_{02}|_{S_m} = 00$ and $x_{13}|_{S_m} = 01$. Interestingly, unlike measurement restriction for v_1, τ_{16} (measurement restriction for v_3) can detect the fault F_1 in an *FN*-detector as explained below. Suppose we proceed to construct an *FN*-detector as follows. Since τ_{16} is measured as $\langle x_{11}|_{S_m}, v_3|_{I_m}, x_{13}|_{S_m} \rangle = \langle v_1 v_2, \phi, v_1^1 v_2^+ \rangle = \langle 10, \phi, 01 \rangle$, on detecting $v_1^+ v_2^+$ to be 10, the *FN*-detector would go to an intermediate state. Following that, if $v_1^+ v_2^+$ is measured as 01, then the final state of the *FN*-detector is visited indicating fault F_1. It may be noted that in the normal sub-machine (Fig. 3) there is no transition which is measured as $\langle 10, \phi, 01 \rangle$, thereby successfully completing the *FN*-detector construction. So, in this case of measurement limitation, the *FN*-detector capable of detecting the fault. In other words, we may say that τ_{16} remains an FD_1-transition under measurement limitation $S_m = \{v_1 v_2\}$ and $I_m = \{\}$.

Thus, it may be concluded that for some measurement limitation, certain FD_i-transition (under full measurement) become non-FD_i-transition. Before we proceed to the next section we formally define *FD*-transition under measurement limitation I_m and S_m.

Definition 6. FD_i-transition under I_m and S_m An F_i-*G*-transition $\tau_{ij} = \langle x_{\tau_{ij}}, \sigma_{\tau_{ij}}, x^+_{\tau_{ij}} \rangle$ is an FD_i-transition under I_m and S_m, if there is a normal-*G*-transition $\tau_{0l} = \langle x_{\tau_{0l}}, \sigma_{\tau_{0l}}, x^+_{\tau_{0l}} \rangle$ such that $x_{\tau_{0l}} E x_{\tau_{ij}}$, $\sigma_{\tau_{0l}}|_{I_m} = \sigma_{\tau_{ij}}|_{I_m}$ and not $x^+_{\tau_{0l}} E x^+_{\tau_{ij}}$. Further, there should not be any normal-*G*-transition $\tau_{0m} = \langle x_{\tau_{0m}}, \sigma_{\tau_{0m}}, x^+_{\tau_{0m}} \rangle$ such that $x_{\tau_{0m}} E x_{\tau_{ij}}, \sigma_{\tau_{0m}}|_{I_m} = \sigma_{\tau_{ij}}|_{I_m}$ and $x^+_{\tau_{0m}} E x^+_{\tau_{ij}}$. The set of all FD_i-transitions for fault F_i under I_m and S_m is denoted as $\Im_{FD_i}|_{I_m, S_m}$.

The inherent problem of constructing the *FN*-detector from the DES model is that the method becomes prohibitively complex even for simple VLSI circuits because the explicit DES model of a circuit is exponential in number of flip-flops in the circuit. In the next section, we propose a scheme which is capable of detecting the *FD*-transitions (with measurement limitation) directly from the circuit description without the need of the explicit DES model and therefore, can be applied to fairly complex circuits.

4. Efficient construction of *FN*-detector

The NSF block is a combinational circuit with two types of inputs namely, the Primary Inputs (PI) I and the secondary inputs S (which are feedback from the flip-flop outputs). The NSF block outputs, denoted collectively as S^+, determine the next state.

The NSF block can be described by the tuple $\mathbb{S} = \langle \Sigma_S, S^+ \rangle$, where $\Sigma_S = X \times \Sigma$is the alphabet of input symbols (patterns) and $S^+ = \{v_1^+, v_2^+, \ldots, v_k^+\}$ is the set of outputs lines. For each $v_i^+ \in S^+, 1 \leqslant i \leqslant k, v_i^+ : \Sigma_S \rightarrow \{0, 1\}$. Thus, an NSF output line v_i^+ also designates the switching function realized on this line. An input combination $\sigma_s \in \Sigma_s$ is a mapping from $V = \{v_1, v_2, \ldots, v_k, \ldots, v_n\}$ to $\{0, 1\}$ represented as an n-tuple $\langle \sigma_s(v_1), \sigma_s(v_2), \ldots, \sigma_s(v_k), \sigma_s(v_{k+1}), \ldots, \sigma_s(v_n) \rangle$, where the first k members constitute a k-tuple of secondary inputs and the remaining $(n-k)$ members constitute an $(n-k)$-tuple of primary inputs.

Let $S_i^+ = \{v_{i1}^+, v_{i2}^+, \ldots, v_{ik}^+\}$ denote the output maps represented by the NSF block under fault F_i; similarly let $S_0^+ = \{v_{01}^+, v_{02}^+, \ldots, v_{0k}^+\}$ denote the output maps represented by the NSF block under normal condition. If we speak of the NSF block without the context of faults (i.e., only in the normal condition), then $S^+ = \{v_1^+, v_2^+, \ldots, v_k^+\}$ denote its outputs. An FD_i-transition $\tau_{im} = \langle x_{\tau_{im}}, \sigma_{\tau_{im}}, x^+_{\tau_{im}} \rangle$ can be determined from the NSF block netlist description in the following manner:

For the given stuck-at fault F_i, determine a value of the input combination of the NSF block i.e., $\sigma_s \in \Sigma_s$ which sensitizes the fault and propagates the effect through the NSF block in at least one of its outputs, i.e., $\exists j, 1 \leqslant j \leqslant k, v_{ij}^+(\sigma_s) \neq v_{0j}^+(\sigma_s)$. As the secondary inputs of the NSF block are the outputs of the state flip-flops, a secondary input pattern denotes the current state from where there is an *FD*-transition. Hence, the first k-tuple of $\sigma_s = initial(\tau_{im}) = x_{\tau_{im}}$.

Similarly, the second $(n-k)$-sub-tuple of $\sigma_s = input(\tau_{im}) = \sigma_{\tau_{im}}$. The output of the NSF block (with fault F_i) corresponding to the input σ_s gives $final(\tau_{im}) = x^+_{\tau_{im}}$ as $\langle v^+_{i1}(\sigma_s), v^+_{i2}(\sigma_s), \ldots, v^+_{ik}(\sigma_s)\rangle$. To determine the set $\Im_{FD_i}$, all possible values of σ_s are to be determined such that $\exists j, 1 \leqslant j \leqslant k, v^+_{ij}(\sigma_s) \neq v^+_{0j}(\sigma_s)$.

Now we study *FD*-transitions under measurement limitation. Given $\Im_{FD_i}$, subject to measurement limitation I_m and S_m, some of the FD_i-transitions may not remain so. An input combination $\sigma_s \in \Sigma_s$ under I_m, S_m is represented as an n-tuple $\langle \sigma_s(v_1), \sigma_s(v_2), \ldots, \sigma_s(v_k), \sigma_s(v_{k+1}), \ldots, \sigma_s(v_n)\rangle$, where $\sigma_s(v_i)$ is d (don't care), if $v_i \notin S_m \cup I_m$. For example, $\langle d, \sigma_s(v_2), \ldots, \sigma_s(v_k), d, \ldots, \sigma_s(v_n)\rangle$ represents the input combination when $v_1 \notin S_m$ and $v_{k+1} \notin I_m$, i.e., NSF block output lines v^+_1 and v^+_{k+1} are not measured. The input combination $\sigma_s \in \Sigma_s$ under I_m, S_m represents a set of input combinations (under full measurement) which are obtained by replacing each d with 0 and 1. For example, if $\sigma_s = \langle d, \sigma_s(v_2), \ldots, \sigma_s(v_k), d, \ldots, \sigma_s(v_n)\rangle$ under $v_1 \notin S_m$ and $v_{k+1} \notin I_m$, then $\sigma_s|_{I_m,S_m} = \{\langle 0, \sigma_s(v_2), \ldots, \sigma_s(v_k),\ 0, \ldots, \sigma_s(v_n)\rangle, \langle 0, \sigma_s(v_2), \ldots, \sigma_s(v_k), 1, \ldots, \sigma_s(v_n)\rangle, \langle 1, \sigma_s(v_2), \ldots, \sigma_s(v_k), 0, \ldots, \sigma_s(v_n)\rangle, \langle 1, \sigma_s(v_2), \ldots, \sigma_s(v_k), 1, \ldots, \sigma_s(v_n)\rangle\}$.

