Diagnosis of Resistive-Open and Stuck-Open Defects in Digital CMOS ICs

James Chien-Mo Li, *Member, IEEE*, and Edward J. McCluskey, *Life Fellow, IEEE*

***Abstract*—A resistive-open defect is an imperfect circuit connection that can be modeled as a defect resistor between two circuit nodes that should be connected. A stuck-open (SOP) defect is a complete break (no current flow) between two circuit nodes that should be connected. Conventional single stuck-at fault diagnosis cannot precisely diagnose these two defects because the test results of defective chips depend on the sequence of test patterns. This paper presents precise diagnosis techniques for these two defects. The diagnosis techniques take the test-pattern sequence into account, and therefore, produce precise diagnosis results. Also, our diagnosis technique handles multiple faults of different fault models. The diagnosis techniques are validated by experimental results. Twelve SOP and one resistive-open chips are diagnosed out of a total of 459 defective chips.**

***Index Terms*—Automatic test pattern generation (ATPG), fault diagnosis, testing, very large scale integration (VLSI).**

I. Introduction

INTEGRATED CIRCUITs (ICs) require thorough production tests to detect production defects before they are sold to the customers. Production defects are defects that are induced accidentally during the manufacturing process. One of the important categories of production defects is open defects. Open defects are breaks or imperfections in circuit interconnections such as wires, contacts, vias, silicides, and so forth. In modern manufacturing technologies, open defects are becoming more frequent [1], [2]. This is partly because the number of vias and contacts increases significantly [1], and partly because of the presence of new process techniques, such as the duodamascen copper process [2]. However, open defects are hard to detect, and hence, often cause test escapes [3]. Test escapes are defective chips that pass production tests. IC test escapes not only dissatisfy the customer, but also cost huge amounts of money and effort to replace them in the system. Effective testing for open defects has been presented in [4].

Once the defective chips are identified, it is important to diagnose the open defects. Diagnosis locates the failure site on the chip, so that a failure analysis can be performed to physically examine the defect. Precise diagnosis of open defects not only saves time and labor spent on failure analysis, but also helps the IC manufacturers fix the process problems and improve the yield. In this paper, we present precise diagnosis techniques for two major types of open defects: resistive-open and stuck-open (SOP) defects.

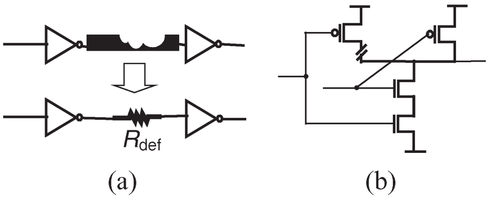


Fig. 1. (a) Resistive open defect. (b) Stuck-open defect.

A resistive-open defect is defined as an imperfect circuit connection that can be modeled as a defective resistor between the circuit nodes that should be connected. Fig. 1(a) shows a defective wire that can be modeled by a resistive-open defect (R_{def}). Examples of resistive-open defects are thin wires [5], ill-formed contacts (vias) [6], or cracks in silicides [3], [7]. It is difficult to detect resistive-open defects, because they have timing-dependent test results, which means the test results change with the test speed.

So far, there are few publications about the diagnosis of resistive-open defects. Needham *et al.* demonstrate a diagnosis case of resistive-open defects [3]. They use e-beam probing to locate the failure site without special diagnosis software. Other related publications, like transition fault simulation and diagnosis are presented in [8]–[11]. However, no real experimental data are shown.

An SOP defect is a complete break between circuit nodes that should be connected. Fig. 1(b) shows an example of an SOP defect. SOP defects have been presented since 1978 [12] and are still reported in modern technologies [3]. Examples of SOP defects can be missing metal wires [13], missing contacts or vias [3], contact misalignment [14], and bad transistors. A circuit with an SOP defect can have sequence-dependent test results, which means the test result depends on the ordering of the test patterns, even though the circuit is fully combinational [12], [15].

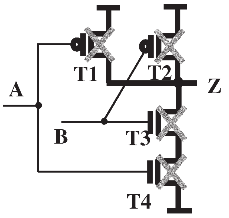
Stuck-open fault-simulation techniques have been presented in [16]–[18]. Their research focuses on fault simulations, which are only parts of diagnosis. Also, there are no experimental results shown. Diagnosis of open defects in circuit interconnections has been presented in [19] and [20]. Their techniques focus on intergate open defects only and ignore

Manuscript received September 14, 2003; revised March 14, 2004. This work was supported by LSI Logic under Contract 16517, by the National Science Foundation under Grant 2DSA414, by Semiconductor Research Corporation under Contract 2000-TJ-845, and by the National Science Council, Taiwan, R.O.C., under Grant NSC-93-2220-E-002. This paper was recommended by Associate Editor S. M. Reddy.

J. C.-M. Li is with the Electrical Engineering Department and Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C.

E. J. McCluskey is with the Center for Reliable Computing, Stanford University, Stanford, CA 94305 USA.

Digital Object Identifier 10.1109/TCAD.2005.852457



A	B	Z_{Good}	Z_{T1}	Z_{T2}	$Z_{T3,T4}$
0	0	1	1	1	1
0	1	1	H	1	1
1	0	1	1	H	1
1	1	0	0	0	H

Fig. 2. Four stuck-open faults in NAND gate (with truth table).

intragate open defects. Techniques used in [19] and [20] are based on single stuck-at fault (SSF) simulations, and do not consider sequence dependence.

We have shown some diagnosis results for SOP defects and resistive-open defects, but the technical details are skipped due to space limitations [4], [15]. In this paper, we will describe the diagnosis techniques in full detail. Our diagnosis is precise, because it takes the sequence-dependent behavior of open defects into consideration. Our technique is also capable of using more than one fault model at the same time. Traditionally, it is assumed that only a single fault is present when generating test sets. Our experimental results show that there are defective chips diagnosed as having multiple faults at the same time. This indicates that we need to be more careful about making the single-fault assumption.

As a byproduct of the diagnosis for SOP defects, we demonstrate that timing skew affects the detection of SOP defects [15]. Timing skew is the uncertainty in the relative timing of two changing signals [21]. Literature [22] discusses how timing skew invalidates the test effectiveness for SOP faults, but no real data are shown.

The research presented in this paper is validated by our experiments on Murphy chips (0.7 μm technology) [23], [24] and ELF35 chips (0.35 μm technology) [25]. Twelve SOP suspect chips and one resistive-open chip are diagnosed (out of 116 defective Murphy chips and 343 defective ELF35 chips).

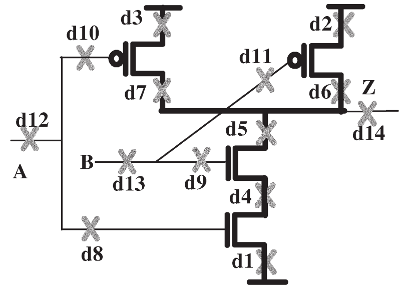
This paper is organized as follows. Section II describes the diagnosis technique for SOP defects. Diagnosis procedures followed by experimental data are presented. Section III presents the diagnosis methods and results of resistive-open defects. Section IV discusses some important issues, and finally, Section V concludes this paper.

II. Diagnosis of Stuck-Open Defects

A. Background

For testing and diagnostic purposes, people use fault models to model failures on logic signals. The SSF, for example, is the most commonly used fault model. In this section, we use the SOP fault model [12] as our diagnosis fault model. An SOP fault in transistor T1 means that transistor T1 fails to conduct. Fig. 2 shows a NAND gate and its four SOP fault sites.

The truth table of a good NAND gate and a faulty NAND gate is also shown in Fig. 2. An "H" entry in the table means that the output is in a high-impedance state and can hold the previous output value. To detect the SOP fault T1, we need to apply two patterns: the first pattern $(\mathrm{AB} = 11)$ pulls down the output Z to logic zero and then the second pattern $(\mathrm{AB} = 01)$ turns on only the transistor T1 to pull up the output Z. The expected output of a good circuit is Z $=$ 01, but the output of a faulty circuit is Z $=$ 00. The fault T1 is therefore detected.



Defects	SOP Fault
d3, d7, d10	T1
d2, d6, d11	T2
d1, d4, d5, d8, d9	T3, T4

Fig. 3. Fourteen open-defect locations (d1–d14) of NAND.

