Diagnosis of Single Stuck-at Faults and Multiple Timing Faults in Scan Chains

James Chien-Mo Li, *Member, IEEE*

***Abstract*—A diagnosis technique to locate single stuck-at faults and multiple timing faults in scan chains is presented. This technique applies single excitation (SE) patterns, in which only one bit is flipped in the presence of multiple faults. With SE patterns, the problem of unknown values in scan chains is eliminated. The diagnosis result is therefore deterministic, not probabilistic. In addition to the first fault, this technique also diagnoses the remaining timing faults by applying multiple excitation patterns. Experiments on benchmark circuits show that average diagnosis resolutions are mostly less than five, even for the tenth fault in the scan chain.**

***Index Terms*—Automatic test pattern generators (ATPG), fault diagnosis, scan chain.**

I. Introduction

FULL-SCAN digital circuits can be partitioned into two components: the scan chains and the combinational logic. Although the former can take up as much as 30% of the silicon area [1], there are fewer diagnosis techniques available for the scan chains than for the combinational logic. Diagnosis for the combinational logic has long been extensively studied for single stuck-at faults (SSF), multiple stuck-at faults, and delay faults [2]–[4]. By contrast, current commercial tools have limited capability to diagnose single stuck-at faults in the scan chains, let alone multiple timing faults. Recently, however, many real chips with faults in their scan chains have been reported by the industry [5]–[9]. Chips with faulty scan chains can cause serious yield loss problems in volume production so diagnosis of scan chain faults is receiving more and more attention.

The first requirement for a scan chain diagnosis technique is the ability to handle multiple faults. Traditionally, SSF model is assumed by most diagnosis tools and automatic test pattern generators (ATPGs). However, it is shown by experiment that SSF model is effective for detection but may not be very precise for diagnosis [10], [11]. One possible explanation for multiple faults is that defects tend to cluster together rather than scatter uniformly on the wafer [12]. The second important requirement for a good scan chain diagnosis technique is the ability to diagnose not only stuck-at faults but also timing faults. Timing faults include slow-to-rise, slow-to-fall, fast-to-rise, and fast-to-fall faults [13], [14]. One reason for including timing faults is that, in nanometer technology, the timing of scan cells can be susceptible to signal integrity problems such as crosstalk, IR drop, and ground bounce [7]. These problems can happen to many scan cells and therefore are better modeled by multiple timing faults rather than by single stuck-at faults. Real diagnosis cases of multiple timing faults in scan chains have been published in [7].

Manuscript received October 12, 2004; revised December 17, 2004. This work was supported in part by the National Science Council under Grant NSC 93-2215-E-002-030.

The author is with the Electrical Engineering Department and the Graduate Institute of Electronic Engineering, National Taiwan University, Taipei, Taiwan 106, R.O.C (e-mail: cmli@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TVLSI.2005.848800

TABLE I
Comparison of Diagnosis Resolutions

Circuit	SA0	STF	FTR
Industry, G=430K L=410 [14]	36	28	28
s15850, G=10K, L=534	7	7	7
s38584, G=20K, L=1,426	12	12	13
B19, G=230K L=409	11	15	20

Based on the preceding reasons, a software technique to diagnose single stuck-at faults and multiple timing faults in scan chains is presented. The technique has two parts. In the first, the fault type and the number of faults are determined. In the second, the diagnosis patterns are generated to locate faults in faulty chains. The first feature of this technique is diversified fault models. Six types of fault models are considered: stuck-at one (SA1), stuck-at zero (SA0), slow-to-rise (STR), slow-to-fall (STF), fast-to-rise (FTR), and fast-to-fall (FTF) faults. Another feature is that it provides fine diagnosis resolution (DR) even in the presence of multiple faults. This is achieved by applying single excitation (SE) patterns, in which only one bit can be flipped by the first fault. With SE patterns, the diagnosis patterns can be generated by widely available single stuck-at fault ATPG tools. This eliminates the need for a customized diagnosis pattern generator. Also, the SE patterns avoid the trouble of simulations with unknown values so the diagnosis results are deterministic, not probabilistic. The idea of SE patterns can be extended to multiple excitation (ME) patterns for diagnosing the remaining timing faults in the scan chains.

Before the technical details, three issues regarding the proposed technique deserve mention here. The first issue is whether dedicated patterns are necessary for diagnosis. Table I compares the worst-case DRs of the proposed technique with a previous technique [14]. The industry design, which has 430 K gates (G) and 22 K scan cells in 54 scan chains, is diagnosed by regular ATPG pattern. The particular chain under diagnosis has 410 (L) scan cells. The ISCAS'89 (s15850, s38584) and the ITC'99T (B19) benchmark circuits are diagnosed by our dedicated diagnosis patterns. Scan cells in each ISCAS circuit are stitched into one single chain. The 6.6-K scan cells in B19, which is modified from the 386 and the Vipor processors, are stitched into 16 chains. The particular scan chain under diagnosis has 409 scan cells. In the table, smaller DRs mean fewer suspicious faulty scan cells in a row and therefore better diagnosis results. Our

1063-8210/$20.00 © 2005 IEEE

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

experimental results show that the proposed diagnosis patterns provide finer DRs than regular ATPG patterns. Moreover, our proposed diagnosis patterns are applicable to multiple faults, while results from [14] are only for single fault diagnosis. Besides this experiment, case study in [8] also shows that traditional diagnosis is not good enough for timing faults in scan chains.

Second, the proposed technique generates one diagnosis pattern for every scan cell. To avoid too many diagnosis patterns, the faulty chains should be identified before pattern generation—that is, the circuits should be tested by the methods proposed in Section III first. For the B19 example, only 409 diagnosis patterns are needed for the particular faulty scan chain. Compared with regular ATPG patterns (10 K patterns, 90% fault coverage), the dedicated diagnosis patterns are much shorter. The proposed technique is especially useful when traditional diagnosis using regular ATPG patterns fails to produce fine DRs due to limited tester memory. Finally, to get very fine DR, this technique requires sequential ATPG, which can potentially require long CPU time and much memory space. To alleviate this problem, our diagnosis technique is divided into two procedures: a combinational diagnosis—coarse resolution—followed by a sequential diagnosis—fine resolution. The former quickly identifies the bounds of the faults so that the computation load of the latter is minimized. In addition, our technique loads the scan chains before running sequential ATPG so the computation for finding initial state is reduced. The B19 experiment is finished in less than 4000 s of CPU time, which demonstrates the feasibility of this technique on large designs.

The organization of the paper is as follows. Section II introduces some basic terms and background knowledge. Section III shows how to determine the fault type and the number of faults. Sections IV and V present the diagnosis techniques for the first and the remaining faults, respectively. Section VI discusses some issues related to the technique and the last section summarizes this paper.

II. Background

A. Scan Chain Fault Models

The scan cells are indexed in descending order, from the scan input (SI) to the scan output (SO). The length L of the scan chain is the total number of scan cells in the chain. The example scan chain in Fig. 1 has five scan cells $(L = 5)$, indexed from four to zero. For a given scan cell i, the cells that are indexed higher than i are called the *upstream cells* of cell i. The cells that are indexed lower than i are called the *downstream cells* of cell i. In Fig. 1, the cells {2,1,0} are downstream cells of cell 3. Cell 4 is the upstream cell of cell 3. The preceding definitions follow the convention proposed in past research such as [14]. The *first fault* f_1 in a faulty scan chain is the faulty cell that has the highest index. The nth fault f_n is the faulty cell that has the nth highest index.

The proposed technique gives an *upper bound* and a *lower bound* of the interval of consecutive cells that contains the faulty cell. The upper bound is the highest index of a group of consecutive scan cells. The lower bound is the lowest index of a group of consecutive scan cells. *Diagnosis resolution* is defined as the difference between the upper bound and the lower bound. The smaller is the DR, the more precise is the diagnosis result. Let j denote the scan clock cycle number and let $g(i, j)$ represent the good value of scan cell i at cycle j. In shift mode, the content of scan cell i is updated by its immediate upstream cell every cycle. Fig. 1 shows the shifting of the scan chain for three cycles. Assuming no inversions between scan cells, the shift operation of a good chain is modeled by this equation: $g(i+1, j) = g(i, j+1)$. For clear explanation, it is assumed that no inversion is inserted between scan cells (see discussion).

