Selection of a Fault Model for Fault Diagnosis Based on Unique Responses

Irith Pomeranz[1]
School of Electrical & Computer Eng.
Purdue University
W. Lafayette, IN 47907, U.S.A.

and

Sudhakar M. Reddy[2]
Electrical & Computer Eng. Dept.
University of Iowa
Iowa City, IA 52242, U.S.A.

Abstract - **We describe a preprocessing step to fault diagnosis of an observed response obtained from a faulty chip. In this step, a fault model for diagnosing the observed response is selected. This step allows fault diagnosis to be performed based on a single fault model after identifying the most appropriate one. We describe a specific implementation of this preprocessing step based on what is referred to as the unique output response of a fault model. As an example, we apply it to the diagnosis of multiple stuck-at faults, selecting between single and double stuck-at faults as the fault model for diagnosis. Experimental results demonstrate improvements compared to diagnosis based on single stuck-at faults, and compared to diagnosis based on both single and double stuck-at faults.**

I. INTRODUCTION

For a chip that fails under the application of a test set, fault diagnosis is the process of identifying the likely locations of the defects present in the chip. Fault diagnosis uses a fault model to compute possible output responses of a faulty circuit. The computation can be done during the diagnosis process, or in advance. During diagnosis, the fault diagnosis process compares the observed response of the failing chip to the responses of the circuit in the presence of modeled faults. The defect is assumed to be present at one of the sites of the modeled faults that best matches the observed response. These locations are referred to as candidate fault sites.

Fault models that have been used for fault diagnosis include single stuck-at faults [1]-[3], multiple stuck-at faults [4]-[7], bridging faults [8]-[14], interconnect open faults [15], and delay faults [16]-[19]. Regardless of the fault model used, it is possible that the observed response of a faulty chip will not match the response of any modeled fault accurately. Matching algorithms [20]-[21] are used in this case to identify the best matches among all the modeled faults. Fault diagnosis processes that accommodate more than one fault model were described in [22]-[25]. In [23], a generalized fault model is used that includes the behavior of several different classical fault models. In [22], [24] and [25], stuck-at faults are used as a basis for diagnosis. Deviations from the behavior of stuck-at faults are accommodated by using scoring algorithms that estimate the likelihood of a defect being associated with the site of a stuck-at fault.

All these approaches are motivated by the fact that diagnosis of a given observed faulty response can be significantly more accurate if the correct fault model is used as a basis for diagnosis. When the fault model used for diagnosis does not match the observed response, both matching and scoring algorithms may produce sets of candidate fault sites that are larger than necessary. In addition, the set may not include a location that matches the defect location.

In this work we investigate the following issue related to the selection of a fault model for fault diagnosis. Given an observed response of a faulty chip, we attempt to select a fault model that is likely to produce the most accurate diagnosis results for the given chip. The selection of a fault model is performed as a preprocessing step to fault diagnosis. The fault diagnosis process then uses a single fault model (the one selected by the preprocessing step). To be effective, the criterion for selecting a fault model needs to be simple to compute. Different fault models may then be used for different observed responses to obtain the most accurate diagnosis results in every case.

The selection of a fault model for diagnosis can be performed from a wide variety of fault models, and there is a large variety of properties that can be used to match an observed response with a fault model. We focus on specific fault models and a specific property in this work. The property we consider is referred to as the *unique response* of a fault model. Let us consider two fault models M_1 and M_2 with sets of faults F_1 and F_2, respectively. The unique response of M_1 under a test set T includes output vectors that can be produced by faults in F_1 but not by faults in F_2 in response to T. We denote the unique response of a fault model M_i by U_i. If the observed response of a faulty chip includes more output

1. Research supported in part by SRC Grant No. 2007-TJ-1643.
2. Research supported in part by SRC Grant No. 2007-TJ-1642.

978-3-9810801-5-5/DATE09 © 2009 EDAA

vectors from U_1 than from U_2, this can be used as an indication that fault diagnosis should be performed using M_1 and not M_2. The unique responses are computed once, before any chip is diagnosed, and they are used to select a fault model for every chip that fails the test set.

We consider single stuck-at faults as the fault model M_1, and double stuck-at faults as the fault model M_2. We use multiple stuck-at faults of multiplicity $1 \leq m \leq 5$ to model the response of a faulty chip.