TABLE I
Modeling 11 IntraGate Stuck-Open Defects by Four Stuck-Open Faults

Pattern	In/Out			Detected SOP Fault*				Detected Defects
	A	B	Z	T1	T2	T3	T4	
1	10	11	01	D				d3,d7,d10,
2	01	11	10			D	D	d1,d4,d5,d8,d9
3	11	10	01		D			d2,d6,d11
4	11	01	10			D	D	d1,d4,d5,d8,d9
5	01	01	10			D	D	d1,d4,d5,d8,d9

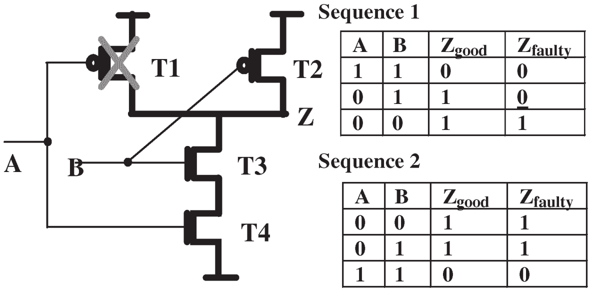
*D = fault detected

Fig. 3 shows the 14 possible locations of open defects of a NAND gate. The intragate open defects (d1 to d11) can be modeled by SOP faults [26]. Since our diagnosis technique uses the SOP fault model, our diagnosis scheme is capable of diagnosing the intragate SOP defects. Intergate open defects (d12 to d14), which are not modeled by SOP faults, can be diagnosed by an interconnect diagnosis technique like that in [19] and [20].

Table I shows how we model the 11 intragate open defects by four SOP faults. Pattern pair 1 has a falling transition in input A and input B stays static 1. This pattern detects SOP fault T1. This pattern also detects the defects d3, d7, and d10. The test pattern pairs required to detect defects d3, d7, and d10 are the same as the pattern pairs required to detect SOP fault T1. So we model defects d3, d7, and d10 by SOP fault T1. The other faults and defects can be examined in the same way. The patterns that are not listed in the table detect no fault and have no defect.

Note that in this table we assume that a PMOS transistor with its gate open (like d10 and d11) is not conducting, i.e., the floating-gate voltage is positive. Our assumption is validated by Johnson's experiment (3 μm technology) [27]. Unfortunately, there are no experimental data available for 0.7 μm technology (for Murphy) or 0.35 μm technology (for ELF35).

Note that the trapped charge polarity could be dependent on many factors. Negative floating-gate voltage has been shown in [28] (0.8 μm technology). The negative charge trapped is not an anomaly; it is due to the nature of the manufacturing process used. If this contamination happens, the PMOS with the floating gate can be partially on. The defective chips may or may not pass Boolean testing. If they fail Boolean testing, our diagnosis technique can still be applied; otherwise, an quiescent power supply current (IDDQ) diagnosis technique, such as [29],



Sequence 1

A	B	Z_{good}	Z_{faulty}
1	1	0	0
0	1	1	0
0	0	1	1

Sequence 2

A	B	Z_{good}	Z_{faulty}
0	0	1	1
0	1	1	1
1	1	0	0

Fig. 4. Example of sequence dependence.

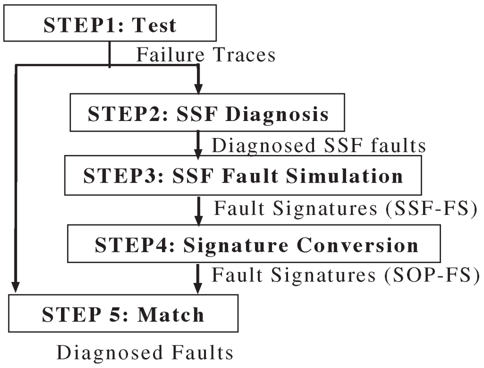

STEP1: Test
Failure Traces
STEP2: SSF Diagnosis
Diagnosed SSF faults
STEP3: SSF Fault Simulation
Fault Signatures (SSF-FS)
STEP4: Signature Conversion
Fault Signatures (SOP-FS)
STEP 5: Match
Diagnosed Faults

Fig. 5. Diagnosis flow.

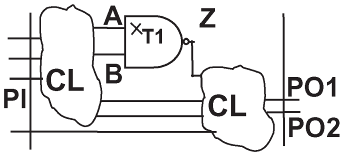


Fig. 6. Example circuit.

TABLE II
FAILURE TRACES FROM TESTER

Failing Pattern	Failing Pin
L	PO1
N	PO2

can be applied to precisely locate the defect. In this paper, the defective chips under diagnosis fail at least one Boolean testing, so our diagnosis technique is applicable.

Our diagnosis technique is based on the SOP fault model and takes sequence dependence into account. Sequence dependence means the test results of defective chips depend on the ordering of the test patterns. Wadsack shows that circuits with SOP faults have sequence-dependent test results [12]. Fig. 4 shows an example of a defective NAND gate with SOP fault in T1. This fault makes one pull-up branch fail, but the other pull-up branch can still work. The first test-pattern sequence can detect this fault because the good output and the faulty output are different in the second pattern. However, if we reverse the test-pattern sequence, as in the second sequence, we cannot detect this fault. The faulty outputs are identical with the good outputs. This is because the charge stored in the output node Z invalidates the test.

Note that the charges stored in floating node Z can change very slowly with time. In our diagnosis experiment, we run the tests at a speed of 1 MHz $(\text{cycle time} = 1\ \mu\text{s})$, so that we assume the charges stored at node Z will not change within such a short time.

Our diagnosis technique is a logic-level diagnosis scheme. We simulate the behavior of SOP faults based on the schematic information provided by the manufacturer's data book [30]. As long as the Murphy and ELF chips are actually implemented as is specified in the book, our logic-level modeling of the SOP fault is correct and the diagnosis results are valid. Also, note that SOP faults can be caused by various physical defects such as missing contacts or bad transistors. Our diagnosis technique reports faults based on their faulty behavior. Whether the actual defects are missing contacts or bad transistors cannot be known, unless a physical failure analysis is performed.

B. Diagnosis Flow

Fig. 5 shows the overall diagnosis flow. Every step in this figure will be described in detail in the following sections. Our major contribution is Step 4, signature conversion. To facilitate our discussion, we use a combinational circuit (Fig. 6) that has an SOP fault T1 in a NAND gate as an example. In this figure, PI stands for primary inputs and PO stands for primary outputs. CL stands for combinational logic.

1) Step 1. Test: Defective chips are tested on the tester and their failure traces (FTs) are recorded. The FTs from the tester contain two pieces of information: failing patterns and failing pins. The failing patterns are the patterns at which the chip failed. The failing pins of a failing pattern are the primary output pins that give erroneous logic values. For the example circuit, suppose the tester observes erroneous output values twice: one failure occurs when the test pattern L is applied and the other failure occurs when the test pattern N is applied. The failures occur at primary output pins PO1 and PO2, respectively. The FTs from the tester would look like Table II.

2) Step 2. SSF Diagnosis: After testing the chip on the tester, the FTs are given to a commercial SSF diagnosis tool and a list of diagnosed SSF faults are produced. The faulty gates are defined as the gates that contain diagnosed faults at their input or output pins. For the same NAND gate example, if the SSF diagnosis tool reports A stuck-at-1 as the diagnosed fault, then the NAND gate is the faulty gate. The reason for doing this SSF diagnosis is to reduce the number of candidate faults, so that we can save computation time and storage space for the next step, SSF fault simulation. Please see the discussions in Section IV for more details about this step.

3) Step 3. SSF Fault Simulation: All the SSF of the inputs of faulty gates obtained from STEP 2 are fault simulated (with the test patterns applied on the tester) and their fault signatures (FSs) are produced. FSs of a fault are the failing patterns and the failing pins obtained from fault simulation with the fault injected. Continuing with the same NAND gate example, Table III shows the FSs of the A stuck-at-1 fault. Three failures are predicted to occur when test patterns L, M, and N are applied, if A stuck-at fault is present.

TABLE III
FAULT SIGNATURES FOR A STUCK-AT 1 FAULT

Failing Pattern	Failing Pin
L	PO1
M	PO2
N	PO2

TABLE IV
EXCITATION CONDITION FOR SOP FAULT T1

A	B	Comment
1	1	Z = 0, initialize pattern
0	1	Z = 1, strobe pattern, also detect A stuck-at 1

TABLE V
EC TABLE FOR NAND (STUCK-OPEN FAULT)

A	B	Excited Fault
10	11	T1
11	10	T2
01	11	T3,T4
11	01	T3,T4
01	01	T3,T4

4) Step 4. Signature Conversion: In this step, we want to convert the FSs for stuck-at faults into the SOPs for SOP faults. Before that, we need two tables: the excitation condition (EC) table and gate-input sequence (GIS) table. They are described in detail below.