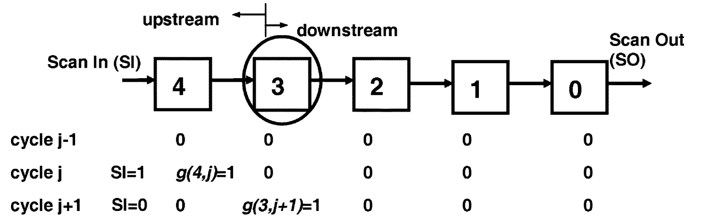

Fig. 1. Example scan chain.

TABLE II
EXCITATION CONDITIONS OF SIX TYPES OF FAULTS

Fault Type	Excitation Conditions	Fault Effect
SA0	$g(i,j) = 1$	$a(i,j) = 0$
SA1	$g(i,j) = 0$	$a(i,j) = 1$
STR	$g(i,j\text{-}1) = 0$ & $g(i,j) = 1$	$a(i,j) = 0$
STF	$g(i,j\text{-}1) = 1$ & $g(i,j) = 0$	$a(i,j) = 1$
FTR	$g(i,j) = 0$ & $g(i,j+1) = 1$	$a(i,j) = 1$
FTF	$g(i,j) = 1$ & $g(i,j+1) = 0$	$a(i,j) = 0$

Let $a(i, j)$ denote the actual content of scan cell i at scan clock cycle j. For a good scan chain, the actual content of every scan cell is always equal to its good content—that is, $a(i, j) = g(i, j)$. For a faulty chain, the actual content of a faulty cell is different from its good content when certain excitation conditions are met. Table II shows the excitation conditions of six types of faults. This table assumes cell i to be the faulty cell and cycle j to be the current clock cycle. A stuck-at zero fault in cell i is excited when a logic one is shifted into cell i. The actual content of cell i becomes logic zero. An STR fault in cell i is excited when cell i is expected to have a rising transition at the current cycle—that is, $g(i, j-1) = 0$ and $g(i, j) = 1$. The effect of the STR fault is that cell i remains zero instead of rising to one. An STF fault does the opposite thing; it remains one when it is expected to have a falling transition. An FTR fault in cell i is excited when cell i is expected to rise in the next cycle—that is, $g(i, j+1) = 1$ and $g(i, j) = 0$. The effect of the FTR fault is that cell i rises one cycle earlier than expected. An FTF fault is excited when cell i is expected to fall in the next cycle. The STR and STF faults are called the *slow-to* faults; the FTR and FTF faults are called the *fast-to* faults.

Causes of scan chain faults can be classified into three categories: the scan cells internal problems, the scan signal timing problems, and the clock timing problems. Examples of the first category include open defects, short-to-ground, and short-to-power defects. The defects affect only the shift mode, not the normal operation mode, because they are located on the scan signal path. In addition, defective transistors (such as

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

leakage and incorrect implants) and highly resistive bridging defects are shown to be culprits for the slow-to faults [16] and the fast-to faults [8], respectively. For the scan signal timing category, signal integrity problems such as crosstalk and ground bounce can cause timing faults [7]. On top of that, the slow-to faults can be caused by setup time violation due to routing congestion when the router gives priority to functional signals over the scan signals. The fast-to faults can be caused by hold-time violations due to insufficient buffers between scan cells. Clock skew problems, due to design errors or process defects, belong to the third category. The slow-to faults occur in cell i if the active clock edge arrives scan cell $i+1$ much later than it arrives scan cell i. The fast-to faults occur in cell i if the active clock edge arrives scan cell $i+1$ much earlier than it arrives scan cell i.

Based on the above discussions, it can be inferred that some slow-to faults (such as routing congestion induced faults) can be eliminated by slow speed shifting. Even if the slow-to faults are eliminated by slow speed shifting, diagnosing them is still important for failure analysis to identify the root causes of problems. Some fast-to faults (such as clock skew induced faults) may be avoided by changing the scan design but others (such as signal integrity induced faults) may not. Even if the fast-to faults can be eliminated by changing the scan design, diagnosing them is still essential for those chips that are already manufactured.

B. *Past Research*

Past research in the diagnosis of scan chains can be classified into two major categories: hardware solutions and software solutions. In the first category, Schafer proposed to add extra routings from one scan chain to a partner scan chain [17]. The contents of the scan chain under diagnosis can be observed by its partner scan chain. Edirisooriya proposed to insert XOR gates into the scan chains so that the contents of the scan cells can be flipped before shifting into the next scan cell [18]. Narayanan and Wu proposed to flip the contents of each scan cell by modifying the scan cell design [13], [19]. These hardware solutions require either custom scan cells or extra hardware. What is worse, the hardware solutions are not applicable to chips that are already designed and manufactured.

In the software category, Kundu proposed a sequential ATPG method to generate diagnosis patterns [1]. For every scan cell i in the chain, a sequence of initialization patterns that controls cell i to the desired value is generated. Kundu's idea is good for single stuck-at faults only. Stanley presented a diagnosis tool that does fault simulations for all latches in the scan chain [5]. A faulty latch is diagnosed if its score is higher than a certain threshold. The score of a fault represents the degree of similarity between a circuit's expected faulty outputs and its actual outputs. Hirase presented an IDDQ diagnosis technique for faults in scan chains [20]. Scan input patterns that have one particular bit opposite a stuck-at value are shifted into the faulty chain. Guo proposed a three-step diagnosis procedure [6], [14]. In the first step, the faulty chain and the fault type is determined by applying three fixed SI patterns. In the second step, the upper bound and the lower bound of the faulty cell is determined by logic simulations with unknown values in all scan cells of the faulty chain. In the third step, fault simulations are performed to obtain the expected faulty outputs. Guo's technique is applicable to single stuck-at faults as well as timing faults. Huang proposed a probabilistic model for intermittent timing faults in scan chains [7]. Huang's technique handles multiple faults by ranking the probability of a group of candidate faults. Neither Huang's nor Guo's technique gives deterministic diagnosis results in the presence of multiple faults in scan chains.

TABLE III
SE PATTERNS OF LENGTH $2L$

Fault Type	Tail (length L)	Head (length L)
SA0	$00000...0$	$\underline{1}0000...0$
SA1	$11111...1$	$\underline{0}1111...1$
STR	$00000...0$	$\underline{1}0(...0...1)_{L-2}$
FTR	$_{L-1}(...0...1)1$	$\underline{0}1111...1$
STF	$11111...1$	$\underline{0}1(...1...0)_{L-2}$
FTF	$_{L-1}(...1...0)0$	$\underline{1}0000...0$

C. *SE Patterns*

SE patterns have been used to diagnose single scan chain faults [24]. *Excitation patterns* for a given type of fault are SI patterns that, when shifted into the scan chain, cause excitations in the faulty scan cells. For example, the pattern $\{\underline{1}01\underline{1}0\}$ is an excitation pattern for the STR fault. (The rightmost bit of a stream of SI pattern is shifted into the chain first.) The underlined bits, which will be flipped after the fault excitation, are the *sensitive bits* (SB). Scan input patterns that cause no excitation for a given type of fault are called *nonexcitation patterns*. For example, the pattern $\{00\,000\}$ is a nonexcitation pattern for the STR fault. SE patterns are excitation patterns that, after flipping of the only sensitive bit, become nonexcitation patterns. For example, $\{0\underline{1}000\}$ is an SE pattern for the STR fault because it becomes $\{00\,000\}$ after the fault excitation. It follows from the definition that there is one and only one sensitive bit in an SE pattern. In this research, SE patterns of length $2L$ are applied for diagnosis purpose. Table III lists SE patterns of length $2L$ for six types of faults. The SE patterns can be divided into two parts: the head portion and the tail portion. Each portion is of equal length L. The sensitive bits (underlined) are always placed at the end (leftmost) of the head portion. The reason will be clear in the coming section.

Only one SE pattern is available for the SA0 and the SA1 faults, respectively. More than one SE pattern is available for the other types of faults. In the STR head portion, the notation $(\ldots 0 \ldots 1)_{L-2}$ represents a bit sequence of length $L-2$. This sequence can be all zeros, all ones, or a sequence of ones followed by a sequence of zeros. Note that the bits shift to the right so the one sequence enters the scan chain before the zero sequence. The STR tail portion is a sequence of L zeros. There are a total of $(L-1)$ distinct SE patterns available for the STR fault. Similarly, there are a total of $(L-1)$ SE patterns for the STF fault. The FTR head portion consists of $(L-1)$ ones followed by a zero. In the FTR tail portion, the notation $_{L-1}(\ldots 0 \ldots 1)$ represents a sequence of length $(L-1)$. The subscript $(L-1)$ is written on the left side of the parenthesis for clarity. There are L distinct SE patterns available for the FTR fault and the FTF fault, respectively.