An FD_i-transition $\tau_{im} = \langle x_{\tau_{im}}, \sigma_{\tau_{im}}, x^+_{\tau_{im}}\rangle$ remains so under I_m and S_m if

- Fault propagation is through a measured NSF output line: If σ_s (here $x_{\tau_{im}} \times \sigma_{\tau_{im}}$) sensitizes the fault and propagates the effect through the NSF block in at least one of its *measured* outputs, i.e., $\exists j, 1 \leqslant j \leqslant k, v^+_{ij}(\sigma_s) \neq v^+_{0j}(\sigma_s) \wedge v_j \in S_m$.
- Under measurement limitation τ_{im} does not become equivalent to any *N*-transition: $\neg\exists$ *N*-transition $\tau_{0l} = \langle x_{\tau_{0l}}, \sigma_{\tau_{0l}}, x^+_{\tau_{0l}}\rangle$, such that $\exists(x_{\tau_{0l}} \times \sigma_{\tau_{0l}}) \in \sigma_s|_{I_m,S_m}$ and $v^+_{0j}(x_{\tau_{0l}} \times \sigma_{\tau_{0l}}) = v^+_{ij}(\sigma_s), \forall v_j \in S_m$.

For example, $\tau_{16} = \langle x_{11}, 1, x_{13}\rangle$ (Fig. 2) is an *FD*-transition. Here, $x_{\tau_{16}} = x_{11} = 10, \sigma_{\tau_{16}} = 1, x^+_{\tau_{16}} = x_{13} = 01$. Also, input combination is $\sigma_s \equiv \langle \sigma_s(v_1), \sigma_s(v_2), \sigma_s(v_3)\rangle = x_{11} \times 1 = 101$. If $S_m = \{v_1, v_2\}$ and $I_m = \{\}$, then input combination set $\sigma_s|_{I_m,S_m} \equiv 101|_{v_3}$ is $10d = \{100, 101\}$. Here, $v^+_{01}(100) = v^+_{11}(101) = 0$ but $v^+_{02}(100) = 0 \neq v^+_{12}(101) = 1$. Similarly, $v^+_{01}(101) = v^+_{11}(101) = 0$ but $v^+_{02}(101) = 0 \neq v^+_{12}(101) = 1$. So $\tau_{16} = \langle x_{11}, 1, x_{13}\rangle$ remains an FD_1-transition even under $S_m = \{v_1, v_2\}$ and $I_m = \{\}$. Now let us consider measurement imitation $S_m = \{v_2\}$ and $I_m = \{v_3\}$; input combination set $\sigma_s|_{I_m,S_m} \equiv 101|_{v_1}$ is $d01 = \{001, 101\}$. Here, $v^+_{01}(001) = v^+_{11}(101) = 0$ and $v^+_{02}(001) = v^+_{12}(101) = 1$. So $\tau_{16} = \langle x_{11}, 1, x_{13}\rangle$ does not remain an FD_1-transition under $S_m = \{v_2\}$ and $I_m = \{v_3\}$. The same conclusion was arrived at in Section 3.2.1.

Given a netlist description of the NSF block of the circuit, the set of *FD*-transitions under I_m and S_m can be determined in the following manner:

1. Simulate the NSF block under normal condition to determine the output response for all input combinations.
2. Insert the stuck-at fault at the proper point in the NSF.
3. Simulate the NSF block with fault for all possible input combinations.
4. Determine all possible values of inputs such that the output is different under fault and normal condition; the corresponding transitions are *FD*-transitions under full measurement (also called test patterns in off-line test terminology [22]).
5. For each *FD*-transition check if it remains so, under I_m and S_m.
6. Repeat steps (1) to (5) for all possible s-a faults.

Step (1) through Step (4) basically involve determining all possible values of input combination for the NSF block that sensitize the fault and propagate its effect through at least one NSF block output; this is called exhaustive TP generation [22]. Exhaustive TP generation is a computationally hard problem. Further, to determine whether a test pattern under full measurement remains so, even under a given measurement limitation requires $O(2^k)$, where $k = n - (|S_m| + |I_m|)$, times analysis of the normal circuit. In other words, a test pattern under full measurement represents a set of patterns under measurement limitation, which are obtained by replacing each unmeasurable input by 0 and 1 ($O(2^k)$ in number). So, like exhaustive test pattern generation procedure, this process of checking test patterns under measurement limitation also involves exponential complexity. Hence, we require optimized techniques for this problem.

The subsections that follow provide the details of these optimization steps. In essence, these optimizations result by representing the NSF outputs as Ordered Binary Decision Diagrams (OBDDs) [42] and devising processing steps to work on these OBDD representations.

4.1. Exhaustive TP generation for the NSF block under full measurement

In this section we discuss the procedure for exhaustive TP generation for a fault F_i for a given output line of the NSF block (under full measurement). Without compromising fault detection capability, we assume that even if F_i is manifested at more than one NSF output line, any one of these lines can be used for its detection. Given an NSF block output v^+_j and a fault F_i, two OBDDs are generated for the Boolean functions v^+_{0j} and v^+_{ij}, the former for the normal condition and the latter under F_i. The two OBDDs are XORed and the exhaustive set of input test vectors for F_i (that is, the exhaustive set of TPs) is the result of "satisfy-all" operation on the resulting XORed OBDD because all paths leading to 1 in the XORed OBDD represent the exhaustive set of input patterns for which the output under normal condition is different from that under the fault. The output response v^+_{ij} , $1 \leqslant j \leqslant k$, for the fault F_i, for the given set of input test vectors, can be easily obtained form the OBDD for v^+_{ij} by applying the test patterns; this process can go hand in hand as the test patterns are obtained. Let us now illustrate this procedure for the s-a-1 fault, termed as F_1, at the fanout net marked *A* in the circuit shown in Fig. 2. In this example, we illustrate the OBDD with the ordering $v_1 \prec v_2 \prec v_3$. Fig. 5 represents the OBDD for the Boolean function $v^+_{02} = v_1 v_2 + \overline{v_1}\ \ \overline{v_2}$. Fig. 6 represents the OBDD for the Boolean function of $v^+_{12} = v_1 v_2 \overline{v_3} + v_1 v_3 + \overline{v_1}\ \ \overline{v_2}$. Fig. 7 represents the OBDD corresponding to $v^+_{02} \oplus v^+_{12}$ obtained by XORing the normal OBDD and the F_1-OBDD illustrated in Figs. 5 and 6, respectively. The exhaustive set of input patterns to test the fault F_1 is obtained using the

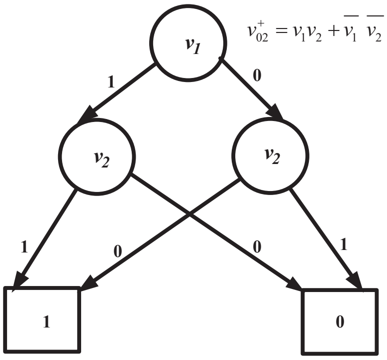

Fig. 5. OBDD for the function v^+_{02} (circuit shown in Fig. 2).