Table IV shows the ECs for a T1 SOP fault. The first input is an initialization pattern that sets up the NAND gate for exciting the fault. The second input is a strobe pattern, after which the output is observed. The second pattern also detects A stuck-at-1 fault. The necessary condition for detecting the SOP fault T1 is that the second pattern detects A stuck-at-1 fault.

For every type of gate, we have an EC table. The EC table for a certain gate contains the gate input pair to excite the SOP faults in the gate. Table V shows an EC table for NAND gates. Every row shows a gate-input pair and the excited SOP fault. For example, the first row of this table means that the gate input pair "A falls and B remains static 1" excites SOP fault T1. The EC table is made based on the schematic information provided by the manufacturer [30].

A fault-free logic simulation is performed on a good circuit to obtain a GIS table, which lists the gate-input values for every test pattern that is applied on the tester. For the same NAND gate example, Table VI shows the GIS table for the faulty NAND gate (diagnosed from Step 2). Every row represents a test pattern applied. The gate output Z is also shown for reference. The whole table starts from the first pattern and ends at the last pattern. Only the test patterns L, M, and N are shown here for illustration purpose.

Now that we have the EC and GIS tables, we can convert the SSF-FS to SOP-FS. For every failing pattern in SSF-FS, we check its previous pattern and see if any SOP fault is excited. If so, the failing pattern and failing pin are copied into the corresponding SOP-FS. If not, the failing pattern and failing pin are discarded. The key idea of this signature-conversion step is that, by using the passing-pattern information, the number of candidate faults is reduced, so the diagnosis resolution is improved.

TABLE VI
GIS TABLE FOR NAND

Pattern	A	B	Z
(Start from pattern 1)	…	…	…
L-1	1	1	0
L	0	1	1
…	…	…	…
M-1	0	0	1
M	0	1	1
…	…	…	…
N-2	1	1	0
N-1	0	1	1
N	0	1	1

TABLE VII
FAULT SIGNATURES FOR STUCK-OPEN FAULT T1

Failing Pattern	Failing Pin
L	PO1
N	PO2

For example, pattern L is the first failing pattern of SSF-FS in Table III. The GIS table (Table VI) shows that input A has a falling transition and input B keeps static 1. According to the first row of the EC table (Table V), SOP fault T1 is excited by pattern pair {L-1, L}. The propagation of the faulty effect is guaranteed by the SSF fault simulation (Step 3). The T1 SOP fault is therefore detected by pattern pair {L-1, L} and the SSF-FS of failing pattern L (first row of Table III) is copied to the SOP-FS for T1 (first row of Table VII). For the second row in Table III, the EC table (Table V) shows that pattern pair {M-1, M} does not excite any SOP fault. The SSF-FS of failing pattern M (second row of Table III), is therefore not copied to any SOP-FS.

For the third row in Table III, the GIS table (Table VI) shows that, for pattern N-1 and N, the gate inputs to NAND are identical. In this case, we go back one more pattern and check gate inputs of pattern pair {N-1, N-2}. For pattern pair {N-2, N-1}, the gate inputs excite fault T1, so the third row of Table III is copied to the SOP-FS for T1. This is because we assume the charge stored at NAND gate output Z holds for two continuous cycles. This assumption is valid as long as we do not test the chip in an extremely slow speed. (In this experiment, the test pattern is applied at a speed of 1 MHz.) Finally, we obtain SOP for SOP fault T1 as shown in Table VII.

In our implementation, the GIS table is stored statically in memory. There is no insufficient-memory problem for our circuits in the Murphy and ELF experiments. For bigger circuits, this implementation could cause memory problems. We might need to generate the GIS table dynamically to save memory.

5) Step 5. Match: In this step, we compare the FTs with the SOPs of SOP faults. Fig. 7 shows the relation between the FS of a certain fault and the FTs from the tester. A failing element has a failing pattern and a failing pin. Those failing elements that are in the FSs and in the FTs from the tester belong to the category of intersection (I). The number of failing elements in I is denoted as |I|. Those failing elements that are found only in FSs but not in the FTs, belong to the category of Simulation Only. Those failing elements that are found only in the FTs but not in the FSs belong to the category of Tester

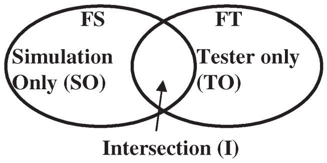


Fig. 7. Relationship between FS and FT.

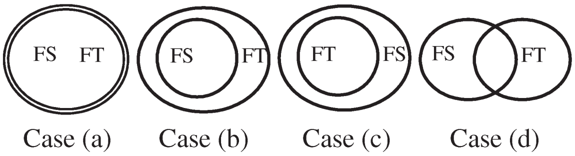


Fig. 8. Four possible matching outcomes: (a) $P = 100$, $M = 100$; (b) $P = 100$, $M < 100$; (c) $M = 100$, $P < 100$; (d) $P < 100$, $M < 100$.

Only. Note that every failing pin is counted as an individual failing element, so that every failing pin is matched. The reason for doing so is that we want to have a strong match (like [19] and [29]) such that the diagnosis resolutions can be improved.

For a given fault, we use two scores to represent how well the FSs match the FTs: the prediction score and the matching score [20]. The prediction score (P) is defined as the number of failing elements in the intersection $(|\mathrm{I}|)$ over the number of failing elements in the FSs $(|\mathrm{FS}|)$.

$$\text{Prediction Score } (P) = \frac{|I|}{|\mathrm{FS}|} \times 100. \qquad (5\text{-}1)$$

The matching score (M) is defined as the number of failures in the intersection $(|\mathrm{I}|)$ over the number of failing elements in the failing traces $(|\mathrm{FT}|)$.

$$\text{Matching Score } (M) = \frac{|I|}{|\mathrm{FT}|} \times 100. \qquad (5\text{-}2)$$

Fig. 8 shows the four possible matching outcomes. A perfect match, like Fig. 8(a), happens if the FSs of a particular fault and the FTs are identical. The matching score is 100 and the prediction score is also 100.

Fig. 8(b) shows a case in which the P score is 100, but the M score is less than 100. All the FSs are observed on the tester, but there remain some FTs that are not predicted. This means that there exist some tester-only failures. One possible explanation is that the fault under diagnosis is one of the multiple faults. Multiple faults mean the presence of more than one fault at the same time in a defective chip. For example, suppose the chip under diagnosis carries two faults. The FSs of fault 1 match some of the FTs, and the FSs of fault 2 match the other FTs. Both fault 1 and fault 2 will have P scores equal to 100, but their individual M scores are less than 100.

To account for the multiple-faults problem, a greedy algorithm can be used for matching. Fig. 9 shows the matching algorithm. The candidate fault list $(\mathbf{F})$ contains all the faults from Step 4. The result fault list $(\mathbf{F}_{\text{results}})$ is the group of diagnosed multiple faults. The result FS $(\mathbf{FS}_{\text{results}})$ contains the FSs of all the faults in $\mathbf{F}_{\text{results}}$. The $\mathbf{F}_{\text{results}}$ and $\mathbf{FS}_{\text{results}}$ are initially empty. An M score and a P score are calculated for every fault in $\mathbf{F}$. We first pick a fault (f_x) with $P = 100$ and the highest M score in $\mathbf{F}$. We save the FSs of f_x in $\mathbf{FS}_{\text{results}}$. If the current $\mathbf{F}_{\text{results}}$ has both P and M scores equal to 100, then the matching process is finished. This is a perfect match.

```
$\mathbf{F}$={set of candidate faults from step 4};
$\mathbf{F}_{\text{results}}$ ={};
$\mathbf{FS}_{\text{results}}$ ={};
foreach f in $\mathbf{F}$   { calculate $P_{score}(f)$ and $M_{score}(f)$};
$f_x$= a fault that $P_{score}(f_x)$ equals 100 and highest $M_{score}(f_x)$ in $\mathbf{F}$;
$\mathbf{F}_{\text{results}}$ = $f_x$;
$\mathbf{FS}_{\text{results}}$ = fault_signature($f_x$);;
While ( $\mathbf{F}$ not empty) do  {
        if ($P_{score}(\mathbf{F}_{\text{results}})$ == 100 && $M_{score}(\mathbf{F}_{\text{results}})$ == 100)
        { abort;}
        else {
                delete $f_x$ from $\mathbf{F}$
                foreach f in $\mathbf{F}$ {inc_$M_{score}(f)$ = $M_{score}(\mathbf{F}_{\text{results}} + f)$;}
                $f_x$ = a fault that $P_{score}(f_x)$ = 100 and highest inc_$M_{score}(f_x)$;
                if ($f_x$ does not exist)  { abort;}
                insert $f_x$ into $\mathbf{F}_{\text{results}}$ ;
                $\mathbf{FS}_{\text{results}}$ = $\mathbf{FS}_{\text{results}}$ + fault_signature($f_x$);
        }
}
report $\mathbf{F}_{\text{results}}$ and scores;
```

Fig. 9. Matching algorithm.