The paper is organized as follows. In Section 2 we define the unique response of a fault model M_i. In Section 3 we describe the selection of a fault model given an observed response that needs to be diagnosed. In Section 4 we present experimental results of fault diagnosis.

II. UNIQUE RESPONSES

We use the following notation to define the unique response of a fault model. All the definitions are given with respect to a test set T.

For a fault model M_i with a set of faults F_i, we denote by $z_{i,j,k}$ the output vector produced by the faulty circuit in the presence of a fault $f_{i,j} \in F_i$ under test vector $t_k \in T$.

Considering all the faults in F_i under the test vector $t_k \in T$, we define $Z_{i,k}$ to be the set of all the output vectors that can be obtained in response to t_k in the presence of all the faults in F_i, i.e., $Z_{i,k} = \{z_{i,j,k} : f_{i,j} \in F_i\}$. $Z_{i,k}$ is obtained by fault simulation of all the faults in F_i under t_k.

The response Z_i of M_i under T is defined as the set of all the sets $Z_{i,k}$, or $Z_i = <Z_{i,k} : t_k \in T>$. The triangular brackets indicate that the order of the sets is important, and $Z_i = <Z_{i,0}\, Z_{i,1}\, \cdots >$.

Next, we consider two fault models M_1 and M_2 with sets of faults F_1 and F_2, respectively, under a test set T. The unique response of M_1 under $t_k \in T$ is defined as $U_{1,k} = Z_{1,k} - Z_{2,k}$. $U_{1,k}$ consists of output vectors produced in response to t_k by faults in F_1, which cannot be produced in response to t_k by faults in F_2. Similarly, we define $U_{2,k} = Z_{2,k} - Z_{1,k}$.

The unique response U_1 of M_1 under T is defined as the set of all the sets $U_{1,k}$, or $U_1 = <U_{1,k} : t_k \in T>$. In a similar way, $U_2 = <U_{2,k} : t_k \in T>$.

For illustration, we consider full-scan ISCAS-89 benchmark circuit $s27$. For F_1 we use the set of single stuck-at faults. There are 32 faults in this set. For F_2 we use the set of double stuck-at faults. There are 520 faults in this set. The test set T consists of six tests, $t_0, t_1, \cdots, t_5$. The sets of output vectors $Z_{1,k}$, for $0 \leq k \leq 5$, are shown in Table 1. These sets are obtained by fault simulation without fault dropping of the faults in F_1 under T. The sets of output vectors $Z_{2,k}$, for $0 \leq k \leq 5$, are shown in Table 2. These sets are obtained by fault simulation without fault dropping of the faults in F_2 under T. For ease of reference we ordered the output vectors in each set $Z_{1,k}$ and $Z_{2,k}$ in increasing binary order.

TABLE 1
OUTPUT RESPONSES FOR F_1

k	$Z_{1,k}$						
0	0100	1000	1001	1011	1100	1101	1110
1	0000	0011	0100	0101	0110	1011	1100
2	0001	1000	1001	1011	1100	1101	
3	0000	0001	0011	0100	0101	1001	
4	0011	0100	1001	1010	1011	1100	1111
5	0001	0010	0011	0100	0111	1011	

TABLE 2
OUTPUT RESPONSES FOR F_2

k	$Z_{2,k}$						
0	0000	0001	0011	0100	0101	0110	1000
	1001	1010	1011	1100	1101	1110	1111
1	0000	0001	0010	0011	0100	0101	0110
	0111	1000	1001	1010	1011	1100	1101
	1110	1111					
2	0000	0001	0011	0100	0101	1000	1001
	1010	1011	1100	1101	1110	1111	
3	0000	0001	0010	0011	0100	0101	0110
	0111	1000	1001	1011	1100	1101	
4	0000	0001	0010	0011	0100	0101	0110
	0111	1000	1001	1010	1011	1100	1101
	1110	1111					
5	0000	0001	0010	0011	0100	0101	0110
	0111	1001	1010	1011	1100	1111	

It should be noted that although the circuit has 32 single stuck-at faults, the maximum number of output vectors in any set $Z_{1,k}$ is seven. For 520 double stuck-at faults, the maximum number of output vectors in any set $Z_{2,k}$ is 16.