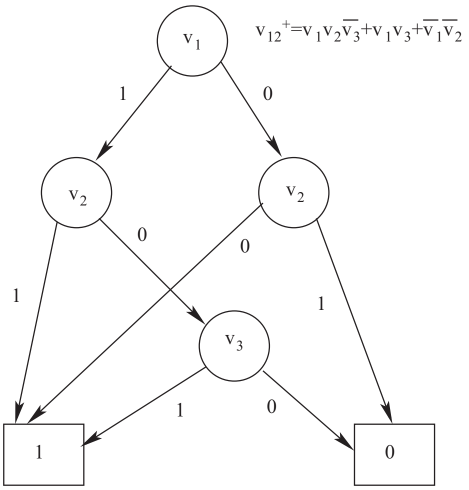
$v_{12}^+ = v_1 v_2 \overline{v_3} + v_1 v_3 + \overline{v_1}\,\overline{v_2}$
v1
1
0
v2
v2
0
0
1
1
v3
1
0
1
0

Fig. 6. OBDD for the function v_{12}^+ (circuit shown in Fig. 2).

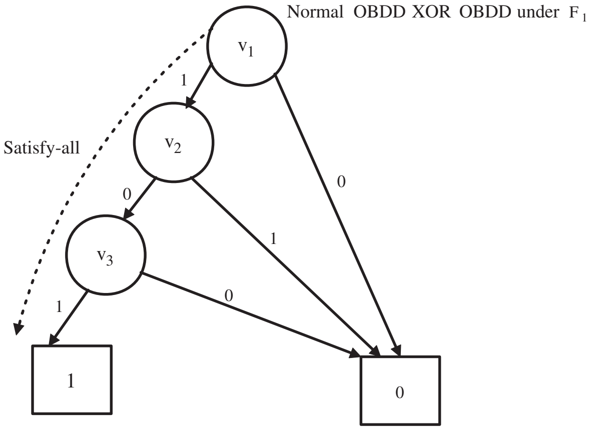

Fig. 7. XOR of the v_{02}^+ and v_{12}^+ OBDD (circuit shown in Fig. 2).

satisfy-all operation on the XORed OBDD (Fig. 7) as $\{\langle v_1 = 1, v_2 = 0, v_3 = 1\rangle\}$, which corresponds to the single path from the root node (v_1) of the XORed OBDD to the leaf node 1. The output response under fault (F_1) for this input test pattern can be easily determined using the F_1-OBDD shown in Fig. 6 as $v_{12}^+ = 1$. Thus, for the given fault F_1 and the NSF block output v_2^+, the exhaustive TP set corresponding to the tuple $\langle v_1 v_2 v_3\rangle$ is given by $\{\langle 101\rangle\}$. The output for $\{\langle 101\rangle\}$ corresponding to the tuple $\langle v_1^+ v_2^+\rangle$ is $\{\langle d1\rangle\}$. Thus the *FD* transition set for fault F_1 and the NSF block output v_2^+ is $\{\langle 10, 1, d1\rangle\}$.

FD-transitions determined by the procedure discussed in the last paragraph detects a fault by monitoring its manifestation at one NSF block output. However, it may happen that for a given TP, F_i is manifested at one output line (say v_1^+) of the NSF block and for another TP, F_i is manifested at some other output line (say v_2^+). While generating the exhaustive *FD*-transitions for the output v_1^+, we ignore the *FD*-transitions which may lead to manifestation of F_1 through v_2^+. Ignoring these *FD*-transitions corresponding to v_2^+ may lead to rise in detection latency because they may include some input combinations which are not covered in the *FD*-transitions for v_1^+. To address this problem, therefore, the exhaustive *FD*-transition sets are generated for each NSF output and for each F_i, which are then used for designing an *FN*-detector for the entire NSF block. Let $\tau_{im}^j = \langle x_{\tau_{im}}^j, \sigma_{\tau_{im}}^j, x_{\tau_{im}}^{+j}\rangle$ denote the m^{th} *FD*-transition for the fault F_i determined at the NSF block output v_j^+. The values of state variables in $x_{\tau_{im}}^j$ are don't care values for members of S^+ whose corresponding members in S do not fall under the cone of influence[3] of the NSF block output (v_j^+) being considered. Further, as the values of the variables corresponding to $x_{\tau_{im}}^j$ conjoined with $\sigma_{\tau_{im}}^j$ are determined using OBDD (based XOR operation), even some of these variables that fall in the cone of influence of v_j^+ may be don't cares. Also, *only one variable* in $x_{\tau_{im}}^{+j}$, namely, v_j^+, that corresponds to the NSF block output through which the fault manifestation is monitored has a Boolean value of 0 or 1; rest are don't cares.

Once the exhaustive set of *FD*-transitions are generated for each fault, we check whether the *FD*-transitions remains so under a given measurement limitation.

4.2. Determination of FD-transitions under measurement limitation

Consider an *FD*-transition $\tau_{im}^j = \langle x_{\tau_{im}}^j, \sigma_{\tau_{im}}^j, x_{\tau_{im}}^{+j}\rangle$ for the fault F_i determined at the NSF block output v_j^+. Now we discuss the procedure to check if $\tau_{im}^j = \langle x_{\tau_{im}}^j, \sigma_{\tau_{im}}^j, x_{\tau_{im}}^{+j}\rangle$ remains an FD_i-transition under measurement limitation I_m and S_m, using OBDDs. Obviously, only those NSF bock outputs are considered whose corresponding $v_j \in S_m$ (i.e., are measurable). In other words, all *FD* transitions where fault manifestation is only through NSF outputs which are unmeasurable, can be directly dropped. So in this case, if v_j is measurable (i.e., $v_j \in S_m$) then we proceed for further verification steps. Following that we determine the value of NSF output (v_j^+) under faulty condition for input $\sigma_{sm}^j = x_{\tau_{im}}^j \times \sigma_{\tau_{im}}^j$ by tracing the path in the OBDD (representing output function for v_{ij}^+) corresponding to values of the variables in σ_{sm}^j; let the value be $val_{im}^j \in 0, 1$. Now, for each input combination under measurement limitation I_m, S_m i.e., $\sigma_s \in \sigma_{sm}^j|_{I_m,S_m}$, we need to determine the value of NSF output (v_j^+) under normal condition for input σ_s by tracing the path in the OBDD (representing output function for v_{0j}^+) corresponding to values of the variables in σ_s; let the value be $val_{0\sigma_s}^j \in 0, 1$. If $val_{0\sigma_s}^j \neq val_{im}^j$, for all σ_s, then τ_{im}^j remains an FD_i-transition under measurement limitation I_m and S_m. Alternatively, if $val_{0\sigma_s}^j = val_{im}^j$ for some input condition $\sigma_s \in \sigma_{sm}^j|_{I_m,S_m}$, then for that σ_s, the corresponding NSF output v_j^+ gives same value both in normal and faulty case; τ_{im}^j does not remain an FD_i-transition under measurement limitation.

Now we will illustrate the concept with the fault s-a-1 in line A (of Fig. 2, whose DES model is shown in Fig. 3) for measurement limitation (i) $I_m = \{\}$ and $S_m = \{v_1, v_2\}$ (ii) $I_m = \{v_3\}$ and $S_m = \{v_2\}$. The FD_1 transition is $\tau_{16}^2 = \langle 10, 1, d1\rangle$ corresponding to NSF output v_2. Here, $\sigma_{s6}^2 = 10 \times 1$. If we traverse the OBDD for v_{12}^+ (for the fault, shown in Fig. 6) for the input combination 101 we have $val_{16}^2 = 1$. For $I_m = \{\}$ and $S_m = \{v_1, v_2\}$, $\sigma_s \in \sigma_{s6}^2|_{I_m,S_m} = \{100, 101\}$. From the OBDD for v_{02}^+ (for the normal circuit, shown in Fig. 5), we have $val_{0\sigma_s}^2 = 0$ for both the input combinations of $\sigma_s = 100$ and 101, which is not equal to $val_{16}^2 = 1$. So $\langle 10, 1, d1\rangle$ remains an FD_1 transition.