If there is no perfect match, we delete f_x from the fault list $\mathbf{F}$. We calculate the incremental M_{score} for every fault in $\mathbf{F}$. The incremental M_{score} of fault f is the M_{score} of fault f, together with all the faults in $\mathbf{F}_{\text{results}}$. We then pick the fault f_x with P score $= 100$ and the highest incremental M score in $\mathbf{F}$. Insert f_x into $\mathbf{F}_{\text{results}}$ and add the FSs of f_x into $\mathbf{FS}_{\text{results}}$. (The adding of FSs removes the overlapped failing pin and failing patterns, in case two faults have the same failing pins in the same failing patterns.) This process is repeated until we have a perfect match or there is no more f_x.

Besides multiple faults, another possible explanation for case (b) is timing skew. Our diagnosis technique can deal with this timing skew problem by modifying the EC table. Please see Section II for a more detailed discussions.

Fig. 8(c) shows a case in which the M score is 100 but the P score is less than 100. All the FTs are contained in the FSs, but there exist some FSs that are not observed on the tester. This means that there exist some simulation-only failures. A possible explanation is that the faulty net under matching is involved in a bridging fault. For example, suppose the chip under diagnosis has a bridging fault (wire-AND) between net X and net Y. The bridging fault is excited only when net X and net Y have different values. When net X is logic 1, failure occurs only when net Y is 0. The FTs are hence a subset of the FSs of the X stuck-at-0 fault. Since we are targeting SOP faults, our diagnosis technique does not have to deal with this case.

Another possible explanation for case 8(c) is the fault masking of multiple faults. If fault masking happens, the pins and patterns that are expected to fail for one fault can pass, due to fault masking by another fault. The matching algorithm in Fig. 9 can be modified to account for the fault masking. The major change is that when we calculate the incremental M score, instead of simply adding the FSs together, we have to run fault simulation with multiple faults present at the same time. In our Murphy experiment, we see no evidence of fault masking, so the matching algorithm for fault masking is not implemented.

Fig. 8(d) shows the last case in which M and P are less than 100. If there is no intersection, both P and M are zero. We do find this case in our experiment. Our diagnosis so far cannot successfully handle this case. There are several possible explanations for this unsuccessful diagnosis. Please see 3) of Section II-C for more explanations.

C. Diagnosis Results of Murphy Chips

In the Murphy experiment, some test patterns have been applied in the following five different orders [32]:

1) insert an all-zero pattern between every pair of original patterns;
2) insert an all-one pattern between every pair of original patterns;
3) reverse the original test pattern sequence (i.e., last pattern applied first);
4) insert a bitwise complement pattern before every original pattern;
5) insert a one-bit shifted pattern before every original pattern.

The outputs are only observed for the original patterns so that the SSF coverages of the original and the modified versions of the test patterns are the same.

Eleven out of 116 defective Murphy chips are found to be sequence dependent but not timing dependent. It means that their test results depend on the order of the patterns, but not on the application speed (as long as the speed does not exceed the specified speed). These chips are possible candidates for carrying SOP defects.

In this diagnosis experiment, 15-detect SSF test sets, which detect every SSF at least 15 times are applied. The 15-detect SSF test sets are among the most effective test sets in the Murphy experiment [24]. All the test sets are applied at nominal voltage. Seven out of the 11 sequence-dependent chips are diagnosed as having single SOP faults. Table VIII summarizes the diagnosis results. These seven chips are numbered as SD.1 to SD.7. The first column shows the number of faults diagnosed by a commercial SSF diagnosis tool. The SSF diagnosis results are not perfect for these chips because their test results are sequence dependent and cannot be modeled well by the SSF model. The second column shows the number of faults diagnosed by our SOP fault diagnosis. Their FTs perfectly match their FSs. They are all diagnosed to a single fault, so our diagnosis gives very precise results. The faulty gate types that contain the SOP faults are also listed. These seven chips are called SOP-suspect chips in the following text.

Note that SD.6 and SD.7 have a large number of diagnosed SSFs. There are mainly two reasons for that. First, the design of this particular circuits under test (CUT) has many equivalent faults. SD.6 and SD.7 are of the same design, referred to as ROB in [23]. There are 23 and 5 equivalent faults out of the diagnosed 32 and 9 faults, respectively. That is, there are 9 and 4 nonequivalent faults (fault class) for SD.6 and SD.7. Second, the commercial diagnosis tool tries to explain all the failing patterns. Because of the timing skew problems of these two chips (more details later), many unexpected failing patterns occur in the FTs. As a result, the tool reports many innocent faults to explain these unexpected failing patterns.

TABLE VIII
DIAGNOSIS RESULTS OF STUCK-OPEN SUSPECT CHIPS (SD.1 TO SD.7)

Chip ID	SSF Diagnosis	Stuck-Open Fault Diagnosis		
	Number of Faults	Number of Faults	Faulty Gate	P/M
SD.1	3	1	NOR	100/100
SD.2	1	1	NOR	100/100
SD.3	4	1	AOI	100/100
SD.4	4	1	OR	100/100
SD.5	1	1	AOI	100/100
SD.6*	32	1	NAND	100/100
SD.7*	9	1	OAI	100/100

* timing skew

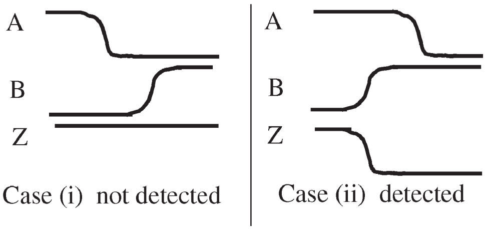


Fig. 10. Two cases of timing skew.

1) Timing Skew: For chips SD.6 and SD.7, their matching results were originally not perfect. Their P scores were 100 but the M scores were less than 100. This is the case (b) of the four match outcomes in Fig. 8. Upon further investigation, it is found that chip SD.6 is likely to have an SOP fault T1 in a NAND gate. It has FSs when the gate-input pair is $\{\mathrm{AB} = 11, \mathrm{AB} = 01\}$. In addition, it has tester-only FTs when the gate-input pair is $\{\mathrm{AB} = 10, \mathrm{AB} = 01\}$. This could be explained by the static hazard caused by timing skew of gate inputs A and B. A static hazard is present if it is possible for a momentary change of output to occur in response to an input change that does not cause the steady-state output to change [33].

Fig. 10 shows how timing skew affects the excitation of SOP faults. Assume that we have a T1 SOP fault in the NAND gate in Fig. 2 and we have a gate input pair $\{\mathrm{AB} = 10, \mathrm{AB} = 01\}$. In case (i), the A input falls first and then B input rises. The output Z can be kept at logic 1 due to the parasitic capacitor holding the charges at the faulty-gate output. In this case, the SOP fault cannot be detected. In case (ii), input B rises first and output A falls later. For a good NAND gate, the output Z is pulled down momentarily and then pulled back up again (i.e., static-1 hazard). For a defective NAND gate, the output Z is pulled down, but cannot be pulled back to 1, due to SOP fault T1. In this case, the SOP fault can be detected (as long as the fault effect is propagated to the primary output). This can explain why chip SD.6 failed the gate-input pair $\{\mathrm{AB} = 10, \mathrm{AB} = 01\}$.

To verify our assumption, we apply a test set in which only the single input of the faulty NAND gate changes at a time, i.e., the GIS is $\{\mathrm{AB} = 10, \mathrm{AB} = 11, \mathrm{AB} = 01\}$. The chip SD.6 fails this test. This shows that the case (ii) in Fig. 10 does detect the SOP fault. Another GIS applied is $\{\mathrm{AB} = 10, \mathrm{AB} = 00, \mathrm{AB} = 01\}$. SD.6 passed this test because $\mathrm{AB} = 00$ eliminates the static-hazard condition. This shows that case (i) cannot detect the SOP fault.

Another experiment is performed to verify our timing-skew assumption. The same 15-detect SSF test sets are applied at a faster speed (5% faster than nominal speed). The purpose of this experiment is to see if the timing skew problem goes away when the test speed is fast. Experimental results show that the chip, SD.6, has the same failures in the fast speed test as the nominal speed test. We try some even faster speed (like 10% faster than nominal speed), but cannot succeed, because even the good circuits fail at 10% faster than nominal speed. We cannot experimentally show that the timing-skew problem goes away with faster tests.