We also note that $Z_{1,k} \subset Z_{2,k}$ for $0 \leq k \leq 5$. Therefore, $U_{1,k} = Z_{1,k} - Z_{2,k} = \phi$ for every k. The same result is obtained when single and double stuck-at faults are considered for all the benchmark circuits we experiment with. The sets $U_{2,k}$, computed as $U_{2,k} = Z_{2,k} - Z_{1,k}$, for $s27$ are shown in Table 3.

TABLE 3
UNIQUE OUTPUT RESPONSES FOR F_2

k	$U_{2,k}$						
0	0000	0001	0011	0101	0110	1010	1111
1	0001	0010	0111	1000	1001	1010	1101
	1110	1111					
2	0000	0011	0100	0101	1010	1110	1111
3	0010	0110	0111	1000	1011	1100	1101
4	0000	0001	0010	0101	0110	0111	1000
	1101	1110					
5	0000	0101	0110	1001	1010	1100	1111

For the selection of a fault model we use only U_1 and U_2. This requires that U_1 and U_2 be precomputed and stored. We do not use Z_1 or Z_2, and these sets do not need to be stored. We note that the computation performed to obtain Z_1 and Z_2 also yields fault dictionaries

for M_1 and M_2. However, we do not assume that fault diagnosis is done based on fault dictionaries.

The discussion above can be extended to three or more fault models. For three fault models M_1, M_2 and M_3, it is possible to define $U_{1,k} = Z_{1,k} - Z_{2,k} - Z_{3,k}$ and $U_1 = <U_{1,k} : t_k \in T>$. In a similar way, $U_{2,k} = Z_{2,k} - Z_{1,k} - Z_{3,k}$, $U_2 = <U_{2,k} : t_k \in T>$, $U_{3,k} = Z_{3,k} - Z_{1,k} - Z_{2,k}$ and $U_3 = <U_{3,k} : t_k \in T>$.

III. SELECTING A FAULT MODEL

Based on the sets U_1 and U_2 (or $U_1, U_2, U_3, \cdots$ if the number of fault models is larger than two), we select a fault model for diagnosis of an observed response of a faulty chip as described in this section. We use the following notation.

The observed response of a faulty chip is denoted by $W = <w_k : k \in T>$. Here , w_k is the output vector produced by the chip in response to $t_k \in T$.

Comparing W with U_i, we denote by n_i the number of test vectors t_k for which $w_k \in U_{i,k}$.

For example, we consider a triple stuck-at fault of $s27$ and the test set used for constructing the set U_2 shown in Table 3. The fault produces the output response W = <1010 0100 1010 1010 1010 0010>. We have $w_0 \in U_{2,0}$, $w_1 \notin U_{2,1}$, $w_2 \in U_{2,2}$ and $w_k \notin U_{2,k}$ for $k > 2$. Therefore, $n_2 = 2$ for this fault. As another example, we consider a triple stuck-at fault of $s27$ that produces the output response W = <1001 0100 1001 0100 1100 0001>. None of the output vectors w_k is included in the corresponding set $U_{2,k}$, and $n_2 = 0$ for this fault.

To select between two fault models M_1 and M_2 for diagnosing an observed response W, we use n_1 and n_2 as follows.

- If $n_1 > n_2$, we use M_1.
- If $n_2 > n_1$, we use M_2.
- If $n_1 = n_2$, we use the fault model that has the smaller number of faults.

We note that the observed response W may contain output vectors that are not produced by either one of the fault models. These output vectors do not affect our decision as to which fault model should be used to diagnose the fault. When $n_1 = n_2$, we prefer the fault model with the smaller number of faults since diagnosis using this fault model is expected to be faster.

For three fault models M_1, M_2 and M_3 with unique responses U_1, U_2 and U_3, respectively, we define n_i as before, for i = 1, 2 and 3. We select between the fault models as follows.

- If $n_i > n_j > n_k$, where $<i,j,k>$ is a permutation of <1,2,3>, we use M_i.
- If $n_i = n_j > n_k$, where $<i,j,k>$ is a permutation of <1,2,3>, we use M_i or M_j, whichever one has a smaller number of faults.
- If $n_1 = n_2 = n_3$, we use M_1, M_2 or M_3, whichever one has a smaller number of faults.

IV. EXPERIMENTAL RESULTS OF FAULT DIAGNOSIS

In this section we describe the results of fault diagnosis experiments with two fault models. We first describe the experiments. We then report the results.