For $I_m = \{v_3\}$ and $S_m = \{v_2\}, \sigma_s \in \sigma_{16}^2|_{I_m,S_m} = \{001, 101\}$. From the OBDD for v_{02}^+, we have $val_{0\sigma_s}^2 = 1$ for combination $\sigma_s = 001$ which is equal to $val_{16}^2 = 1$. So $\langle 10, 1, d1\rangle$ does not remain an FD_1 transition.

[3] In terms of VLSI testing the sub-circuit that is in the transitive fanins of an (NSF) output, v_j^+ say, is called the “sub-circuit in the cone of influence of v_j^+” [22].

Note: The above procedure to check if τ_{im}^{j} remains an *FD*-transition under measurement limitation I_m, S_m requires exponential number (with respect to unmeasurable lines) of checks in the normal OBDD. However, using OBDD we can perform this step efficiently as discussed below. In the OBDD representing the NSF output corresponding to v_j^+, under normal condition, the following steps are required.

1. If $v_k \in S_m \cup I_m$ and $\sigma_{sm}^{j}(v_k) = 0$ (or 1) then in the node of the OBDD corresponding to v_k, eliminate the edge corresponding to 1 (or 0).
2. Delete all edges and nodes unreachable from the root after elimination of the edge.
3. Repeat step 1 and 2 for all $v_k \in S_m \cup I_m$.
4. In the resultant OBDD if there is a path from root to leaf whose value is same as that of the corresponding faulty OBDD for input combination σ_{sm}^{j} then τ_{im}^{j} does not remain an FD_i-transition under measurement limitation; else it remains an FD_i-transition.

In simple words, given an FD_i-transition, we first replace all the variables of the input combination with *d* which are unmeasurable. Now, in the normal OBDD, given a variable of the input combination, we determine the corresponding nodes and keep the edge representing 0 or the edge representing 1 or both the edges, if the value of the variable is 0 or 1 or *d*, respectively. This process is repeated for all variables of the input combination. In the resultant OBDD, if there is a path to a leaf node whose value is same as that of the faulty OBDD for the input combination of the given FD_i-transition, then it does not remain an FD_i-transition under the given measurement limitation.

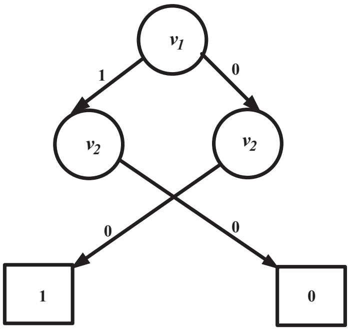

Fig. 8. OBDD for v_{02}^{+} after edges and nodes for *FD*-transition $\langle 10, 1, d1\rangle$ being eliminated for $I_m = \{v_3\}$ and $S_m = \{v_2\}$.

Let us consider the same s-a-1 fault (of Fig. 2, whose DES model is shown in Fig. 3) for measurement limitation $I_m = \{v_3\}$ and $S_m = \{v_2\}$. The FD_1 transition in this case is $\tau_{16}^{2} = \langle 10, 1, d1\rangle$ corresponding to NSF output v_2. As already discussed, $val_{16}^{2} = 1$. In the normal OBDD (Fig. 5), as $v_1 \neg\in S_m$ we retain both the edges for the variable v_1. As $v_2 \in S_m$ and value in the *FD*-transition is 0, we eliminate the edges corresponding to 1 in the nodes for v_2. Finally, as $v_3 \in I_m$ and value in the *FD*-transition is 1, we need to eliminate the edges corresponding to 0 in the nodes for v_3; however, this need not be done as the nodes for v_3 are redundant and already eliminated by the OBDD construction. Fig. 8 shows the normal OBDD for v_{02}^{+} after edges and nodes for *FD*-transition $\langle 10, 1, d1\rangle$ are eliminated for $I_m = \{v_3\}$ and $S_m = \{v_2\}$. It may be noted that there is path from the root node to leaf node with value 1, which is same as in the faulty OBDD (Fig. 6) for input combination $\langle 101\rangle$. So $\langle 10, 1, d1\rangle$ does not remain an FD_1 transition. In a similar way, it can be shown that τ_{16}^{2} remains FD_1-transition under measurement limitation $I_m = \{\}$, $S_m = \{v_1, v_2\}$. The *FN*-detector comprising FD_1-transition $\tau_{16}^{2} = \langle 10, 1, d1\rangle$ under measurement limitation $I_m = \{\}$ and $S_m = \{v_1, v_2\}$ for NSF output v_2^{+} is shown in Fig. 9.

4.3. Complexity of FN-detector construction for the circuit under test

As discussed in Sub-Section 3.2, the construction of the *FN*-detector involves the following broad steps for each *FD*-transition – (i) create an initial state (z_0), if not already there; (ii) create a final state (z_f), if not already there; (iii) create an intermediate state z_i; (iv) add two transitions, one is from z_0 to z_i and another from z_i to z_0 ; (v) add a transition from z_i to z_f. As these steps involve constant

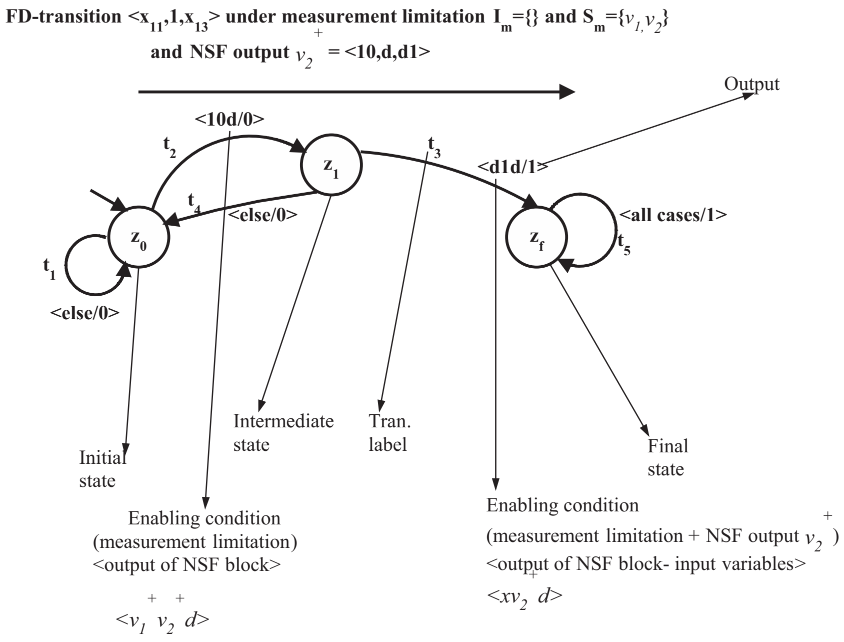

Fig. 9. The *FN*-detector for *FD*-transition τ_{16} under $I_m = \{\}, S_m = \{v_1, v_2\}$ and NSF output v_2^{+}.

computation, complexity of *FN*-detector construction is of the same order as that of the generation of *FD*-transitions. However, generation of *FD*-transitions for a CUT takes significant amount of time. In summary, the complexity of construction of *FN*-detector mainly depends on the complexity of generation of all *FD*-transitions for the CUT. The steps along with their time complexity for generation of *FD*-transitions for a fault F_i through the NSF block output v_j^+ are as follows.