It is shown by our experiment that timing skew causes the unexpected detection of the SOP fault. Note that timing skew can also invalidate the test for SOP faults [16]–[22]. For example, test sequence $\{\mathrm{AB} = 11, \mathrm{AB} = 01\}$ is supposed to detect the T1 SOP fault, but this can be invalidated by a static-1 hazard on B. The static-1 hazard makes B drop to logic 0 temporarily and the output of NAND will be charged to logic 1. The defective chip therefore passes this test. Although this case is not observed in our experiment, our experimental data provide evidence that timing skew does affect the test effectiveness for SOP faults.

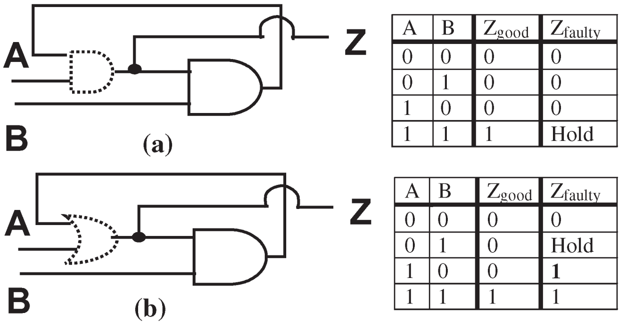
To take this timing skew problem into account, we add one entry $\{\mathrm{AB} = 10, \mathrm{AB} = 01\}$ to the EC table for NAND gate as an additional gate input pair for exciting SOP fault T1. After making this modification, the FSs of the SOP fault in the NAND gate perfectly match the FTs of SD.6. For chip SD.7, we find similar timing-skew problems. After making similar modifications to the EC table, the SOP fault diagnosis gives perfect match results.

2) Multiple Faults: For chip SD.8, the diagnosis was not successful at first. We only found an SOP fault in an OR gate with a prediction score of 100 and matching score of 9. This is case (b) in Fig. 8. After further investigation, we find an SSF in a NAND gate that has a prediction score of 100 and matching score of 91. The combination of these two faults gives a perfect match ($\mathrm{P} = 100$, $M = 100$). We therefore improve our diagnosis technique to handle more than one fault model. We feed the SSF-FS to Step 5 (see Fig. 5) so our diagnosis is now capable of handling both stuck-at and SOP fault models.

Table IX shows the diagnosis results for chips SD.8 and SD.9. Chip SD.8 has two faults diagnosed and the match is perfect. By contrast, the SSF diagnosis gives 34 faults (24 of them are equivalent, 10 of them are nonequivalent faults). For SD.9, one single SOP fault is diagnosed. The faulty gate is an OAI gate and the matching score is 53. Besides this SOP fault, there are two SSFs identified. Each one of them had a matching score of 5. The union of these three faults gave a matching score of 63, and prediction score of 100. Although the match is not perfect, this is the highest score we have got so far. Compared with SSF diagnosis that gave 20 faults (14 equivalent faults, 6 nonequivalent faults), our diagnosis results are much easier for failure analysis. SD.9 has failing patterns that cannot be explained by existing candidate faults. This means that we may need to include more faults to make the M score higher. Using the passing-pattern information does not include more faults, so it does not help in the case of SD.9.

TABLE IX
DIAGNOSIS RESULTS OF CHIPS SD.8 AND SD.9

Chip	SSF Diagnosis	Stuck-Open Fault Diagnosis		
ID	Number of Faults	Number of Faults	Faulty Gate	P/M
SD.8	34	2	NAND (SSF) OR (SOP)	100/100
SD.9	20	3	NAND (SSF) OAI (SSF) OAI (SOP)	100/63



A	B	Z_{good}	Z_{faulty}
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	Hold

A	B	Z_{good}	Z_{faulty}
0	0	0	0
0	1	0	Hold
1	0	0	**1**
1	1	1	1

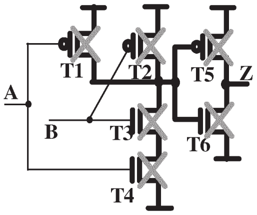
Fig. 11. Feedback-bridging fault (AZ) in an AND gate.

Multiple faults might sound very unlikely at first. But it is shown that defects tend to cluster together rather than distribute evenly on the wafers [34]. Multiple faults could be caused by clustered defects in the same die. We try to trace the locations of the diagnosed faults in SD.8 and SD.9. The faults are located in different logic cones, according to the commercial SSF diagnosis tool. We do not know, however, their physical locations because the layout information is unavailable.

3) Failed Diagnosis: Our diagnosis fails to find any single stuck-at or SOP faults for two other sequence-dependent chips, SD.10 and SD.11. Neither their P score nor M score is 100. This is case (d) in Fig. 8. There are three possible reasons for this. One possible reason is that the defects are not well modeled by stuck-at or SOP fault models. This could be because the defects are large in area or the defects are clustered. The second possibility could be that some of the library cells are not implemented exactly as shown in the data book. In this case, our simulated FSs can be invalid, and therefore fail to find a good match with the real data.

The third possible explanation can be noninverting feedback bridging faults (NFBF) [35]. Fig. 11 shows an AND gate with a feedback-bridging fault between its input A and output Z. This feedback-bridging fault can be modeled as a wire AND, as shown in Fig. 11(a). From the truth table of Fig. 11(a), it is seen that the faulty gate holds the previous output value when the inputs are $\mathrm{AB} = 11$. There is no input that can produce an output 1. Once the output Z is pulled down to 0 by any of the inputs ($\mathrm{AB} = 00$, $\mathrm{AB} = 01$, $\mathrm{AB} = 10$), this faulty AND gate always produces the output 0. The faulty AND gate behaves as having a Z stuck-at-0 fault. The wire AND model cannot explain the sequence-dependent behavior of our defective chips.

The feedback-bridging fault can also be modeled as a wired OR as shown in Fig. 11(b). The truth table in Fig. 11(b) shows that the fault is detected when the input is $\mathrm{AB} = 10$, because the faulty output is different from the fault-free output (we mark this entry in bold). The faulty output holds the previous



A	B	Z	Z_{T1}	Z_{T2}	$Z_{T3,4,5}$	Z_{T6}
0	0	0	0	0	0	H
0	1	0	H	0	0	H
1	0	0	0	H	0	H
1	1	1	1	1	H	1

Fig. 12. Six stuck-open faults in an AND gate.

state when input is $\mathrm{AB} = 01$. For input sequence $\{\mathrm{AB} = 00, \mathrm{AB} = 01, \mathrm{AB} = 11\}$, the output sequence is $\mathrm{Z} = \{001\}$. The fault cannot be detected because the faulty outputs are the same as the good outputs. For input sequence $\{\mathrm{AB} = 11, \mathrm{AB} = 01, \mathrm{AB} = 00\}$, we will get output $\mathrm{Z} = \{110\}$. The fault is detected and the faulty AND gate has sequence-dependent test results.

We carefully examined the test results of chips SD.10 and SD.11, but could not find an NFBF across one single gate that explains the FTs of these two chips. It is possible that there exists an NFBF that is across multiple gates, rather than only one gate. It is also possible that the real feedback-bridging defects have more complex behaviors than the AND/OR models shown above. Whichever reason it may be, we have to stop the discussion here because the diagnosis of NFBF without layout information is challenging and out of the scope of this paper.

Since feedback-bridging fault and SOP fault both cause sequence-dependent behavior, how do we know that the nine SOP-suspect chips (SD.1 to SD.9) are truly SOP defects? To answer this question, we consider all the six SOP faults in an AND gate, as shown in Fig. 12. As explained earlier in Section II, SOP fault T1 can cause sequence-dependent test results. Comparing the truth table of SOP fault T1 in Fig. 12 to the truth table of the feedback-bridging fault AZ (wire or model) in Fig. 11(b), we can see that the faulty outputs are different when input $\mathrm{AB} = 10$. This means that the FS of an SOP fault is different from that of a feedback-bridging fault. Since the FTs of the chips SD.1 to SD.8 match perfectly the FSs of SOP faults, they are not feedback-bridging faults. The same arguments can be applied to SOP fault T2 and the feedback-bridging fault between input B and output Z. For SOP faults T3, T4, and T5, the faulty AND gate behaves as having a Z stuck-at-0 fault. For SOP fault T6, the faulty circuit behaves as having a Z stuck-at-1 fault. Stuck-open faults T3 to T6 cannot explain the sequence-dependence behavior of our defective chips.