A. Fault Diagnosis Experiments

We use single stuck-at faults as the model M_1, and double stuck-at faults as the model M_2. Throughout the discussion we only use single stuck-at faults that belong to the collapsed set of single stuck-at faults. In this way we eliminate the need to consider equivalent faults for the candidates found by the fault diagnosis process. For the test set T we use a compact deterministic test set for single stuck-at faults. We find U_1 and U_2 by fault simulation without fault dropping of T as described in Section 2. With the selected fault models, we always obtain $U_1 = <\phi : t_k \in T>$.

To model observed responses, we use 100 stuck-at faults of multiplicity m, for $1 \le m \le 5$. We perform fault simulation of each fault under T to find the corresponding observed response W. We denote a fault injected to model a faulty observed response by f_{obs}.

Given W, we compute n_1 and n_2 as described in Section 3. With $U_1 = <\phi : t_k \in T>$, we obtain $n_1 = 0$ for every observed response. Based on the rules of Section 3, with $n_1 = 0$, if $n_2 = 0$, we diagnose the faulty response using single stuck-at faults; and if $n_2 > 0$, we diagnose the faulty response using double stuck-at faults.

Let M_i be the fault model selected for fault diagnosis. During fault diagnosis, we compare W to the output response produced in response to T by every fault $f_{i,j} \in F_i$. It is possible that W will not match the output response of any fault $f_{i,j} \in F_i$ perfectly. We use the number of bits where W is equal to the output response of $f_{i,j}$ to decide whether $f_{i,j}$ should be included in the set of candidate faults for W. All the faults in F_i that produce a response with the maximum number of bits equal to the corresponding bits of W are included in the set of candidate faults. We denote the set of candidate faults by C.

Based on f_{obs} and C, we update the following parameters. We say that C provides a perfect match for f_{obs} if one of the components of f_{obs} is included in one of the faults of C. For example, we consider f_{obs} = 20/0,25/1,26/1 of $s27$. Here, g/a is the fault g stuck-at a. Using double stuck-at faults for diagnosis, we obtain C = {8/0,24/1 24/1,25/1 25/1,26/1}. We say that C is a perfect match for f_{obs} since the component 25/1 of f_{obs} is included in the fault 24/1,25/1 in C (it is also included in 25/1,26/1, and 26/1 is also included in this fault). We count the number of perfect matches using a variable

denoted by n_{perf}. This variable is incremented by one every time a perfect match is obtained for a fault f_{obs}.

Perfect matches are important since they provide accurate information about the site of the fault, aiding failure analysis that follows fault diagnosis.

We also use the size of C as an indication of the accuracy of the fault diagnosis process. With a smaller set C, failure analysis has fewer locations to consider. We add the size of C to a variable n_{cand}. We report on the average value of n_{cand} considering all the observed responses.

For comparison, we run two additional fault diagnosis processes. The first fault diagnosis process uses only single stuck-at faults to perform fault diagnosis. This process represents a conventional fault diagnosis process with single stuck-at faults. The second fault diagnosis process uses a set of faults that consists of both single and double stuck-at faults to perform fault diagnosis. This process represents a fault diagnosis process that accommodates multiple fault models. It also provides an upper bound on n_{perf} that will be obtained with the proposed diagnosis process.

B. Results

We applied the fault diagnosis processes to full-scan ISCAS-89, ITC-99 and finite-state machine benchmark circuits. We consider circuits with at least 200 and at most 1000 collapsed single stuck-at faults. This allows us to enumerate all the double stuck-at faults and consider them explicitly during fault diagnosis (efficient fault diagnosis processes can be used instead). The results of fault diagnosis are reported in Tables 4, 5 and 6 as follows.

In Table 4, under column *faults* we show the number of single stuck-at faults (subcolumn *ssa*) and the number of double stuck-at faults (subcolumn *dsa*).

Under column *unique dsa*, subcolumn *vect*, we show the average number of output vectors in a set $U_{2,k}$, for $t_k \in T$. If each double stuck-at fault had produced a different output vector, this number would have been equal to the number of double stuck-at faults. In effect, this number is significantly smaller. In some cases it is also smaller than the number of single stuck-at faults. In these cases, storage of U_2 will require fewer bits than storage of a full fault dictionary for single stuck-at faults. U_2 can be stored in a compacted form to reduce storage requirements.