- Generate OBDD, B_N, for the expression v_j^+ under normal condition. The time complexity to generate B_N is $O(|B_N|.log(|B_N|))$, where $|B_N|$ is the size of the B_N [42,47]. Again, the size of the OBDD depends on the variable ordering. The variable ordering makes a significant difference to the size of the OBDD. In best case, the size of the OBDD grows linearly with the number of variables and in worst case it grows exponentially with the number of variables. This implies that, for $B_N, O(n) \leqslant |B_N| \leqslant O(2^n)$, where n is the number of input variables of the CUT (NSF in this case). Thus, the lower bound and upper bound time complexities for generation of B_N are $O(n.log(n))$ and $O(2^n)$, respectively.
- Generate OBDD, B_{F_i}, for the expression v_{ij}^+ i.e., OBDD for NSF output v_j^+ under a stuck-at fault F_i. The time complexity of generation of B_{F_i} is same as that of B_N because $|B_N| \equiv |B_{F_i}|$.
- Perform XOR operation between B_N and B_{F_i} and generate XOR-OBDD (say B_{XOR}). The time complexity of performing XOR-operation is $O(|B_N|.|B_{F_i}|)$ [42,47].
- Generate *FD*-transitions under full measurement by applying *satisfy-all* operation to the XOR-OBDD. The time complexity of satisfy-all operation on B_{XOR} is $O(|B_{XOR}|.|\Im_{FD_i}|)$, where $|B_{XOR}|$ is the size of B_{XOR} and $|\Im_{FD_i}|$ is number of *FD*-transitions for the fault F_i.
- Determine the *FD*-transitions which remain so under the given measurement limitation. This can be accomplished by modifying the normal OBDD B_N. The modification is performed by traversing all the nodes of B_N (in the worst case) and removing some of its edges. The time required to do the same is $O(|B_N|)$ [48]. Finally, the modified normal OBDD B_N, is compared with the faulty OBDD B_{F_i}, which involves complexity of the $O(|B_N|.|B_{F_i}|)$.
- Thus, the total time complexity to generate *FD*-transitions for a fault F_i is $O(|B_N|.log(|B_N|) + |B_{F_i}|.log(|B_{F_i}|) + |B_N|.|B_{F_i}| + |B_{XOR}|.|\Im_{FD_i}| + |B_N| + |B_N|.|B_{F_i}|) \equiv O(|B_N|.|B_{F_i}|)$.

The above steps are repeated twice for all lines of the CUT because each line has two type of faults-stuck-at-0 and stuck-at-1. If a circuit has total *nl* number of lines, then the above OBDD based operations are repeated $2nl$ times in order to generate all the *FD*-transitions under a given measurement limitation. Thus, the total time complexity is $O(nl.|B_N|.|B_{F_i}|)$. However, practically the complexity is much lower than the worst case upper bound mentioned above. This is due to fault collapsing, which significantly decreases the number of stuck-at faults by taking a single fault to represent a group of faults that have the same effect on the circuit. Table 1 shows the effect of fault collapsing on different ISCAS 89 benchmark circuits [49]. From the table it may be noted that due to fault collapsing there is more than 50% reduction in the number of faults that need to be really considered.

Table 1
Effect of fault collapsing.

Circuit name	# of initial faults (= $2nl$)	# of faults after collapsing
S386	772	384
S510	1020	564
S5378	10756	4603
S9234	18468	6927
S15850	31700	11727

Table 2
Number of primary inputs: circuit verses it's largest and smallest cones.

Circuit name	# Primary inputs	Cone level	
		# Primary inputs for largest cone	# Primary inputs for smallest cone
S27	4	3	2
S386	7	6	3
S510	19	15	8
S9234	19	13	10
S15850	14	11	7

In the worst case scenario the complexity of generation of the exhaustive set of *FD*-transitions for a circuit is exponential with respect to the number of primary inputs of CUT because $O(n^2) \leqslant |B_N|.|B_{F_i}| \leqslant O(2^n)$. Since, at a time we generate *FD*-transitions for a fault through an individual NSF block output, thus, the complexity reduces to the number of primary inputs belonging to that NSF block output. The number of primary inputs belonging to individual NSF block output of a circuit can be determined using cones of influence [35]. A cone is a sub-circuit of an NSF block output, which contains all gates, nets and inputs that are transitive fan-ins of the output. Table 2 shows the information about the number of primary inputs of different ISCAS 89 benchmark circuits and their largest and smallest cones. It can be observed from the table that the number of primary inputs belonging to a cone is lower than the number of primary inputs of the original circuit. So the practical time complexity of generation of *FD*-transitions and finally the construction of *FN*-detector is much less than the value determined formally above.

5. Implementation and experimental evaluation

The technique discussed in Sections 3 and 4 was used to design a tool "ML-OLT", which generates an *FN*-detector (in Verilog RTL) given a digital sequential circuit (in netlist format). The Verilog code can be synthesized using any standard synthesis tool, design library and user defined constraints. Results regarding detection latency, fault coverage and area overhead for different measurement limitations are discussed in this section.

5.1. Tradeoffs in FN-detector design: detection latency, fault coverage, measurement limitation and area overhead

The fault detection capability of an *FN*-detector can be expressed in terms of two parameters namely, fault coverage and detection latency. The former deals with coverage of all possible faults conforming to the single s-a fault model. Detection latency implies the number of times a fault is manifested at some output of the NSF block due to occurrence of an *FD*-transition at NSF input, however, is not detected because that *FD*-transition is not considered in the *FN*-detector due to measurement limitation. Detection latency may increase due to two factors namely, (i) some *FD*-transitions may be kept out of the purview of the *FN*-detector or (ii) measurement limitation, which in turn eliminates some *FD*-transitions. Thus, in the case where all *FD*-transitions are captured in the *FN*-detector, any fault is captured immediately after it causes the first measurable behavioral difference; thus the detection latency is zero. Clearly, if some of the *FD*-transitions are dropped in the construction of the *FN*-detector, then the detection latency may increase. This is because the *FD*-transitions that are not taken in the *FN*-detector may occur before the *FD*-transitions that are taken.

In [35], the area overhead was reduced by eliminating some *FD*-transitions. In the present work we will use measurement limitation as a tradeoff factor to minimize area overhead by compromising detection latency and fault coverage. In [35], for most of the cases (i.e., for the *FD*-transition set selected for the *FN*-detector) all primary inputs and NSF block output lines had to be measured. In the experimental results we will show that similar detection latency, fault coverage and area overhead can be achieved by the proposed scheme compared to [35], however, with reduction in the lines to be measured.

The CAD tool "ML-OLT" was used to generate OLT circuits for different ISCAS'89 benchmark circuits under various measurement limitations. First we illustrate results (as graphs) for fault coverage, detection latency and area overhead of the circuit s1488 under different combinations of measurement limitations. We have illustrated results where 1 or 2 lines are considered unmeasurable. The following measures were used to determine the values of fault coverage, detection latency and area overhead for a tester given a measurement limitation.

- *Detection Latency (DL):* For a fault F_1 (which is covered), let there be nFD_1 number of *FD*-transitions under full measurement. After measurement limitation let $nmlFD_1$ be the number of *FD*-transitions that remain. Detection latency is $(\lceil nFD_1/nmlFD_1 \rceil)$-1.
- *Area overhead (AO):* (Area of the *FN*-detector after synthesis)/(Area of circuit under test after synthesis).
- *Fault coverage (FC):* We consider a fault to be covered if at least one *FD*-transition of the fault remains in the *FN*-detector under measurement limitation. Fault coverage = (number of faults covered)/(number of faults in the circuit) $*$ 100%.