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

TABLE IV
TEST PATTERNS TO DETERMINE FAULT TYPE AND NUMBER OF FAULTS

Pattern	Pattern 1 (L)	Pattern 2 (L)
Scan in	11110000	00001111
Scan out of STR*	11**00**0000	00001111
Scan out of STF*	11110000	00**11**1111
Scan out of FTR*	1111**11**00	00001111
Scan out of FTF*	11110000	000011**00**11
Scan out of SA0	**0000**0000	0000**0000**
Scan out of SA1	1111**1111**	**1111**1111

* two faults assumed

III. Determining Fault Types and Number of Faults

To determine the fault type, Guo proposed to apply three test patterns [14]. These three test patterns cannot determine the number of faults in the presence of multiple timing faults. This paper proposes to apply two test patterns, listed in Table IV, to determine the fault type and the number of timing faults in a scan chain. Each pattern is of equal length L. The scan cells are divided into halves. In the first pattern, the downstream half is all zeros and the upstream half is all ones. The second pattern is the bitwise complement of the first pattern. On the tester, the scan chain is initialized by shifting the first (rightmost) bit of the first pattern for L times. Alternatively, the scan chain can be initialized by asserting the independent reset or preset signals if available. After the initialization, the first pattern is scanned in, followed by an immediate scan out without any system clocks (SCK). The same procedure is repeated for the second pattern.

The SOs of six types of faults are listed in Table IV with mismatched bits in bold. If there exist f STR faults, there will be f more zeros than expected in the scan out of the first pattern. In this table, two timing faults are shown for demonstration. If there exist f FTR faults, there will be f more ones than expected in the scan out of the first pattern. Similarly, STF faults and FTF faults are detected in the second pattern by f more ones and f more zeros than expected, respectively. The number of faults equals the number of bits flipped in the SOs. The maximum number of timing faults countable is half the length of scan chains—that is, $L/2$. The SA0 and SA1 faults can be identified by all-zero or all-one SO patterns, respectively. However, the number of stuck-at faults cannot be counted by this technique because the SOs of SA0 and SA1 faults are always all zeros or ones, respectively.

All chains in a circuit are tested by both patterns in Table IV. The scan chains that fail either pattern are faulty chains; those chains that pass both patterns are good. The faulty chains will be diagnosed one chain by one chain—that is, one scan *chain under diagnosis* (ChUD) at one time. The scan chains other than the ChUD are the scan *chains under no diagnosis (ChUNDs)*. The ChUD must be faulty and ChUNDs can be either good or faulty. Outputs of good ChUNDs are observed but outputs of faulty ChUNDs have to be masked.

IV. Diagnosis of the First Fault

Fig. 2 shows a flow chart to diagnose the first fault f_1. Given a netlist, a fault type, and an SE pattern, the SI and *primary input* (PI) patterns are generated by two *automatic diagnosis pattern generators* (ADPGs): the combinational (C-ADPG) and the sequential (S-ADPG). The former generates diagnosis patterns of coarse resolution within a short time; the latter generates diagnosis patterns of fine DRs at a cost of relatively long CPU time. The expected good primary outputs (PO_g) and expected good SOs (SO_g) are also generated by the ADPGs. The circuit under diagnosis (CUD) is then tested on the automatic test equipment (ATE) and the actually observed primary outputs (PO_a) and SOs (SO_a) are logged into a file. Finally, PO_a and SO_a are matched with PO_g and SO_g to obtain the final diagnosis results: the upper bound and the lower bound of the first fault, denoted as $U(f_1)$ and $L(f_1)$, respectively.

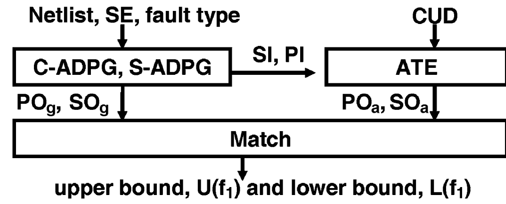

Fig. 2. Flow chart to diagnose f_1.

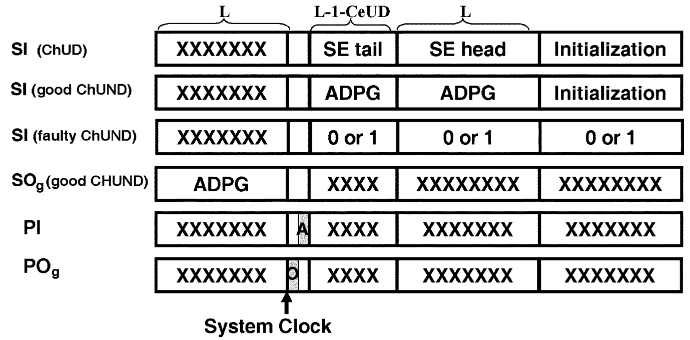

Fig. 3. C-ADPG pattern format.

A. C-ADPG

1) Combinational Diagnosis Procedure: This paper proposes a *combinational diagnosis procedure* (CDP) to diagnose faults in scan chains. The CDP is applied to one single scan cell each time a ChUD is loaded. That particular scan cell is called the *scan cell under diagnosis (CeUD)*. The CDP for a given CeUD is described as follows.

1) Initialize the ChUD by shifting in the first bit of the SE pattern for L times.
2) Shift in the SE pattern until the sensitive bit reaches the CeUD.
3) Apply a PI pattern.
4) Observe the primary outputs (PO_a). Record the PO_a in a file. No pass/fail decision is made immediately.
5) Pulse a SCK and shift out the scan chains. Observe the SOs (SO_a) of good ChUNDs. Mask the SOs of the other chains.

Fig. 3 shows the format of the C-ADPG patterns applied and observed during the CDP. For the STR and STF faults, the initialization pattern is obtained by replicating the first bit of SE patterns by L times. This initialization ensures that the head portion of the SE pattern is shifted into the scan chain without being changed by faults. Alternatively, the scan chains can be initialized by asserting the independent preset or reset signals if available. After the initialization, the head portion of the SE pattern is

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

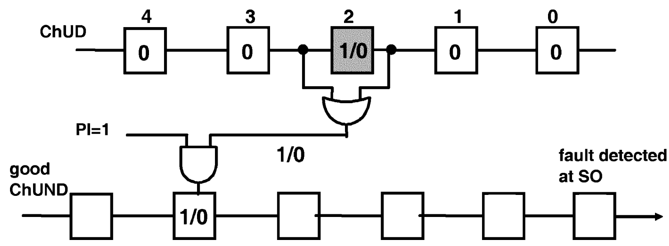

Fig. 4. Fault detected at SO of good ChUND.

then shifted in without observing any SO or PO. At this time, the sensitive bit is located at the most upstream scan cell. The ChUD is again shifted a specific number of times until the sensitive bit reaches the position of the CeUD. The number of shifts equals $L-1$ minus the index of the CeUD. At this time, the sensitive bit is located at the CeUD. Note that the number of shift cycles should be computed by the C-ADPG so the ATE does not have to keep track of the number of shifts when applying patterns. One PI pattern is applied—marked as "A"—and the corresponding primary outputs are observed—marked as "O." A SCK is pulsed to capture the responses into the scan chains. Finally, the scan chains are shifted out. The SOs of good ChUNDs are observed; the SOs of the other chains are masked.

Fig. 4 illustrates an example in which CDP detects a flipped sensitive bit at the SO of a good ChUND. The upper scan chain is the ChUD and the lower scan chain is a good ChUND. Suppose cell 2 is the CeUD and the fault type is STR. The SE pattern $\{00\underline{1}00\}$ is shifted into the chain and the sensitive bit is flipped by the fault in cell 2. The actual contents of scan cells are $\{00\,000\}$. In the figure, the number on the left of the slash sign represents the good value and the number on the right is the faulty value. By applying the $\mathrm{PI}=1$, the fault effect of the flipped sensitive bit is captured in the ChUND after on SCK. The fault effect can be observed at the SO of the good ChUND.

To ensure only one sensitive bit at a time, all-zero patterns or all-one patterns are applied to faulty ChUNDs. If the faulty ChUNDs have stuck-at zero faults, all-zero patterns are scanned in. If the faulty ChUNDs have stuck-at one faults, all-one patterns are scanned in. If the faulty ChUNDs have any of the four faults (STR, STF, FTR, FTF), either all-zero or all-one patterns are scanned in. This constraint ensures only one fault is excited at a time. On the contrary, the SI patterns to the good ChUNDs do not have to be constrained; they are specified by the C-ADPG.