4) Diagnosis Results of ELF35 Chips: To demonstrate that our technique is also applicable to other technologies, we repeat our diagnosis experiment in our next test chips, ELF35. The ELF35 is fabricated in 0.35 $\mu\mathrm{m}$ standard-cell technology [36]. It has three layers of metal. It has six CUT; four of them are combinational circuits (datapath and random logic) and two of them are sequential circuits (2901 microprocessor slices). The total gate count is 265 000 gates. The ELF35 chip is designed to evaluate the effectiveness of different test techniques. Please see [25] for more details about the chip design. We have applied very thorough tests on the ELF35 chips, including SSF tests, multiple-detect SSF tests, transition fault test, path delay fault tests, $\mathrm{I_{DDQ}}$ testing, VLV testing, and so on. We even apply exhaustive and superexhaustive tests to some CUTs with a small numbers of inputs. Out of the 9600 packaged chips, we have identified 343 of them as defective.

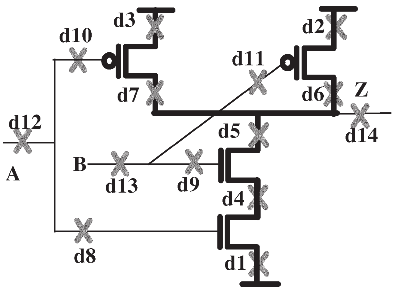
We repeat our SOP diagnosis experiment on the ELF35 chips. The diagnosis flow is the same as that of Murphy. The only difference is that we need to build new EC tables because the library cell names change. In this experiment, we apply 15-detect SSF test sets as well as 3-detect SSF test sets. The reason for choosing the multiple-detect SSF test sets is because they are shown to be very effective test sets. The 15-detect SSF test sets are very long, so in some cases the number of failing patterns exceeds 255 (the maximum allowed on the tester). For this reason, we also applied the 3-detect test sets, which are shorter, so that no chip produced more than 255 failing patterns. We have successfully diagnosed three sequence-dependent chips that are suspected to have SOP defects. They are numbered as ELF.1 to ELF.3. For ELF.1, the defective CUT is a squarer. The faulty gate is diagnosed to be a two-input NOR gate. A similar timing-skew problem is observed. When we apply input sequence $\{\mathrm{AB} = 10, \mathrm{AB} = 01\}$, the good output sequence is $\{\mathrm{Z} = 0, \mathrm{Z} = 0\}$ but the defective output sequence is $\{\mathrm{Z} = 0, \mathrm{Z} = 1\}$. We fix this timing-skew problem in the same way as Section II-C1. We add one entry $\{\mathrm{AB} = 10, \mathrm{AB} = 01\}$ to the EC table, and we get a perfect match. For ELF.2, the defective CUT is a multiplier. There are two faults diagnosed, one is a stuck-at fault in an inverter, the other one is an SOP fault in a NOR gate. These two faults are present at the same time. ELF.3 is similar to ELF.2. The defective CUT is again a multiplier. There are two faults and the faulty gates are an inverter and a NAND.

III. Diagnosis of Resistive-Open Defects

A. Background

For diagnosing resistive-open defects, we use the transition-fault model as our diagnosis fault model. Transition faults [37] are defined as a localized timing failure that is large enough such that the delay of paths through some gate to observable outputs exceeds the clock interval. Fig. 13 shows the 14 possible resistive-open defect locations in a NAND gate. The 11 intragate resistive-open defects (d1 to d11) can be modeled as single-transition faults. The intergate resistive-open defects (d12, d13, and d14) cannot be modeled as any single-transition fault. They have to be modeled as combinations of two transition faults [38].

Table X shows how we model the resistive-open defects by transition faults. For intragate defects, we can find a single-transition fault for each defect. For example, test patterns 1 and 7 detect intragate defects d3, d7, and d10. They also detect the A slow-to-fall (Af) fault. So we model d3, d7, and d10 by the Af fault. For intergate-open defects, they are not modeled by any single-transition fault. They are modeled by a combination of two transition faults. For example, test patterns 1, 2, 5, and 7 detect d12. They also detect Af or Ar faults. So d12 is modeled by the combination of the Af and the A slow-to-rise



Defects	Transition Fault
d3,d7,d10	Af
d2,d6,d11	Bf
d1,d4,d5, d8,d9	Zf
d12	Af + Ar
d13	Bf + Br
d14	Zf + Zr

Fig. 13. Fourteen possible defect locations and their transition-fault models.

TABLE X
MODELING RESISTIVE OPEN DEFECTS BY TRANSITION FAULTS

Number	In/Out			Detected Defects		Detected Transition Faults*					
	A	B	Z	Intragate	Intergate	Af	Ar	Bf	Br	Zf	Zr
1	10	11	01	d3,d7,d10	d12,d14	D					D
2	01	11	10	d1,d4,d5,d8,d9	d12,d14		D			D	
3	11	10	01	d2,d6,d11	d13,d14			D			D
4	11	01	10	d1,d4,d5,d8,d9	d13,d14				D	D	
5	01	01	10	d1,d4,d5,d8,d9	d12,d13,d14		D		D	D	
6	10	10	01		d14						D
7	10	01	11	d3,d7,d10	d12	D					
8	01	10	11	d2,d6,d11	d13			D			

*Af = A slow-to-fall fault, Ar = A slow-to-rise fault, D = fault detected

TABLE XI
EC TABLE FOR NAND (TRANSITION FAULT)

A	B	Excited Transition Fault
10	X1	Af
01	X1	Ar
X1	10	Bf
X1	01	Br
01	01	Zf
11	01	
01	11	
10	10	Zr
11	10	
10	11	

faults. The other patterns that are not listed in the table detect no defect and no fault.

B. Diagnosis Flow

We follow the diagnosis flow shown in Section II. There are two changes in Step 4, the signature conversion. The first difference is that we need an EC table for the transition faults. Table XI shows the transition fault EC table for the NAND gate.

The other difference in signature conversion is that we do not go back more than one cycle if the preceding pattern (N-1) is the same as pattern N. This is because we assume that circuits with resistive-open defects can pass the test pattern if we wait for more than one cycle. The same assumption is made in [38]. This assumption will be verified by experimental data in the next section. To illustrate this difference in signature conversion, the same NAND gate example as the one used in Section II-B is considered. Assume that we have a resistive-open defect located in d3 that can be modeled as an Af fault. For the GIS in Table VI, failing patterns L and M are treated in the same way as before. The pattern pair {L-1, L} excites the Af fault and pattern pair {M-1,M} does not excite the Af fault. Failing pattern N is treated differently from before. The failing pattern {N-1, N} cannot excite the Af fault because we do not trace back to pattern N-2 this time.

C. Diagnosis Results of Murphy Chips

The Murphy experimental results show that 39 out of 116 defective chips are timing dependent. These chips are potential candidates to have resistive-open defects. In our diagnosis experiment, all the timing-dependent chips are tested at nominal V_{DD}, room temperature, and characterized speed.

Out of the 39 defective chips, we successfully diagnose one chip to have a resistive-open defect. The diagnosed faulty gate is an inverter. The FTs match the combination of two transition faults (slow-to-rise and slow-to-fall) at the output of the inverter. The defect could be an intergate resistive open. This chip is called the resistive-open suspect chip. It has to be mentioned that the matching is not perfect. There are a few failing patterns (and failing pins) in the FSs that are not in the FTs. This could be because the difference of delay between the faulty circuit and the good circuit is so small that the fault can only be detected through long paths, but not through short paths [39].

A shmoo experiment is performed on this resistive-open suspect chip [4]. The results of the shmoo experiment confirm our previous assumption that circuits with resistive-open defects can pass the test, if we wait for more than one cycle. The delay of a good circuit is 14 ns at nominal V_{DD} (5 V) and the delay of a defective circuit is 23 ns. If the resistive-open suspect chip is tested at a clock period of 14 ns, it fails the test (14 ns $<$ 23 ns). If we double the test clock-cycle time from 14 to 28 ns (i.e., 50% slower test speed), the resistive-open suspect chip can pass the test (28 ns $>$ 22 ns).