5.2. Evaluation results: detection latency, fault coverage and area overhead

Figs. 10–12 illustrate detection latency, area overhead and fault coverage, respectively of s1488 versus different combinations of measurement limitations of one and two input lines of NSF. Under this case of one and two input lines being unmeasured, s1488 has 105 combinations of unmeasurable lines. In the graph, points in the *x*-axis represent the line(s) not being measured; for example, the first point $V0$ represents that input $V0$ is not measured whereas $V0\_V1$ represents that $V0$ and $V1$ are not measured. To keep the markings of the *x*-axis legible we illustrate only 55 combinations. The following points may be noted:

- Fig. 10: Impact of not measuring different (single or double) input lines may have different impact on detection latency. As already discussed, making line(s) unmeasurable results in converting some *FD*-transitions to non-*FD*-transitions which leads to raise in detection latency. Broadly speaking, the input lines whose transitive fanouts have more fault sites (i.e., gates) have higher sensitivity to detection latency.
- Fig. 11: Lower the detection latency, higher the area overhead. Higher detection latency implies that unmeasurable lines resulted in converting more *FD*-transitions to non-*FD*-transitions compared to a situation with lower detection latency. Generally, speaking an *FN*-detector with less *FD*-transitions involve less states resulting in lower area and vive-versa. Same detection latency (due to different combinations of unmeasured lines) may also result in different area overheads. Detection latency implies that some *FD*-transitions are not considered, however, it does not specify which

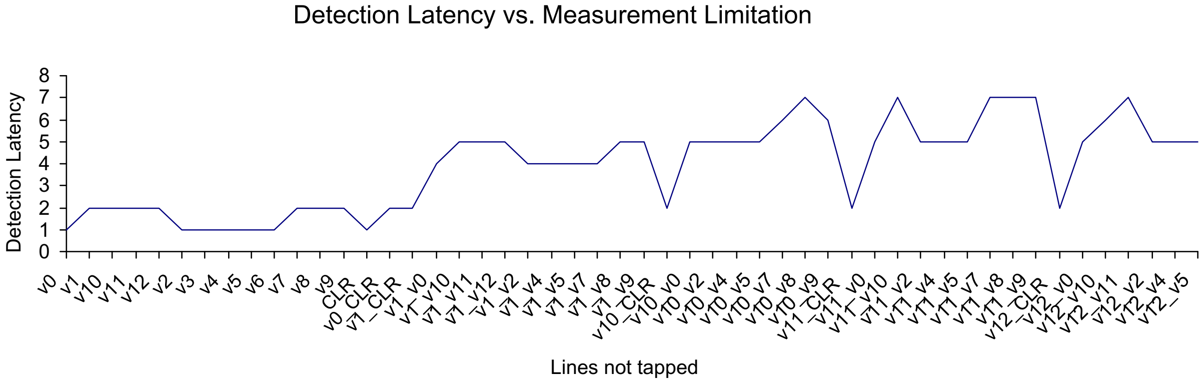

Fig. 10. Detection latency for s1488 versus different combinations of measurement limitations of one and two input lines of NSF.

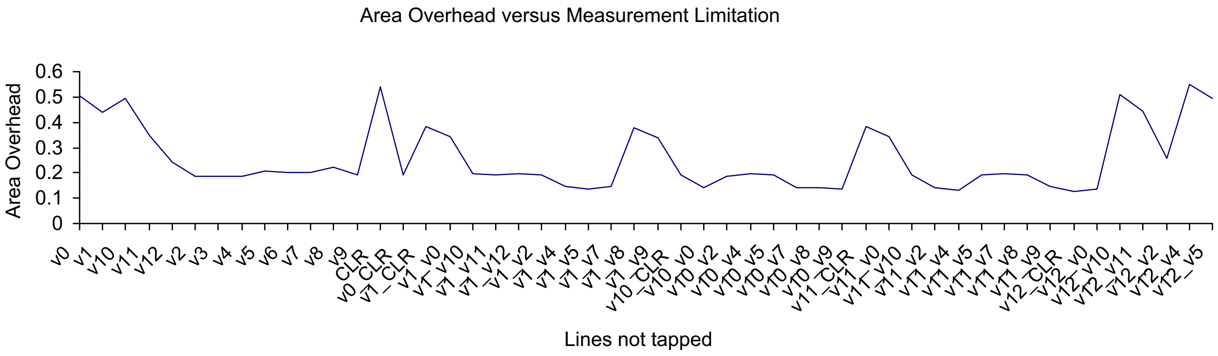

Fig. 11. Area overhead for s1488 versus different combinations of measurement limitations of one and two input lines of NSF.

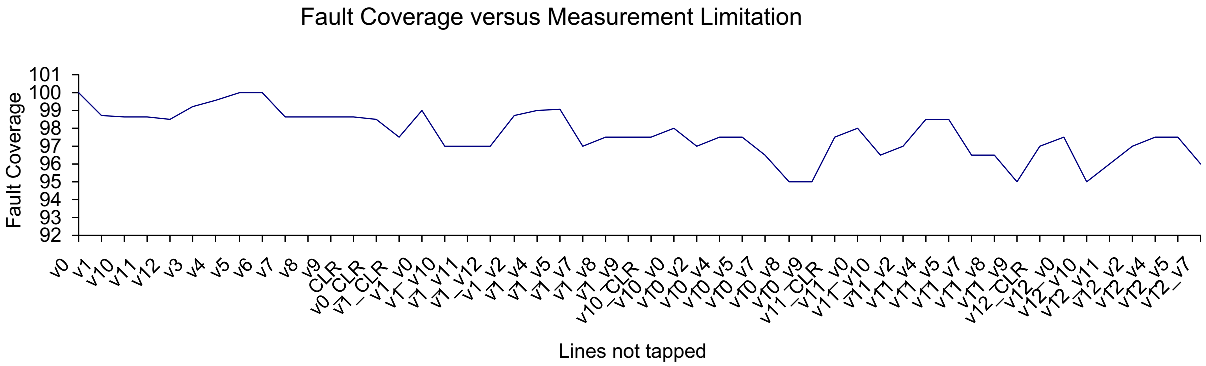

Fig. 12. Fault coverage for s1488 versus different combinations of measurement limitations of one and two input lines of NSF.

Table 3
Area overhead (AO) for different combinations of measurement limitations, resulting detection latency (DL) and AO comparison with [35,34].