Under column *unique dsa*, subcolumn *faults*, of Table 4 we show the percentage of double stuck-at faults that produce output vectors, which are not produced by any single stuck-at fault. These are the faults that contribute to U_2. It can be seen that the percentage of such faults is high.

In Tables 5 and 6, under column *ssa* we show the results of fault diagnosis using only single stuck-at faults,

TABLE 4
CIRCUIT PARAMETERS

	faults		unique dsa	
circuit	ssa	dsa	vect	faults
s208	215	22963	77.70	36.33
s298	308	47224	432.12	44.29
s344	342	58245	925.60	45.87
s382	399	79325	722.12	46.09
s420	430	92151	286.98	41.31
s510	564	158680	285.63	33.52
s526	555	153654	827.62	42.04
s641	467	108711	2295.59	53.85
s820	850	360762	481.84	15.07
b03	452	101846	1053.35	60.87
b08	489	119230	667.56	39.95
b09	420	87904	796.33	54.23
b10	512	130718	706.68	45.70
bbsse	238	28161	126.39	30.07
cse	357	63486	181.52	27.83
dk16	532	141188	109.26	24.23
donfile	287	41008	17.86	18.99
dvram	425	90019	750.85	39.39
ex2	312	48473	57.52	24.18
ex6	229	26062	169.42	34.59
fetch	345	59275	533.11	35.26
log	313	48753	1043.21	41.76
nucpwr	447	99584	1460.05	37.82
rie	552	151972	1614.60	37.34

under column *ssa&dsa* we show the results of fault diagnosis using a set that consists of both single and double stuck-at faults, and under column *ssa|dsa* we show the results of the proposed process, that uses either single or double stuck-at faults for each observed response.

In a row with a value m_0 under column m we show the results of fault diagnosis for observed responses that are obtained by injecting stuck-at faults of multiplicity m_0, for $1 \leq m_0 \leq 5$. For every fault diagnosis process, under subcolumn *perf* we show the number of observed responses for which perfect diagnosis was obtained (perfect diagnosis was defined in Subsection 4.A). Under subcolumn *cand* we show the average number of candidate fault sites obtained for the observed responses. In the case of the proposed procedure we also show under subcolumn *ssa* the number of observed responses for which diagnosis is based on single stuck-at faults, and under subcolumn *dsa* the number of observed responses for which diagnosis is based on double stuck-at faults.

The following points can be seen from Tables 5 and 6. When the observed response is that of a single stuck-at fault ($m = 1$), the proposed procedure selects to use the single stuck-at fault model for diagnosis in all the cases considered. This yields perfect fault diagnosis in all the cases.

When the observed response is that of a double stuck-at fault, the proposed procedure may select to use single stuck-at faults for diagnosis if the observed response does not have any of the unique output vectors of a double stuck-at fault. The proposed procedure selects to use the double stuck-at fault model for diagnosis in a percentage of cases that is close to the percentage of dou-

TABLE 5
RESULTS OF FAULT DIAGNOSIS
(ISCAS-89 AND ITC-99)