IV. DISCUSSIONS

A. SSF Diagnosis

One of the questions regarding our diagnosis flow is this: Why do we need SSF diagnosis in Step 2? The reason for doing this SSF diagnosis is to reduce the number of candidate faults so that we can save computation time and storage space in the following steps, Step 3 (SSF fault simulation) and Step 4 (signature conversion). Another question asked is: Can we miss any SOP fault by doing an SSF diagnosis? To answer this question, we need to understand the matching algorithm of the commercial SSF diagnosis tool [31]. This tool first searches for SSF that the FSs match perfectly with the FTs. They call these faults “class I,” which corresponds to case (a) in Fig. 8. If there are no class I faults diagnosed, then the tool searches for the SSF, the FSs of which contain the FTs. They call these faults “class II,” which corresponds to our case (c) in Fig. 8. The tool reports class-II faults if there are no class-I faults diagnosed.

Suppose we have an SOP fault T1 in our example circuit in Fig. 6, and we feed the FTs to the SSF diagnosis tool. The faulty circuit has sequence-dependent test results, so there exists no SSF that perfectly matches the FTs. If the SSF tool fails to report any class-I fault, then it goes on to search for the class-II faults. The SOP fault will fall into this class-II category. This is because the necessary condition to detect SOP fault T1 is that the second pattern detects the input A stuck-at-1 fault (see Table IV). The input A stuck-at-1 fault will be reported by the tool as a class-II fault because the input A stuck-at-1 FSs contain the T1 SOP FSs. Unless there is an SSF that perfectly matches the FTs, which is unlikely for the sequence-dependent chips, the A stuck-at-1 fault will not be missed by the commercial SSF diagnosis tool. This is why we can use the commercial SSF diagnosis tool in Step 2 without missing the SOP faults.

If we are still worried about missing any candidate faults by doing the SSF diagnosis, we can skip the SSF diagnosis. If we do skip it, we could get more candidates by increasing the run time of the fault-simulation step. Based on our experience, the SSF diagnosis tool sometimes gives us too many faults (e.g., SD.6 and SD.7 in Table VIII), rather than too few faults. For the failed diagnosis (SD.10 and SD.11), since there is no SSF of $P\ \mathrm{score} = 100$ or $M\ \mathrm{score} = 100$, we cannot diagnose them by SOP fault anyway. So skipping the SSF diagnosis does not improve the diagnosis precision for our experiments.

B. Diagnosis of Sequential CUT

The circuits under diagnosis in this paper are all combinational CUTs. In the case of scan-based sequential circuits, the two-pattern test can still be applied. One of the two-pattern test techniques is the launch-on-shift (LoS), or the skew load. In this technique, the first pattern (P_1) is applied in the second-to-the-last scan cycle and the second pattern (P_2) is applied in the last scan cycle. The response is then captured by a system clock followed by a scan out. Another two-pattern test technique is the launch-on-capturer (LoC), or broadside load. In this technique, the first pattern is applied by the last scan and the second pattern is applied by a system clock. The response of the CUT is then captured by another system clock followed by a scan out. For either LoS or LoC, our diagnosis technique can still be applied with modifications. The first modification is that the GIS table now contains pattern pairs $(\mathrm{P}_1, \mathrm{P}_2)$. The cycle to record the P_1 and P_2 patterns depends on the technique used. P_1 is recorded in the second-to-the-last scan cycle and the last scan cycle for the LoS and LoC, respectively. P_2 is recorded in the next cycle of P_1. The second modification is in the signature conversion. We only compare GIS entries within the test pattern pairs, not across different test pattern pairs. This is because the contents of the scan chain are reloaded between scan patterns. The other steps of our diagnosis remain mostly unchanged. Note that the tests do not have to be applied at speed to detect the SOP defects. In fact, the diagnosed SOP Murphy chips are timing independent, which means the test results do not change with different test timing.

C. Sequence Dependence of Sequential CUT

In our experiment, the circuits are combinational, so test patterns can be arbitrarily reordered. In the case of scan-based sequential circuits, sequence dependence cannot be determined by arbitrarily reordering the scan patterns. However, there are some special cases in which test pattern pairs $(\mathrm{P}_1, \mathrm{P}_2)$ can be reordered to $(\mathrm{P}_2, \mathrm{P}_1)$. For example, $\mathrm{P}_1 = (101010)$ and $\mathrm{P}_2 = (010101)$ is a pair of scan patterns that can be swapped when applying the LoS test technique. This is because P_1 is a shifted version of P_2, and vice versa. The other possible way to reorder the pattern pair is to use the double-latch level sensitive scan design, in which P_1 and P_2 are stored in the L_1 and L_2 latches, respectively [40]. The reordering of P_1 and P_2 is possible by swapping the contents of the L_1 and L_2 latches.

D. Choosing Diagnosis Algorithm

The diagnosis results can be poor if a wrong fault model or a wrong diagnosis algorithm is assumed. Unfortunately, given a defective chip, there is usually no simple way to determine in advance which diagnosis algorithm to choose. In our Murphy and ELF experiments, defective chips are thoroughly tested so their behaviors are well understood [24]. The diagnosis algorithm can therefore be chosen according to their behavior. We use the SOP diagnosis on those timing-independent and sequence-dependent CUT because the sequence-dependent behavior is observed by previous research in intragate open defects [12], not observed by any research in intergate open defects. We use the resistive-open diagnosis on those timing-dependent CUTs because resistive-open defects are known to cause timing problems [3]. For choosing a right diagnosis technique, it is therefore very important to classify the CUTs according to their behaviors.

E. Interaction of Multiple Faults

In the presence of multiple faults, the fault effects can interact with one another. For example, a failing pin of a fault can be masked in the presence of multiple faults. In our experiment, we trace the logic cones of the diagnosed faults of chips SD.8 and SD.9. No overlapping between the logic cones of diagnosed faults is found. So far, there is no evidence of multiple faults interaction in our experiment.

The proposed technique, nevertheless, can be modified in two ways to handle the interaction of multiple faults. The first solution is to do the match (Step 5) on the failing-pattern basis. That is, the failing elements, which used to contain both the failing patterns and failing pins, now contain only the failing patterns. In this way, the individual failing pins are ignored so that the masked failing pins do not affect the match results. The disadvantage of this solution is the loss of diagnosis resolution. Because the modified-match condition is weaker than the original one, the matched faults of the former can be more than those of the latter. More matched faults mean poorer diagnosis resolution. In addition, there is a small chance that multiple faults happen to totally cancel each other out in some failing patterns. If that happens, the modified-match method cannot diagnose these faults successfully.

The second solution is to insert a multiple stuck-at fault simulation between the original Step 4 (SSF fault simulation) and Step 5 (signature conversion). The multiple stuck-at fault simulator is different from the traditional SSF simulator in that the former assumes the presence of multiple stuck-at faults at the same time. All possible groups of stuck-at faults are fault simulated and the produced FSs are then converted into the SOP FSs. This solution is more accurate than the previous one because the masked failing pins are removed from the stuck-at FSs by the multiple stuck-at fault simulation. This solution, however, can be very time consuming if the number of possible groups of stuck-at faults is large. In addition, the multiple stuck-at fault simulator is not widely available nowadays and the implementation of the simulator is out of the scope of this paper.

V. Summary

Diagnosis schemes for resistive-open and SOP defects are presented in this paper. The diagnosis made uses the SOP, stuck-at, and transition-fault models. The diagnosis takes the sequence dependence of open defects into account. It is capable of diagnosing multiple faults of more than one fault model. The diagnosis techniques have been demonstrated on the Murphy and ELF35 chips. Nine Murphy chips and three ELF35 chips to have SOPs (out of a total of 116 defective Murphy chips and 343 defective ELF35 chips, respectively) have been identified. One resistive-open defect on Murphy chips was also diagnosed. The experimental results show that there are defective chips that have multiple faults at the same time. Besides diagnosis, it was also experimentally demonstrated that timing skew affects the detection of SOP defects.

Acknowledgment

The authors would like to thank all reviewers, especially Dr. S. Mitra, for his helpful comments, and Advantest Cooperation USA for the tester-time donation.

References

[1] K. M. Thompson, "Intel and the myths of test," *IEEE Des. Test. Comput.*, vol. 13, no. 1, pp. 79–81, 1996.

[2] A. Stamper, T. L. McDevitt, and S. L. Luce, "Sub-0.25-micron interconnect scaling: Damascene copper versus subtractive aluminum," in *Proc. IEEE Advanced Semiconductor Manufacturing Conf.*, Boston, MA, 1998, pp. 337–346.

[3] W. Needham, C. Prunty, and E. Yeoh, "High volume microprocessor escapes, an analysis of defects our tests are missing," in *Proc. Int. Test Conf.*, Washington, DC, 1998, pp. 25–34.

[4] J. C.-M. Li, C. W. Tseng, and E. J. McCluskey, "Testing for resistive and stuck opens," in *Proc. Int. Test Conf.*, Baltimore, MD, 2001 pp. 1049–1058.