Circuit			AO for different combinations of measurement limitations					
Circuit	No. of PI	No. of flops-(Gate count)	One line		Two lines		Three lines	
			Best case AO	Worst case AO	Best case AO	Worst case AO	Best case AO	Worst case AO
S27	4	3-(10)	AO:2.41 DL:2 AO[]:2.45	AO:2.99 DL:1 AO[]:2.99	AO:2.34 DL:4 AO[]:2.39	AO:2.66 DL:1 AO[]:2.99	AO:2.11 DL:9 AO[]:2.19	AO:2.51 DL:2 AO[]:2.45
S298	3	14-(119)	AO:0.86 DL:2 AO[]:1.01	AO:1.36 DL:1 AO[]:1.88	AO:0.63 DL:4 AO[]:0.78	AO:1.34 DL:1 AO[]:1.88	AO:0.42 DL:8 AO[]:0.62	AO:1.32 DL:1 AO[]:1.88
S386	7	6-(118)	AO:0.86 DL:2 AO[]:1.11	AO:1.38 DL:1 AO[]:1.80	AO:0.62 DL:4 AO[]:0.85	AO:1.25 DL:1 AO[]:1.80	AO:0.51 DL:7 AO[]:0.70	AO:1.13 DL:1 AO[]:1.80
S510	19	6-(211)	AO:0.79 DL:2 AO[]:1.1	AO:1.2 DL:1 AO[]:1.75	AO:0.70 DL:3 AO[]:0.88	AO:1.1 DL:1 AO[]:1.75	AO:0.60 DL:6 AO[]:0.72	AO:1.0 DL:1 AO[]:1.75
S5378	35	179-(2779)	AO:0.4 DL:2 AO[]:0.44	AO:0.98 DL:1 AO[]:0.98	AO:0.23 DL:3 AO[]:0.31	AO:0.98 DL:1 AO[]:0.98	AO:0.18 DL:6 AO[]:0.25	AO:0.97 DL:1 AO[]:0.98
S9234	19	228-(5597)	AO:0.35 DL:2 AO[]:0.39	AO:0.99 DL:1 AO[]:0.99	AO:0.21 DL:3 AO[]:0.3	AO:0.97 DL:1 AO[]:0.99	AO:0.15 DL:6 AO[]:0.21	AO:0.97 DL:1 AO[]:0.99
S15850	14	597-(9772)	AO:0.3 DL:2 AO[]:0.36	AO:0.99 DL:1 AO[]:0.99	AO:0.20 DL:3 AO[]:0.29	AO:0.99 DL:1 AO[]:0.99	AO:0.18 DL:5 AO[]:0.25	AO:0.98 DL:1 AO[]:0.99

FD-transitions. As circuit area does not only depend on the number of minterms but also on the specific minterms and don't cares [50], same detection latency (same number of *FD*-transitions) may also result in different area overheads.

- Fig. 12: All combinations have fairly good fault coverage. Good fault coverage can be achieved because if at least one *FD*-transition for the fault is present in the detector, the fault is covered.

For some other ISACS benchmarks, we report in Table 3 the worst (i.e., highest) and best (lowest) values of area overheads for 1, 2 and 3 combinations of unmeasurable lines. The detection latencies corresponding to those combinations (of measurement limitation) are also reported. Following that, for each of these detection latencies we determine area overhead for the online tester. The table also compares area overhead of the proposed scheme with the ones reported in [35,34] for similar detection latencies. It may be noted that we have not explicitly reported fault coverage because in all cases reported in the table, high coverage could be achieved. We have mainly dealt with area overhead and detection latency because they are the most sensitive parameters.

Columns 1-3 provide information about the circuit. Columns 4,6 and 8 correspond to the lowest (best) area overhead obtained among all combinations of 1, 2 and 3 unmeasured lines, respectively. Columns 5,7 and 9 correspond to the (worst) highest area overhead obtained among all combinations of 1, 2 and 3 unmeasured lines, respectively. Corresponding to each circuit (i.e., rows 4–10) we have reported values regarding, (i) AO: Area overhead of the *FN*-detector for the corresponding combination of unmeasured lines (ii) DL: detection latency due to the particular combination of unmeasured lines and (iii) Area overhead for the *FN*-detector designed using the scheme of [35,34] to achieve the particular detection latency; given a detection latency these schemes eliminate *FD*-transitions randomly such that the given latency is maintained. For example, row 4-column 4, represents the single input line in circuit s27 (NSF block), whose unmeasurability gives the best area overhead. The first element "AO:2.41" states that best case area overhead is 2.41 for the corresponding single line being unmeasured. The second element "DL:2" states that for the corresponding single line being unmeasured we get detection latency as 2. The third element "AO[]:2.45" states that for detection latency 2, the area overhead of the *FN*-detector designed using the scheme [35,34] is 2.45. The following conclusions can be derived.

- Trends are similar to the case of s1488.
- The ranges between best case area overhead and worst case area overhead are fairly high. Different input lines have different impact on transforming *FD*-transitions to non *FD*-transitions. This range can be utilized as a design parameter to tradeoff area overhead versus detection latency.
- For a given detection latency area overhead for the proposed scheme is lower compared to that of [35,34]. The scheme of [35,34] randomly eliminates *FD*-transitions while the proposed scheme performs this elimination by not measuring some input lines. Not measuring some input lines of the NSF implies that they are not tapped by the *FN*-detector. This reduces the fanouts of those input lines, resulting in less buffering (drivers) and hence lower area.

6. Conclusion and discussion

The present paper proposed a DES based technique for OLT of digital circuits with measurement limitation. The scheme illustrated how measurement imitation can be used as a tradeoff parameter to minimize area overhead at the cost of detection latency. Experimentally it was verified that measurement limitation do not have high impact on fault coverage. Further, it was also found that for a given detection latency, area overhead of the proposed scheme is lower compared to other similar schemes reported in the literature. Measurement limitation implies less lines to be tapped by the tester from the CUT. This lowers the number of driving buffers, thereby resulting in lower area overheads.

Theoretically speaking, the methodology can design on-line fault detectors for any digital circuit and any combination of measurement limitations. The computational complexity of constructing the on-line fault detector has been analyzed in this paper and has been shown that the input size of the CUT has a major role to play in the time required for the construction. So, the runtime complexity of the CAD tool developed based on the current methodology may reach impractical limits, typically for large circuits. This is because of the fact that in such cases generation of OBDDs itself becomes too complex. To address the issue, we need to enhance the scalability of the DES based OLT scheme.

References

[1] S.-I. Ao, M. Amouzegar, B.B. Rieger, Intelligent Automation and Systems Engineering, vol. 103, Springer Science and Business Media, 2011.

[2] G. Dimitrakopoulous, P. Demestichas, Intelligent transportation systems, IEEE Vehicular Technol. Mag. 5 (2010) 77–84.

[3] R. Jegadeeshwaran, V. Sugumaran, Brake fault diagnosis using clonal selection classification algorithm (CSCA)–a statistical learning approach, Eng. Sci. Technol., Int. J. 18 (1) (2015) 14–23. Elsevier.

[4] V. Indira, R. Vasanthakumari, R. Jegadeeshwaran, V. Sugumaran, Determination of minimum sample size for fault diagnosis of automobile hydraulic brake system using power analysis, Eng. Sci. Technol., Int. J. 18 (1) (2015) 59–69. Elsevier.

[5] O. Khan, S. Kundu, Hardware/software codesign architecture for online testing in chip multiprocessors, IEEE Trans. Depend. Secure Comput. 8 (5) (2011) 714–727.

[6] J. Cui, S. Wang, A model-based online fault detection and diagnosis strategy for centrifugal chiller systems, Int. J. Therm. Sci. 44 (10) (2005) 986–999. Elsevier.

[7] A. Ramirez-Trevino, E. Ruiz-Beltran, I. Rivera-Rangel, E. Lopez-Mellado, Online fault diagnosis of discrete event systems. A petri net-based approach, IEEE Trans. Autom. Sci. Eng. 1 (4) (2007) 31–39.

[8] A. Satoh, T. Sugawara, N. Homma, T. Aoki, High-performance concurrent error detection scheme for AES hardware, in: International Workshop on Cryptographic Hardware and Embedded Systems, 2008, pp. 100–112.

[9] D.M. Shalev, J. Tiran, Condition-based fault tree analysis (CBFTA): a new method for improved fault tree analysis (FTA), reliability and safety calculations, Reliab. Eng. Syst. Safety 92 (9) (2007) 1231–1241. Elsevier.

[10] Q. Yao, J. Wang, G. Zhang, A fault diagnosis expert system based on aircraft parameters, in: Web Information System and Application Conference, 2015, pp. 314–317.

[11] V. Muralidharan, V. Sugumaran, V. Indira, Fault diagnosis of monoblock centrifugal pump using SVM, Eng. Sci. Technol., Int. J. 17 (3) (2014) 152–157. Elsevier.