2) C-ADPG Method: As a SE pattern is shifting along the ChUD, the sensitive bit is flipped as soon as it passes the fault. To locate the most upstream fault is to determine, as early as possible, the position where the sensitive bit is flipped. To *detect a flipped sensitive bit* is equivalent to detecting a stuck-at S fault in the cell that contains the sensitive bit, where S is opposite to the good value of the sensitive bit. Diagnosis patterns can be easily generated by a combinational SSF ATPG because all scan cells except the sensitive bit are fully specified. The steps of the C-ADPG for a given SE pattern are described as follows.

1) Right shift the SE pattern until the sensitive bit reaches the CeUD in the head portion. Initialize the scan cells of the ChUD to the head portion of the shifted SE pattern.
2) Initialize scan cells in faulty ChUNDs to all ones or zeros.

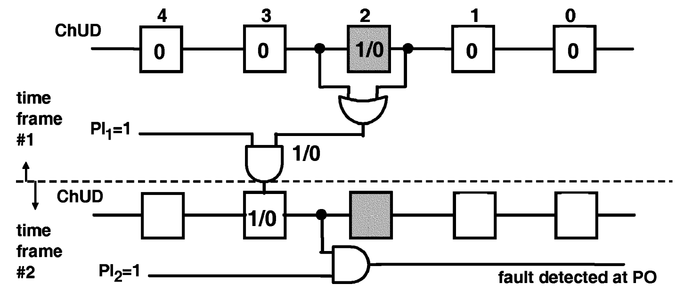

Fig. 5. Fault detected at PO.

3) Inject a stuck-at S fault at the output of CeUD; S is opposite to the good value of the sensitive bit.
4) Run combinational SSF ATPG to detect the fault. Allow observation only in PO and SO of good ChUNDs. Mask the SOs of the other chains.
5) If ATPG is successful, cell i is marked as C-observable.

For a given SE pattern, a *C-observable cell* is a cell for which the C-ADPG successfully generates patterns to detect a flipped sensitive bit at that cell. A *C-unobservable cell* is a cell for which the C-ADPG fails to generate patterns to detect a flipped sensitive bit at that cell. For a set of SE patterns, a scan cell is C-observable if it is C-observable by at least one pattern in the set; otherwise, it is C-unobservable. Scan cells are C-unobservable for one of the following reasons. The first is lack of propagation paths due to the constraints of SE patterns. The second is that scan cells of ChUNDs are forced to all ones or zeros. The third reason is due to the structure of the circuit.

B. S-ADPG

1) Sequential Diagnosis Procedure: To enhance the DR, the CDP unobservable cells must be diagnosed by the *sequential diagnosis procedure* (SDP). The SDP for a given CeUD is described as follows.

1) Initialize ChUD by shifting in the first bit of the SE pattern.
2) Keep shifting in the SE pattern until the sensitive bit reaches the CeUD.
3) Apply a PI pattern.
4) Observe primary outputs (PO_a) and pulse a SCK. Record the PO_a in a file. No pass/fail decision is made.
5) Repeat steps 3 and 4 for a specified number of times.
6) Shift out the scan chains. Observe SOs of good ChUNDs. Mask SOs of the other chains.

In Step 1, the ChUD is initialized in the same way as the CDP. In Step 2, the sensitive bit is shifted along the scan chain under diagnosis until it reaches the scan cell under diagnosis. At the same time, the all-one or all-zero patterns are shifted into faulty ChUNDs. This prevents the excitation of the faults in faulty ChUNDs. Scan input patterns to good ChUNDs are specified by the S-ADPG. In the third and fourth steps, the PIs are applied, the SCKs are pulsed, and the primary outputs are observed. Steps 3 and 4 are repeated for a certain number of times. Finally, the contents in good ChUNDs are scanned out and observed.

Fig. 5 illustrates an example in which the SDP detects a flipped sensitive bit at the primary outputs. Suppose cell 2 is the cell under diagnosis and the fault type is STR. Notice that

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

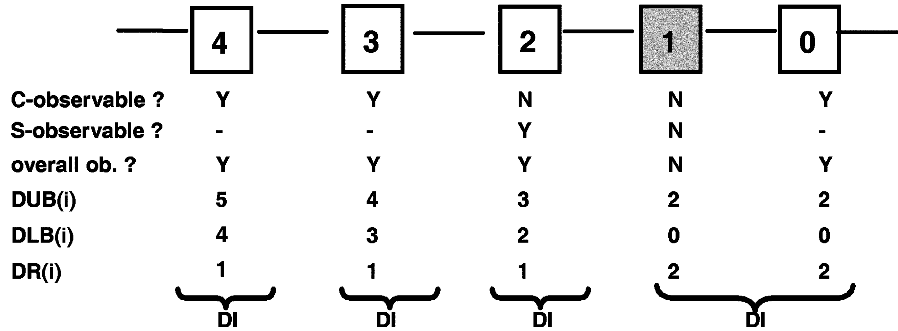

Fig. 6. DUB, DLB, and DR.

the upper chain and the lower chain are the same ChUD in different time frames. In the first time frame, the SE pattern $\{00\underline{1}00\}$ is shifted into the chain and the sensitive bit is flipped by the fault. This flipped sensitive bit cannot be detected given any PI_1. After one system clock, the fault effect of the flipped sensitive bit is captured by cell 3 in the second time frame. The fault effect can now be propagated from cell 3 to a primary output given a $\mathrm{PI}_2 = 1$.

2) S-ADPG Method: The following steps describe how the S-ADPG is carried out for a given SE pattern.

1) Right shift the SE pattern until the sensitive bit reaches the CeUD. Initialize the SIs of scan cells in the ChUD to the head portion of the shifted SE pattern.
2) Inject a stuck-at S fault at the SI of the CeUD. S is opposite to the good value of the sensitive bit.
3) Initialize the SIs of all scan cells in faulty ChUNDs to all ones or all zeros.
4) Assert the scan-enable signal (shift mode) and pulse one clock. By doing so, all the scan cells are initialized and the fault is injected into the CeUD.
5) De-assert the scan_enable (normal operation mode) and run sequential ATPG to detect the injected fault. Allow observation in the primary outputs.
6) After the last system clock, observe SOs of good ChUNDs. Mask the SOs of the other chains.
7) If ATPG is successful, the CeUD is S-observable.

For a given SE pattern, an *S-observable cell* is a cell for which the S-ADPG successfully generates patterns to detect the flipped sensitive bit in that cell. For a set of SE patterns, a cell is S-observable if it is S-observable by at least one SE pattern in the set; otherwise, it is S-unobservable. Note that a time limit should be given when generating patterns so that the S-ADPG is not trapped in an infinite loop in the presence of redundant or untestable fault.

One major difference between the sequential SSF ATPG and the S-ADPG is the time frame that contains the fault. Sequential SSF ATPG assumes that the fault is present in every time frame. The S-ADPG assumes that the fault is present only in the first time frame, not in any later time frame. This is because the scan cells are assumed to be faulty only in shift mode, not in normal operation mode. Based on this assumption, the fault is injected at the SI, not the output, of the cell under diagnosis.

3) Diagnosis Resolution: A fault can be diagnosed if it is detected by either the CDP or the SDP. Hence, a cell is *observable* if it is either C-observable or S-observable. A cell is *unobservable* if it is both C-unobservable and S-unobservable. The *diagnosis upper bound* (DUB) of cell i, $\mathrm{DUB}(i)$, is the index of the nearest upstream observable cell of cell i (excluding cell i itself). The *diagnosis lower bound* of cell i, $\mathrm{DLB}(i)$, is the index of the nearest downstream observable cell of cell i (including cell i itself). The DR of a cell i, $\mathrm{DR}(i)$, is equal to $\mathrm{DUB}(i)$ minus $\mathrm{DLB}(i)$. A group of consecutive cells that have the same DUB and DLB are in the same *diagnosis interval*. The DRs of cells in the same diagnosis intervals are equal. Fig. 6 shows an example chain under diagnosis. Cells 4, 3, and 0 are C-observable; the others are C-unobservable. After the S-ADPG, cell 2 becomes S-observable but cell 1 remains S-unobservable. Overall, cells 4, 3, 2, 0 are observable and cell 1 is unobservable (represented by a gray box). The DUB, DLB, and DR numbers are shown in the figure. The DUB of the most upstream cell is L by default. Cells 1 and 0 form a single diagnosis interval. Cells 4, 3, and 2 form three individual diagnosis intervals.