[5] E. Yeoh and M. We, "The application of novel failure analysis techniques for advanced multi layered CMOS devices," in *Proc. Int. Test Conf.*, Washington, DC, 1997, pp. 304–309.

[6] K. Baker, G. Gronthoud, M. Lousberg, I. Schanstra, and C. Hawkins, "Defect-based delay testing of resistive vias-contacts," in *Proc. Int. Test Conf.*, Atlantic City, NJ, 1999, pp. 467–476.

[7] C. W. Tseng, X. Shao, D. M. Wu, and E. J. McCluskey, "Cold delay defect screening," in *Proc. VLSI Test Symp.*, Montreal, QC, Canada, 2000, pp. 183–188.

[8] J. A. Waicukauski and E. Lindbloom, "Transition fault simulation by parallel pattern single fault propagation," in *Proc. Int. Test Conf.*, Washington, DC, 1986, pp. 542–549.

[9] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyenggar, "Transition fault simulation," *IEEE Des. Test. Comput.*, vol. 4, no. 2, pp. 32–38, 1987.

[10] H. Cox and J. Rajski, "Stuck-open and transition fault testing in CMOS complex gates," in *Proc. Int. Test Conf.*, Washington, DC, 1988, pp. 688–694.

[11] I. Pomeranz and S. M. Reddy, "On diagnosis and diagnostic test generation for pattern-dependent transition faults," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 6, pp. 791–800, Jun. 2001.

[12] R. L. Wadsack, "Fault modeling and logic simulation of CMOS and MOS integrated circuits," *Bell Syst. Tech. J.*, vol. 57, no. 5, pp. 1449–1488, May–Jun. 1978.

[13] J. Soden and R. Treece, "CMOS IC stuck-open fault electrical effects and design considerations," in *Proc. Int. Test Conf.*, Washington, DC, 1989, pp. 423–430.

[14] B. W. Woodhall, B. D. Newman, and A. G. Sammuli, "Empirical results on undetected CMOS stuck-open failures," in *Proc. Int. Test Conf.*, Washington, DC, 1987, pp. 166–170.

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

[16] H. Konuk, "Testing for opens in digital circuits," Ph.D. thesis, Comput. Eng. Dept, Univ. California, Santa Cruz, 1996.

[17] Z. Barzilai *et al.*, "Efficient fault simulation of CMOS circuits with accurate models," in *Proc. Int. Test Conf.*, Washington, DC, 1986, pp. 520–529.

[18] C. Di and J. Jess, "On accurate modeling and efficient simulation of CMOS opens," in *Proc. Int. Test Conf.*, Baltimore, MD, 1993 pp. 857–882.

[19] S. Venkataraman and S. B. Drummonds, "A technique for logic fault diagnosis of interconnect open defects," in *Proc. VLSI Test Symp.*, Montreal, QC, Canada, 2000, pp. 313–318.

[20] C. Hora, W. Beverloo, M. Lousberg, and R. Segers, "On electrical fault diagnosis in full-scan circuits," in *Proc. IEEE Int. Workshop Defect Based Testing*, Los Angeles, CA, 2001, pp. 17–22.

[21] E. J. McCluskey, R. H. Wilcox, and W. C. Mann, *Redundancy Techniques for Computing Systems*. Washington, DC: Spartan, 1962, pp. 9–46.

[22] S. M. Reddy, M. K. Reddy, and J. G. Kuhl, "On testable design for CMOS logic circuits," in *Proc. Int. Test Conf.*, Philadelphia, PA, 1983, pp. 435–444.

[23] P. Franco *et al.*, "An experimental chip to evaluate test techniques chip and experiment design," in *Proc. Int. Test Conf.*, Washington, DC, 1995, pp. 653–662.

[24] E. J. McCluskey and C. W. Tseng, "Stuck-fault vs. actual defects," in *Proc. Int. Test Conf.*, Atlantic City, NJ, 2000, pp. 3343–3356.

[25] J. C.-M. Li, J. T.-Y. Chang, C. W. Tseng, and E. J. McCluskey, "ELF35 Experiment—Chip and Experiment Design," Center for Reliable Computing, Stanford Univ., CA, CRC Tech. Rep., No. 99-3, Oct. 1999.

[26] E. R. Hhnatek, *Digital Integrated Circuit Testing from a Quality Perspective*. New York: Van Nostrand, 1993.

[27] S. Johnson, "Residual charge on the faulty floating gate MOS transistor," in *Proc. Int. Test Conf.*, Washington, DC, 1994, pp. 555–561.

[28] H. Konuk and F. J. Ferguson, "An unexpected factor in testing for CMOS opens: The die surface," in *Proc. VLSI Test Symp.*, Princeton, NJ, 1996, pp. 422–429.

[29] R. C. Aitken, "Diagnosis of leakage faults with IDDQ," *J. Electron. Test.: Theory Appl.*, vol. 3, no. 4, pp. 81-89, 367–375, 1992.

[30] LSI Logic, 0.7-Micron Array-Based Product Databook, pp. 1–20, 1993.

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

[32] J. Chang *et al.*, "Analysis of pattern-dependent and timing dependent failure in an experimental test chip," in *Proc. Int. Test Conf.*, Washington, DC, 1998, pp. 184–193.

[33] E. J. McCluskey, *Logic Design Principles*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[34] I. Koren, Z. Koren, and C. H. Stapper, "A statistical study of defect maps of large area VLSI IC's," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 2, no. 2, pp. 249–256, 1996.

[35] K. C. Y. Mei, "Bridging and stuck-at faults," *IEEE Trans. Comput.*, vol. C-23, no. 7, pp. 720–727, Jul. 1974.

[36] LSI Logic, G10-p Cell-Based ASIC Products, Feb. 1997.

[37] Z. Barzilai and B. K. Rosen, "Comparison of AC self-testing procedures," in *Proc. Int. Test Conf.*, Philadelphia, PA, 1983, pp. 89–94.

[38] S. Chakravarity and A. Jain, "Fault models for speed failures caused by bridges and opens," in *Proc. VLSI Test Symp.*, Monterey, CA, 2002, pp. 373–378.

[39] C. W. Tseng and E. J. McCluskey, "Multiple-output propagation transition fault test," in *Proc. Int. Test Conf.*, Baltimore, MD, 2001, pp. 358–366.

[40] E. B. Eichelberger *et al.*, *Structured Logic Testing*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

James Chien-Mo Li (S'93–M'93) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1993, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1997 and 2002, respectively.

He has been an Assistant Professor with the Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan, since 2002. His research interests include BIST, defect-based testing, and fault diagnosis.

Edward J. McCluskey (S'51–M'55–SM'59–F'65–LF'94) received the A.B. degree (*summa cum laude*) in mathematics and physics from Bowdoin College, Brunswick, Maine, in 1953, and the B.S., M.S., and Sc.D. degrees in electrical engineering from Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1953, 1953, and 1956, respectively. He has honorary doctorates from the University of Grenoble, Grenoble, France, and Bowdoin College.

He worked on electronics switching systems at the Bell Telephone Laboratories from 1955 to 1959. Then, he moved to Princeton University, Princeton, NJ, where he was Professor of Electrical Engineering and Director of the University Computer Center. In 1966, he joined Stanford University, Stanford, CA, where he is a Professor of Electrical Engineering and Computer Science, as well as Director of the Center for Reliable Computing. He founded the Stanford Digital Systems Laboratory (now the Computer Science Laboratory) in 1969 and the Stanford Computer Engineering program (now the Stanford Computer Science M.S. degree program) in 1970. He developed the first algorithm for designing combinational circuits—the Quine–McCluskey logic minimization procedure—as a doctoral student at MIT. At Bell Labs and Princeton, he developed the modern theory of transients (hazards) in logic networks and formulated the concept of operating modes of sequential circuits. His Stanford research focuses on logic testing, synthesis, design for testability, and fault-tolerant computing. He has published several books and book chapters, as well as hundreds of papers. His most recent book is *Logic Design Principles with Emphasis on Testable Semiconductor Circuits* (Englewood Cliffs, NJ: Prentice-Hall, 1986).

Dr. McCluskey served as the first President of the IEEE Computer Society. His most recent honors include election to the National Academy of Engineering, 1998, the 1996 IEEE Emanuel R. Piore Award, and the IEEE Computer Science Golden Core membership. In 1984, he received the IEEE Centennial Medal and the IEEE Computer Society Technical Achievement Award in Testing. The IEEE Computer Society honored him with the 1991 Taylor L. Booth Education Award. He is a Fellow of AAAS and ACM.