[12] N. Sakthivel, B.B. Nair, M. Elangovan, V. Sugumaran, S. Saravanmurugan, Comparison of dimensionality reduction techniques for the fault diagnosis of mono block centrifugal pump using vibration signals, Eng. Sci. Technol., Int. J. 17 (1) (2014) 30–38. Elsevier.

[13] A. Benveniste, E. Fabre, S. Haar, C. Jard, Diagnosis of asynchronous discrete-event systems: a net unfolding approach, IEEE Trans. Autom. Control 48 (5) (2003) 714–727.

[14] S.H. Zad, R.H. Kwong, W.M. Wonham, Fault diagnosis in discrete-event systems: framework and model reduction, IEEE Trans. Autom. Control 48 (7) (2003) 1199–1212.

[15] C.B. Dutta, U. Biswas, Failure diagnosis in real time stochastic discrete event systems, Eng. Sci. Technol., Int. J. 18 (4) (2015) 616–633. Elsevier.

[16] J. Chen, R.J. Patton, Robust Model-based Fault Diagnosis for Dynamic Systems, vol. 3, Springer Science and Business Media, 2012.

[17] S. Narasimhan, G. Biswas, Model-based diagnosis of hybrid systems, IEEE Trans. Syst. Man Cybern., Part A 37 (3) (2007) 348–361.

[18] P. Bhowal, D. Sarkar, S. Mukhopadhyay, A. Basu, Fault diagnosis in discrete time hybrid systems–a case study, Inf. Sci. 177 (5) (2007) 1290–1308. Elsevier.

[19] M. Krysander, E. Frisk, Sensor placement for fault diagnosis, IEEE Trans. Syst. Man Cybern., Part A: Syst. Humans 38 (6) (2008) 1398–1410.

[20] K. Tada, S. Kawamoto, N. Yamada, T. Kimura, M. Iwahashi, T. Tadaumi, K. Kato, Sensor placement optimization in on-site photovoltaic module inspection robot for fast and robust failure detection, in: Photovoltaic Specialist Conference, 2015, pp. 1–3.

[21] P. Hongxia, W. Xiuye, Optimal placement of sensor in gearbox fault diagnosis based on VPSO, in: International Conference on Natural Computation, 2010, pp. 3383–3387.

[22] M. Bushnell, V. Agrawal, Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits, vol. 17, Springer Science and Business Media, 2006.

[23] G.E. Moore, Cramming more components onto integrated circuits, Proc. IEEE 86 (1) (1998) 82–85.

[24] D. Hodges, H. Jackson, R. Saleh, Analysis and design of digital integrated circuits, in: Deep Submicron Technology (Special Indian Edition), Tata McGraw-Hill Education, 2005.

[25] J. Rueter, S. Rehm, J.-W. Mohr, F. Hensel, Automatic test equipment, US Patent App. 14/259, 043, 2014.

[26] P. Moorthy, S.S. Bharathy, An efficient test pattern generator for high fault coverage in built-in-self-test applications, in: International Conference on Computing, Communications and Networking Technologies, 2013, pp. 1–4.

[27] M. Favalli, C. Metra, Online testing approach for very deep-submicron ICs, IEEE Des. Test Comput. (2002) 16–23.

[28] M. Nicolaidis, Y. Zorian, On-line testing for VLSI – a compendium of approaches, J. Electron. Test. 12 (1) (1998) 7–20. Springer.

[29] D. Gizopoulos, M. Psarakis, S.V. Adve, P. Ramachandran, S.K.S. Hari, D. Sorin, A. Meixner, A. Biswas, X. Vera, Architectures for online error detection and recovery in multicore processors, in: Design, Automation and Test in Europe Conference and Exhibition, 2011, pp. 1–6.

[30] Y.-Y. Chen, Concurrent detection of control flow errors by hybrid signature monitoring, IEEE Trans. Comput. 10 (54) (2005) 1298–1313.

[31] L. Tan, Y. Tan, J. Xu, CFEDR: control-flow error detection and recovery using encoded signatures monitoring, in: International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, 2013, pp. 25–32.

[32] W. Hong, R. Modugu, M. Choi, Efficient online self-checking modulo 2n̂+ 1 multiplier design, IEEE Trans. Comput. 60 (9) (2011) 1354–1365.

[33] D. Vasudevan, P. Lala, A technique for modular design of self-checking carry-select adder, in: International Symposium on Defect and Fault Tolerance in VLSI Systems, 2005, pp. 325–333.

[34] P. Drineas, Y. Makris, SPaRre: selective partial replication for concurrent fault-detection in fsms, IEEE Trans. Instrum. Measure. 52 (6) (2003) 1729–1737.

[35] S. Biswas, S. Mukhopadhyay, A. Patra, A formal approach to on-line monitoring of digital VLSI circuits: theory, design and implementation, J. Electron. Test. 21 (5) (2005) 503–537. Springer.

[36] S. Biswas, P. Srikanth, R. Jha, S. Mukhopadhyay, A. Patra, D. Sarkar, On-line testing of digital circuits for n-detect and bridging fault models, in: Asian Test Symposium, 2005, pp. 88–93.

[37] Y. Balasubrahamanyam, G.L. Chowdary, T.J.V.S. Subrahmanyam, A novel low power pattern generation technique for concurrent BIST architecture, Int. J. Comput. Technol. Appl. 3 (2) (2012) 561–565.

[38] I. Voyiatzis, A. Paschalis, D. Gizopoulos, C. Halatsis, F.S. Makri, M. Hatzimihail, An input vector monitoring concurrent BIST architecture based on a precomputed test set, IEEE Trans. Comput. 57 (8) (2008) 1012–1022.

[39] I. Voyiatzis, C. Efstathiou, C. Sgouropoulou, Symmetric transparent online BIST for arrays of word-organized rams, in: International Conference on Design and Technology of Integrated Systems in Nanoscale Era, 2013, pp. 122–127.

[40] P.K. Biswal, S. Biswas, A binary decision diagram based on-line testing of digital VLSI circuits for feedback bridging faults, Microelectron. J. 46 (7) (2015) 598–616. Elsevier.

[41] A.A. Kumar, Fundamentals of Digital Circuits, PHI Learning Pvt. Ltd., 2014.

[42] R.E. Bryant, Graph-based algorithms for boolean function manipulation, IEEE Trans. Comput. 100 (8) (1986) 677–691.

[43] R. Leveugle, G. Saucier, Optimized synthesis of concurrently checked controllers, IEEE Trans. Comput. 39 (4) (1990) 419–425.

[44] S.H. Robinson, J.P. Shen, Direct methods for synthesis of self-monitoring state machines, in: International Symposium on Fault-Tolerant Computing, 1992, pp. 306–315.

[45] N. Das, P. Roy, H. Rahaman, On line testing of single feedback bridging fault in cluster based FPGA by using asynchronous element, in: International On-Line Testing Symposium, 2008, pp. 190–191.

[46] S. Biswas, S. Mukhopadhyay, A. Patra, D. Sarkar, Unified technique for on-line testing of digital circuits: delay and stuck-at fault models, J. Circuits Syst. Comput. 17 (6) (2008) 1069–1089.

[47] N. Kettle, Anytime Algorithms for ROBDD Symmetry Detection and Approximation, Kent University, 2008. Ph.D. thesis.

[48] C. Meinel, T. Theobald, Algorithms and Data Structures in VLSI Design: OBDD-Foundations and Applications, Springer Science & Business Media, 2012.

[49] F. Brglez, D. Bryan, K. Kozminski, Combinational profiles of sequential benchmark circuits, in: International Symposium on Circuits and Systems, vol. 3, 1989, pp. 1929–1934.

[50] Z. Kohavi, N.K. Jha, Switching and Finite Automata Theory, Cambridge University Press, 2009.