C. Match

After testing the CUD, the actually observed PO_a and SO_a are recorded in a file. The PO_a and SO_a are then compared with the good outputs, PO_g and SO_g. The *first mismatch cell* is the most upstream cell in which a mismatch occurs. The first mismatch cell is obtained from either the CDP or the SDP. The upper bound of the first fault $U(f_1)$ and the lower bound of the first fault $L(f_1)$ are the DUB and the DLB of the first mismatch cell, respectively. The *DR of the first fault* is the DR of the first mismatch cell. Take the scan chain in Fig. 6 for instance. If the first mismatch cell is cell 0, then the upper bound of the first fault is $\mathrm{DUB}(0) = 2$ and the lower bound of the first fault is $\mathrm{DLB}(0) = 0$. This means the fault can be between the output of cell 2 and the output of cell 0. The DR of the first fault is two.

D. Experimental Results

To demonstrate the effectiveness of the proposed technique, experiments are performed on ISCAS'89 and ITC'99 benchmark circuits of various sizes. A commercial tool that supports both the combinational and sequential pattern generation is used as the ATPG engine. Table V shows DRs of six types of faults. Each CUD has two rows. The first row (C) shows DRs obtained from the C-ADPG only. The second row (C+S) shows DRs obtained from the C-ADPG plus the S-ADPG—that is, a cell is observable if it is either C-observable or S-observable. In each cell of this table, a slash sign separates two DR numbers: the *average DR* and the *worst DR*. The average DR is obtained by taking the

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

TABLE V
DIAGNOSIS RESOLUTIONS (AVERAGE/WORST)

CUD	ADPG	SA0	SA1	STR	STF	FTR	FTF
s713	C	1.0 / 1	1.2 / 2	1.0 / 1	1.2 / 2	1.1 / 2	1.0 / 1
G393; L=19	C+S	1.0 / 1	1.2 / 2	1.0 / 1	1.2 / 2	1.1 / 2	1.0 / 1
s838	C	1.0 / 1	25 / 28	1.0 / 1	25 / 28	4.0 / 4	1.0 / 1
G=446; L=32	C+S	1.0 / 1	1.0 / 1	1.0 / 1	1.0 / 1	1.0 / 1	1.0 / 1
s5378	C	10.3 / 32	7.3 / 29	8.5 / 32	4.1 / 17	3.7 / 17	7.3 / 26
G=2,779; L=179	C+S	4.2 / 13	2.5 / 10	3.6 / 13	2.1 / 10	1.8 / 7	2.0 / 7
s9234	C	47.8 / 87	10.9 / 19	46.2 / 87	10.5 / 19	10.9 / 19	32.1 / 66
G=5,597; L=211	C+S	15.4 / 33	5.7 / 14	6.9 / 19	3.3 / 8	4.8 / 14	5.5 / 13
s15850	C	5.8 / 15	5.4 / 15	5.3 / 15	5.4 / 15	5.3 / 15	5.8 / 15
G=9,772; L=534	C+S	2.7 / 7	2.1 / 7	2.1 / 7	2.0 / 7	2.1 / 7	2.7 / 7
s38584	C	11.7 / 29	11.3 / 31	10.9 / 31	11.2 / 30	11.3 / 31	11.7 / 29
G=19,253; L=1,426	C+S	4.1 / 12	4.6 / 15	3.4 / 10	3.8 / 12	5.6 / 13	3.6 / 12
B19	C	6.8 / 23	8.6 / 20	7.6 / 23	7.9 / 20	8.6 / 20	6.8 / 23
G=231,320; L=409	C+S	3.6 / 11	6.5 / 20	3.4 / 11	5.2 / 15	4.5 / 20	3.6 / 11

average of $\mathrm{DR}(i)$ among all scan cells in a chain. The worst DR is the maximum among all $\mathrm{DR}(i)$ in a chain. The CUDs in the table are listed in increasing order of their sizes. Their numbers of combinational logic gates (G) and the length of scan chains (L) are shown for reference. All scan cells in every ISCAS circuit are stitched into one single scan chain. The 6.6 K scan cells in B19 are stitched into 16 chains; the particular chain under diagnosis has 409 scan cells. Three systems clocks are applied in the S-ADPG. All available SE patterns are exhaustively used except large CUDs, s15850, s38584, and B19.

The average DRs are less than five and worst DRs are less than fifteen in most cases. For small CUDs such as s713 and s838, their worst DRs are mostly less than two scan cells. For large CUD like s15850, the worst DR is 7, which is 1.3% of the chain length. For s38584, the worst DR is 15, which is only 1.1% of the chain length. DRs of the preceding circuits are pretty good for the physical failure analysis. For the largest CUD, B19, the worst DRs is 20, which is less than 5% of the chain length. However, there are some cases in which DRs are not ideal. For s5378, the worst DRs of STR and SA0 are 13, which is 7% of the total 179 cells. For s9234, the worst DR of SA0 is 33, which is about one seventh of the total 211 cells. There are mainly two reasons for the unsatisfactory DRs. S5378 has poor DRs because of the constraints of SE patterns. S9234 has poor DRs due to the structure of circuits. It has been shown that S9234 has quite a few redundant faults that are untestable [21]. A further analysis finds that about 6% of the faults in s9234 are redundant. This explains why the DRs of s9234 are not satisfactory. DRs of s9234 and s5378 can be improved by splitting the scan cells into more than one scan chain.

For the S-ADPG, one key factor that affects DRs is the number of SCKs. Table VI shows DRs of one, two, three, and four SCKs for STF fault. For the ISCAS circuits, DRs improve as the number of SCKs increases from one to three. However, DRs make little improvement above three SCKs. The ISCAS benchmark circuits are not very complicated designs so their *sequential depths* are shallow. The sequential depth of a circuit represents the maximum level of flip-flops (latches) that a fault effect must propagate through before reaching observation points [22]. For this diagnosis technique, observation points can be either primary outputs or SOs of good ChUNDs. The number of SCKs needed in the S-ADPG is dependent on the sequential depth of the CUD. The deeper the sequential depth is, the more SCKs are needed. The reason why DRs of B19 does not improve after two or more SCKs is because B19 has many good ChUNDs. The fault effects can be observed by good ChUNDs so it does not require many SCKs.

TABLE VI
DRS OF 1 TO 4 SYSTEM CLOCKS (AVERAGE/WORST)

CUD	1 SCK	2 SCK	3 SCK	4 SCK
s713	1.2 / 2	1.2 / 2	1.2 / 2	1.2 / 2
s838	1.0 / 1	1.0 / 1	1.0 / 1	1.0 / 1
s5378	2.4 / 10	2.2 / 10	2.1 / 10	2.1 / 10
s9234	7.3 / 19	5.4 / 18	3.3 / 8	3.3 / 8
s15850	3.5 / 15	2.5 / 7	2.3 / 7	2.2 / 7
s38584	6.1 / 16	4.7 / 13	3.8 / 12	2.1 / 12
B19	7.6 / 20	5.2 / 15	5.2 / 15	5.2 / 15

TABLE VII
CPU TIME (s)

CUD	C-ADPG	1 SCK	2 SCK	3 SCK	4SCK
s713	0.01	0.02	0.06	0.09	0.12
s838	0.02	2.59	11.92	14.73	19.95
s5378	0.33	1.79	2.62	5.02	11.09
s9234	0.41	24.41	123.48	182.35	393.84
s15850	2.64	91.00	345.69	894.60	1,655.79
s38584	10.00	497.97	1,267.67	2,422.10	3,122.91
B19	59.93	1539.68	3,187.21	3,456.96	3,885.30

Table VII compares the CPU time for the C-ADPG and the S-ADPG of different numbers of SCKs (same STF fault). The C-ADPG is performed on all faults in the chain. The S-ADPG is performed on C-unobservable faults only. The CPU time needed for the C-ADPG is significantly shorter than that of the S-ADPG. Although the C-ADPG does not produce very low DRs, it is still worth doing because it helps to reduce the S-ADPG time. For the ISCAS circuits, the CPU time of four SCKs can be up to ten times longer than that of one SCK. For B19, the reason why the CPU time does not increase much after two SCKs is because there are many good ChUNDs available. Fault effects can be captured by good ChUNDs without many SCKs. The B19 experiments are done within a reasonable amount of time, showing the feasibility of the technique on

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

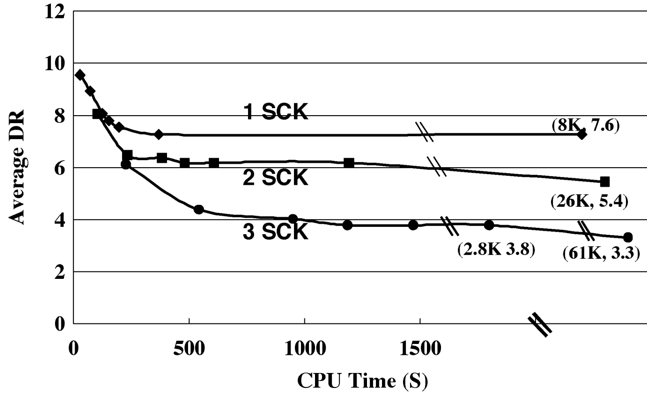

Fig. 7. Average DRs versus CPU time (s9234, STF).

large designs. The CPU time shown here is obtained for one SE pattern. If more than one SE pattern is used, the total CPU time needed is approximately proportional to the number of SE patterns.

Besides the number of SCKs, another important factor that affects DRs is the number of SE patterns. For a set of SE patterns, a cell is observable if it is observable by at least one SE pattern in the set. The more SE patterns are used in the ADPG, the lower DRs become. To compare the effects of the SCKs and the SE patterns on DRs, Fig. 7 shows average DRs versus the CPU time (s9234, STF). There are three curves representing one, two, and three SCKs, from top to bottom. Every curve has seven data points, representing 1, 2, 3, 4, 5, 10, and all $(L-1)$ SE patterns, from left to right. The first SE pattern chosen is all ones in the $(\ldots 1 \ldots 0)_{L-2}$ sequence in Table III. The second SE pattern chosen has a half zeros followed by a half ones in the $(\ldots 1 \ldots 0)_{L-2}$ sequence. The third pattern has all zeros in the $(\ldots 1 \ldots 0)_{L-2}$ sequence. The reason for choosing these three very different SE patterns is to increase the chance of getting different observable cells. It is seen from this figure that increasing the number of SCKs is very effective in improving DRs. On the contrary, increasing the number of SE patterns beyond three gives only marginal improvement in the DRs. Based on this figure, an efficient S-ADPG approach can be concluded. First, use an arbitrary SE pattern and one SCK. If DRs are not good enough, increase the number of SCKs instead of SE patterns, because the former is more effective than the latter given the same CPU time. Increase the number of SCKs until the number reaches the sequential depth of the CUD. If DRs are still not satisfactory, then add more SE patterns. Choose very different SE patterns so that the chance of getting different observable cells is high.

The DRs of s9234 and s5378 can be further improved by splitting the scan cells into two scan chains. Table VIII compares worst DRs of two versions: the one-chain version and the two-chain version. For the one-chain version, all scan cells in a CUD are chained into one single scan chain. For the two-chain version, scan cells are divided evenly into two scan chains. In this experiment, only one faulty chain is assumed at a time. It can be seen that worst DRs of the two-chain versions are much better than those of the one-chain versions. The worst DRs are equal to or better than five for all types of faults. This experiment shows that DRs of ISCAS circuits in Table V are pessimistic and can be greatly improved by increasing the number of scan chains.

TABLE VIII
WORST DRs OF S9234 AND S5378 (ONE CHAINS VERSUS TWO CHAINS)

fault	s5378 (1ch)	s5378 (2ch)	s9234 (1 ch)	s9234 (2 ch)
SA0	13	5	33	4
SA1	10	4	14	4
STR	13	4	19	3
STF	10	4	8	2
FTR	7	3	14	2
FTF	7	4	13	4

TABLE IX
DE AND ASE PATTERNS

fault type		Tail (L)	Head (L)
STR	DE	000000...0	110 (0...1...) L-3
	ASE	000000...0	100 (0...1...) L-3
FTR	DE	L-2 (0...1...) 10	0111111...1
	ASE	L-2 (0...1...) 11	0111111...1
STF	DE	111111...1	001 (1...0...) L-3
	ASE	111111...1	011 (1...0...) L-3
FTF	DE	L-2 (1...0...) 01	1000000...0
	ASE	L-2 (1...0...) 00	1000000...0

V. DIAGNOSIS OF REMAINING TIMING FAULTS

In this section, a modified technique is presented to diagnose remaining timing faults in scan chains. To diagnose the nth fault is to locate the upper bound and the lower bound of the nth fault, denoted as $U(f_n)$ and $L(f_n)$, respectively.

A. ME Patterns

Double excitation (DE) patterns are excitation patterns that, after flipping the sensitive bit by the first fault, become SE patterns. Table IX shows DE patterns of length $2L$. There are two consecutive sensitive bits in a DE pattern. The *first sensitive bit* (single underlined) can be flipped by the first fault; the *second sensitive bit* (double underlined) can be flipped by the second fault. After flipping the first sensitive bit, the DE pattern becomes its *associated single excitation (ASE)* pattern. Every DE patterns has exactly one ASE pattern. For instance, $\{00\,000\ \underline{11}011\}$ is a DE pattern for the STR fault. After flipping the first sensitive bit, it becomes $\{00\,000\ \underline{1}0011\}$, which is the ASE pattern of the DE pattern. The second sensitive bit of the DE pattern becomes the only sensitive bit of the ASE pattern. There are totally $(L-2)$ distinct DE patterns, and, hence, $(L-2)$ ASE patterns, available for the STR fault. It can be observed that, for the same fault type, the number of ASE patterns in Table IX is smaller by one than the number of SE patterns in Table III. It follows from this observation that diagnosing the second fault is more difficult than diagnosing the first fault because the former has one fewer available SE patterns than the latter.

Similarly, *triple excitation (TE)* patterns are excitation patterns that, after flipping the first and the second sensitive bits by f_1 and f_2, become SE patterns. ME patterns can be defined in the same way. TE patterns are used to diagnose the third fault, and the nth ME patterns are used to diagnose the nth fault. In general, there are $(L-n)$ distinct nth ME patterns for the STR

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

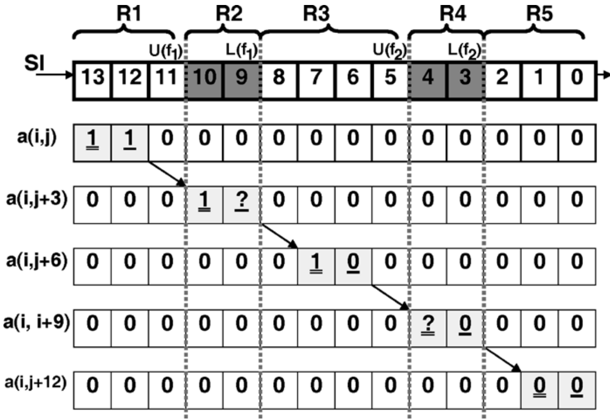

Fig. 8. Contents of scan cells as DE pattern shifts right.

and the STF faults, respectively. There are $(L-n+1)$ distinct nth ME patterns for the FTR and FTF faults, respectively.

Fig. 8 demonstrates how the contents of scan cells change when a DE pattern is shifting in a faulty scan chain. Suppose there are two STR faults in this chain. The upper bound $U(f_1)$ and the lower bound $L(f_1)$ of the first fault are cell 11 and cell 9, respectively; the upper bound $U(f_2)$ and the lower bound $L(f_2)$ of the second fault are cell 5 and cell 3. The chain can be divided into five regions (R1–R5) by these four bounds. The first sensitive bit is single underlined and the second sensitive bit is double underlined. The positions of the sensitive bits in the DE pattern are painted in light gray. In the first region (R1), the first sensitive bit is not flipped yet. In the second region (R2), the first sensitive bit is flipped by the first fault somewhere between $U(f_1)$ and $L(f_1)$ but the exact location is unknown. In the third region (R3), the first sensitive bit is flipped while the second sensitive bit remains unchanged. The flipped sensitive bit is detected because cells in R3 are observable. From this point on, the actual contents in the scan cells become the ASE pattern of the DE pattern. The second sensitive bit is flipped somewhere in region R4 and, finally, the second sensitive bit is observed to be flipped in region R5.

B. ADPG for Remaining Faults

Starting from region R3, diagnosing the second fault is virtually the same as diagnosing the first fault. Locating the second fault is to find out, as early as possible, the position where the sensitive bit of the ASE pattern is flipped. Therefore, detecting the second sensitive bit in a DE pattern is equivalent to detecting the sensitive bit in the ASE pattern of the DE pattern. To put it in another way, let $\mathrm{ADPG}_2(\mathrm{de})$ denotes the ADPG patterns for the second fault given a DE pattern de. Let $\mathrm{ADPG}_1(\mathrm{ase})$ denotes the ADPG patterns for the first fault given the pattern ase, which is the ASE pattern of de. The $\mathrm{ADPG}_2(\mathrm{de})$ is actually equal to $\mathrm{ADPG}_1(\mathrm{ase})$. Since the ADPG_1 patterns for the first fault are already generated in Section IV, the ADPG_2 patterns for the second fault need not to be generated again. All we need to do is to save the ADPG_1 patterns together with their corresponding SE patterns into a database. The ADPG_2 patterns can be easily retrieved from the same database.

The observability of the second fault is defined in the same way as the first fault. For a given DE pattern, a cell is f_2 *observable* if there exists one ADPG_2 pattern that detects a flipped second sensitive bit at that cell; otherwise, that cell is f_2 *unobservable*. For a set of DE patterns, a scan cell is f_2 observable if it is f_2 observable by at least one DE pattern in the set. A scan cell is f_2 *unobservable* if it is f_2 *unobservable* by all DE patterns in the set. The f_2 DUB of cell i $\mathrm{DUB}_2(i)$ is the index of the nearest upstream f_2 observable cell (excluding cell i itself). The f_2 *diagnosis lower bound* (DLB) of cell i $\mathrm{DLB}_2(i)$ is the index of the nearest downstream f_2 observable cell (including cell i itself). The f_2 DR of a cell i $\mathrm{DR}_2(i)$ is $\mathrm{DUB}_2(i)$ minus $\mathrm{DLB}_2(i)$. The numbers for the nth fault are defined in the same way. It can be proved that if cell i is f_1 unobservable, it must be f_2 unobservable. The reason is because the set of ASE patterns is a subset of the SE patterns. If there exists no ADPG_1 pattern to detect a flipped sensitive bit at cell i given any SE pattern, it is impossible to generate an ADPG_2 pattern given a subset of SE patterns. It follows that if a cell is f_n unobservable, then it must be f_{n+1} unobservable. The DR_{n+1} must be equal or worse than the DR_n.

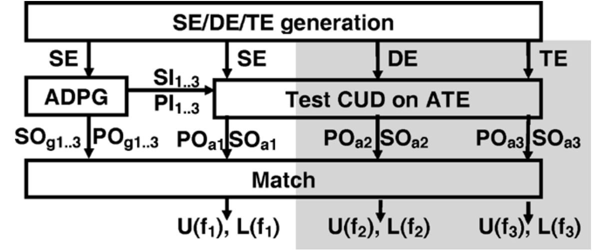

Fig. 9. Flow chart to diagnose the first three faults.

```
// first fault
n=1;
i = first mismatch cell between SOa1POa1 and SOg1POg1;
U(f1) = DUB1(i); L(f1) = DLB1(i);
// remaining faults
While ( n ≤ desired value) do
{
        n++;
        i = first mismatch cell between SOanPOan
              and SOgnPOgn, i≤L(fn-1) ;
        U(fn) = DUBn(i); L(fn) = DLBn(i);
}
```

Fig. 10. Match algorithm.

C. Match

Fig. 9 shows the flow to diagnose the first three faults. It is similar to Fig. 2 except for the highlighted parts. First, all SE, DE, TE patterns are exhaustively generated. The SE patterns are fed to the ADPG tools to generate the ADPG_1, ADPG_2, and ADPG_3 patterns. SE patterns together with their corresponding ADPG_1 patterns, including both SI patterns (SI_1) and PI patterns (PI_1), are applied to the CUD. The observed primary outputs and SO patterns are recorded as PO_{a1} and SO_{a1}, respectively. Then DE patterns together with the corresponding ADPG_2 patterns are applied to the CUD. The observed outputs are recorded as PO_{a2} and SO_{a2}. Similar things happen to TE patterns.

Fig. 10 shows the match algorithm for the multiple timing fault diagnosis. The first half of this algorithm, which shows

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

the match for the first fault, is the same as what is presented in Section IV. The first mismatch cell i is the most upstream cell that mismatch between the good outputs, $\mathrm{SO}_{g1}\mathrm{PO}_{g1}$, and actual outputs, $\mathrm{SO}_{a1}\mathrm{PO}_{a1}$, occurs. The upper bound and the lower bound of the first fault are $\mathrm{DUB}_1(i)$ and $\mathrm{DLB}_1(i)$, respectively. The second half of this match algorithm is for remaining faults. For the second fault, the first mismatch cell i is the most upstream cell that mismatch between $\mathrm{SO}_{g2}\mathrm{PO}_{g2}$ and $\mathrm{SO}_{a2}\mathrm{PO}_{a2}$ occurs. The index i must be smaller than or equal to $L(f_1)$ because the second fault must be in the downstream of the first fault. The equal sign is needed to deal with the case in which multiple timing faults in one diagnosis interval. Take Fig. 8 for example, it is possible that the second fault is located in the same region as the first fault (R2). If that happens, mismatches occur in cell $L(f_1)$ for both the first fault and the second fault. In this case, the bounds of the first and the second faults are the same. Therefore, the equal sign has to be included when finding the first mismatch cell.

Note that this match algorithm for multiple faults does not require extra storage space. Because the expected PO_{gn} and SO_{gn} of ADPG_n are retrieved from the same database as the ADPG_1 (see Section V-B), the size of the fault dictionary does not grow with the multiplicity of faults. This is an important advantage that enables diagnosis of multiple faults in the same scan chain.

D. Experimental Results

Experiments are performed on the ISCAS circuits for the first, second, third, fifth, and tenth faults. In the experiment, the number of SCKs in S-ADPG is three and all scan cells are stitched into one scan chain. Experimental results show that, for most CUDs and fault types, worst DRs of the tenth fault are the same as those of the first fault. Only two exceptions are observed. The worst DR of s838 degrades from one to two for the fifth FTR fault. The worst DR of s9234 degrades from 13 to 22 for the fifth FTF fault. This can be due to the redundant fault problem as mentioned earlier. The experimental results show that our technique is still very effective for the tenth timing fault in the scan chain.

VI. Discussions

A. Negative Polarity

If the number of inversions between the SI and cell i is even, then cell i is of *positive polarity*. If the number of inversions between the SI and cell i is odd, cell i is of *negative polarity*. A group of consecutive scan cells of negative polarity is called a *negative polarity interval* (NPI). A group of consecutive scan cells of positive polarity is called a *positive polarity interval* (PPI). Fig. 11 shows a scan chain that has four scan cells (8, 7, 6, 5) of negative polarity. They form a NPI and the other cells are of positive polarity. A *local single excitation* (LSE) pattern is a SE pattern within a specific polarity interval. When the SE pattern reaches the NPI, the locally complemented SE pattern becomes an LSE pattern in the NPI. The fault type of the LSE pattern is the complement of the original SE pattern. The complement fault types of SA0, STR, and FTR faults are SA1, STF, and FTF faults, respectively. LSE patterns still have the SE property in the specified polarity interval. In Fig. 11, a SE patterns for

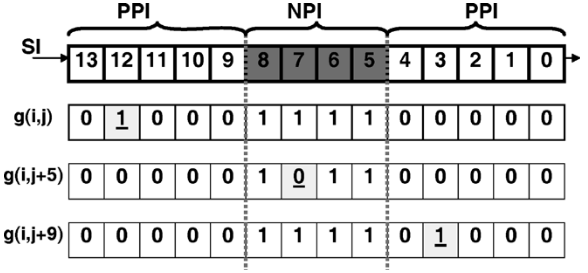

Fig. 11. LSE pattern and NPI.

SA0, $\{0\underline{1}00\ldots0\}$, is shifted into the scan chain. When the sensitive bit reaches the NPI, the pattern within the NPI becomes $\{1\underline{0}11\}$, which is an LSE pattern for SA1. If there is a single stuck-at one fault in the NPI, the sensitive bit in $\{1\underline{0}11\}$ will be flipped and this LSE pattern becomes $\{1111\}$, which is a nonexcitation pattern. Since LSE patterns still satisfy the SE condition within the given NPI, our diagnosis is still applicable. However, the diagnosed fault type depends on $U(f)$ and $L(f)$, of the diagnosed fault. There are three possible diagnosis outcomes. If $U(f)$ and $L(f)$ fall in the same PPI, then the diagnosed fault type is the same as that of the SE pattern. If $U(f)$ and $L(f)$ fall in the same NPI, then the diagnosed fault type has to be complemented. For the example in Fig. 11, if $U(f)$ and $L(f)$ are cell 7 and cell 6, respectively, then the fault type is SA1, not SA0. If $U(f)$ and $L(f)$ fall in different polarity intervals, then the fault type cannot be determined.

B. Limited Accessibility to Primary Outputs

For the high-density packaged CUT or the embedded cores, the primary outputs sometimes are not easily accessible. In the case of limited accessibility to primary outputs, an alternative solution is to add boundary scan, such as JTAG or P1500 wrappers, to the circuits. For our diagnosis technique to work correctly, one important point should be addressed when designing the boundary scan architecture. It is required that the boundary scan chain is able to shift independently of the internal scan chains. In this way, the primary outputs can be observed in the CDP or the SDP without affecting the contents of the internal scan chains.

C. Memory Requirement of S-ADPG

Although S-ADPG is implemented by sequential ATPG, it actually requires less memory than regular sequential ATPG. Because S-ADPG generates one pattern for a fault at a time, the memory requirement is smaller than regular sequential ATPG which generates patterns for many faults at a time. On top of that, S-ADPG loads all chains before generating sequential patterns so the initial state of the CUD is known. This requires far less memory than a regular sequential ATPG in which no initial state is assumed. Also, the number of SCKs in S-ADPG is specified by the user so the memory requirement can be controlled. Last, S-ADPG can be implemented by the "fast-sequential" ATPG mode, which is much efficient than traditional full sequential ATPG [23]. Table VIII shows that one S-ADPG on B19 circuit (230 K gate count) can be finished in less than 4000 s CPU time on a SUN work station equipped with 8-GB memory.

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.

TABLE X
TABLE OF ABBREVIATIONS

Abbreviation	meaning
ADPG	Automatic Diagnosis Pattern Generation
ASE	Associate Single Excitation
C-ADPG	Combinational ADPG
CDP	Combinational Diagnosis Procedure
CeUD	scan Cell Under Diagnosis
ChUD	Chain Under Diagnosis
ChUND	Chain Under No Diagnosis
CUD	Circuit Under Diagnosis
DE	Double Excitation
DLB	Diagnosis Lower Bond
DR	Diagnosis Resolution
DUB	Diagnosis Upper Bond
FTF	Fast-To-Fall
FTR	Fast-To-Rise
ME	Multiple Excitation
NPI	Negative Polarity Interval
PI	Primary Input
PPI	Positive Polarity Interval
PO	Primary Output
SA0	Stuck-At 0
SA1	Stuck-At 1
S-ADPG	Sequential ADPG
SDP	Sequential Diagnosis Procedure
SE	Single Excitation
SI	Scan Input
SO	Scan Output
STF	Slow-To-Fall
STR	Slow-To-Rise
TE	Triple Excitation

VII. SUMMARY

This paper presents a technique that diagnoses single stuck-at faults as well as multiple timing faults in scan chains. The diagnosis consists of two parts. The first part determines the fault type and the number of faults and the second part locates the faulty cell. The proposed technique has the advantage of handling scan chains with multiple faults because of the application of SE and ME patterns. This diagnosis technique provides deterministic results without the trouble of simulations with unknown values. For most benchmark circuits, the presented technique achieves average DRs finer than five scan cells, even for the tenth faults in the chain.

APPENDIX

See Table X.

REFERENCES

[1] S. Kundu, "Diagnosis scan chain faults," *IEEE Trans. Very Large Scale Intergr. (VLSI) Syst.*, vol. 2, no. 4, pp. 512–516, Dec. 1994.

[2] J. A. Waicukauski and E. Lindbloom, "Failure diagnosis of structured VLSI," *IEEE Des. Test. Comput.*, vol. 6, no. 4, pp. 49–60, Aug. 1989.

[3] M. Abramovici, M. A. Breuer, and A. D. Fridman, *Digital Systems Testing and Testability Design*. Piscataway, NJ: IEEE Press, 1995, ch. 12.

[4] N. K. Jha and S. Gupta, *Testing of Digital Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2003, ch. 10.

[5] K. Stanley, "High-accuracy flush-and-scan software diagnostic," *IEEE Des. Test. Comput.*, vol. 18, no. 6, pp. 56–62, Nov.–Dec. 2001.

[6] R. Guo and S. Venkataranman, "A new technique for scan chain failure diagnosis," in *Proc. Int. Symp. Testing and Failure Analysis*, 2002, pp. 723–732.

[7] Y. Huang, W.-T. Cheng, S. M. Reddy, C.-J. Hsieh, and Y.-T. Hung, "Statistical diagnosis for intermittent scan chain hold-time fault," in *Proc. IEEE Int. Test Conf.*, 2003, pp. 319–327.

[8] B. Kruseman, A. Majhi, C. Hora, S. Eichenberger, and J. Meirlevede, "Systematic defects in deep sub-micron technologies," in *Proc. IEEE Int. Test Conf.*, 2004, pp. 290–298.

[9] P. Song, F. Stellari, T. Xia, and A. Weger, "A novel scan chain diagnostics technique based on light emission from leakage current," in *Proc. IEEE Int. Test Conf.*, 2004, pp. 140–147.

[10] E. J. McCluskey and C. W. Tseng, "Stuck-fault versus actual defects," in *Proc. IEEE Int. Test Conf.*, 2000, pp. 3343–3356.

[11] J. C.-M. Li and E. J. McCluskey, "Diagnosis for sequence dependent chips," in *Proc. IEEE VLSI Test Symp.*, 2002, pp. 187–192.

[12] I. Koren, Z. Koren, and C. H. Stapper, "A statistical study of defect maps of large area VLSI ICs," *IEEE Trans. Very Large Scale Intergr. (VLSI) Syst.*, vol. 4, no. 2, pp. 249–256, Jun. 1996.

[13] Y. Wu, "Diagnosis of scan chain failures," in *Proc. Int. Symp. Defect and Fault Tolerance in VLSI Systems*, 1998, pp. 217–222.

[14] R. Guo and S. Venkataranman, "A technique for fault diagnosis of defects in scan chains," in *Proc. IEEE Int. Test Conf.*, 2001, pp. 268–277.

[15] S. Davidson, "ITC99 benchmark circuits—preliminary results," in *Proc. IEEE Int. Test Conf.*, 1999, p. 1125.

[16] P. High, D. Vallett, and A. Patel *et al.*, "Failure analysis of timing and IDDQ failures from the SEMATECH test methods experiment," in *Proc. IEEE Int. Test Conf.*, 1998, pp. 43–52.

[17] J. Schafer, F. Policastri, and R. Mcnulty, "Partner SRLs for improved shift register diagnostics," in *Proc. IEEE VLSI Test Symp.*, 1992, pp. 198–201.

[18] S. Edirisooriya and G. Edirisooriya, "Diagnosis of scan path failures," in *Proc. IEEE VLSI Test Symp.*, 1995, pp. 250–255.

[19] S. Narayanan and A. Das, "An efficient scheme to diagnose scan chains," in *Proc. IEEE Int. Test Conf.*, 1997, pp. 704–713.

[20] J. Hirase, N. Shindou, and K. Akahori, "Scan chain diagnosis using IDDQ current measurement," in *Proc. Asian Test Symp.*, 1999, pp. 153–157.

[21] M. A. Iyer and M. Abramovici, "FIRE: A fault-independent combinational redundancy identification algorithm," *IEEE Trans. Very Large Scale Intergr. (VLSI) Syst.*, vol. 4, no. 2, pp. 295–301, Jun. 1996.

[22] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Norwell, MA: Kluwer, 2000, p. 226.

[23] *TetraMAX ATGP User Guide*, V-2003.12, Dec. 2003.

[24] J. C.-M. Li, "Diagnosis of timing faults in scan chains using single excitation patterns," *IEICE Trans. Electron.*, vol. E88-A, no. 4, pp. 1024–1030, Apr. 2005.

James Chien-Mo Li (M'02) received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1993, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1997 and 2002, respectively.

During his Ph.D. studies, he worked with Logic Vision, Sun and Xilinx, as a Summer Intern Engineer. Since 2002, he has been an Assistant Professor with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. His research interest includes design for testability, built-in self test, defect-based testing, and fault diagnosis.

Authorized licensed use limited to: National Taiwan University. Downloaded on February 4, 2009 at 00:46 from IEEE Xplore. Restrictions apply.