		ssa		ssa&dsa		ssa|dsa			
circuit	m	perf	cand	perf	cand	ssa	dsa	perf	cand
s208	1	100	1.12	100	17.00	100	0	100	1.12
s208	2	98	1.11	100	4.03	63	37	98	1.18
s208	3	94	1.10	100	1.46	36	64	96	1.34
s208	4	94	1.10	100	1.44	17	83	98	1.40
s208	5	94	1.13	100	1.45	5	95	100	1.45
s298	1	100	1.28	100	20.64	100	0	100	1.28
s298	2	98	1.43	100	4.42	55	45	99	1.59
s298	3	98	1.30	100	1.99	23	77	100	1.75
s298	4	98	1.26	100	2.07	7	93	100	1.99
s298	5	98	1.27	100	1.75	4	96	100	1.70
s344	1	100	1.16	100	20.00	100	0	100	1.16
s344	2	99	1.27	100	4.50	54	46	99	1.37
s344	3	97	1.20	100	1.74	18	82	99	1.60
s344	4	97	1.13	100	1.49	6	94	100	1.49
s344	5	95	1.20	100	1.43	2	98	99	1.44
s382	1	100	1.20	100	30.32	100	0	100	1.20
s382	2	99	1.35	100	2.04	61	39	99	1.36
s382	3	98	1.21	100	2.24	28	72	100	1.40
s382	4	96	1.17	100	1.71	9	91	99	1.60
s382	5	96	1.19	100	1.45	4	96	100	1.42
s420	1	100	1.20	100	29.77	100	0	100	1.20
s420	2	99	1.24	100	4.72	60	40	99	1.33
s420	3	95	1.14	100	5.15	35	65	97	1.36
s420	4	91	1.06	100	2.58	14	86	98	1.35
s420	5	82	1.08	100	2.45	10	90	98	1.27
s510	1	100	1.05	100	38.87	100	0	100	1.05
s510	2	99	1.04	100	4.71	75	25	100	1.06
s510	3	96	1.05	100	1.83	39	61	98	1.10
s510	4	93	1.07	99	1.15	26	74	99	1.13
s510	5	89	1.06	99	1.16	17	83	99	1.13
s526	1	100	1.18	100	8.27	100	0	100	1.18
s526	2	100	1.26	100	3.85	60	40	100	1.35
s526	3	99	1.34	100	1.74	38	62	100	1.55
s526	4	98	1.28	100	1.81	19	81	100	1.66
s526	5	96	1.19	100	1.62	7	93	100	1.53
s641	1	100	1.07	100	26.39	100	0	100	1.07
s641	2	100	1.14	100	2.05	53	47	100	1.21
s641	3	97	1.09	100	1.48	22	78	100	1.25
s641	4	98	1.02	100	1.24	7	93	100	1.09
s641	5	96	1.04	100	1.09	3	97	100	1.09
s820	1	100	1.27	100	85.25	100	0	100	1.27
s820	2	100	1.44	100	5.44	87	13	100	1.49
s820	3	100	1.49	100	7.64	71	29	100	1.57
s820	4	99	1.42	100	2.35	52	48	99	1.62
s820	5	100	1.39	100	2.36	35	65	100	1.59
b03	1	100	1.03	100	15.23	100	0	100	1.03
b03	2	99	1.11	100	1.06	27	73	100	1.09
b03	3	100	1.02	100	1.17	6	94	100	1.16
b03	4	98	1.10	100	1.16	3	97	100	1.16
b03	5	96	1.08	100	4.91	2	98	100	1.18
b08	1	100	1.48	100	31.91	100	0	100	1.48
b08	2	97	1.40	100	3.59	60	40	97	1.45
b08	3	95	1.13	100	3.89	31	69	100	1.59
b08	4	94	1.09	100	2.55	18	82	97	1.32
b08	5	96	1.12	100	5.74	14	86	100	1.43
b09	1	100	1.11	100	15.87	100	0	100	1.11
b09	2	98	1.11	100	6.02	56	44	98	1.13
b09	3	99	1.20	100	5.04	21	79	99	1.45
b09	4	96	1.11	100	5.03	8	92	100	1.37
b09	5	97	1.07	100	4.81	1	99	100	1.24
b10	1	100	1.09	100	23.82	100	0	100	1.09
b10	2	100	1.08	100	1.20	54	46	100	1.10
b10	3	97	1.06	100	1.26	23	77	100	1.20
b10	4	99	1.03	100	1.20	10	90	100	1.17
b10	5	96	1.03	100	1.16	3	97	100	1.15

ble stuck-at faults with unique output responses (shown in the last column of Table 4).

When the observed response is that of a stuck-at fault of multiplicity higher than two, decreasing numbers of single stuck-at faults are selected for diagnosis. This is due to the fact that faults with multiplicity higher than two produce more of the unique output vectors of double stuck-at faults.

Compared to diagnosis based on single stuck-at faults, the proposed procedure achieves perfect diagnosis in more cases due to the decision to use double stuck-at faults in some cases. While diagnosis based on single stuck-at faults produces decreasing numbers of perfect diagnosis results as m is increased, no such reduction is observed for the proposed procedure.

Compared to diagnosis based on single and double stuck-at faults (representing the case where diagnosis is done based on multiple fault models), the proposed procedure produces smaller sets of candidates since it focuses on a single fault model.

V. CONCLUDING REMARKS

We described a procedure for selecting a fault model that is likely to be effective for diagnosing a given observed response of a faulty chip out of a given set of fault models. This procedure can be applied as a preprocessing step to fault diagnosis. After the appropriate fault model is selected, fault diagnosis can proceed with the selected fault model. Such a preprocessing step is important since the accuracy of fault diagnosis can be improved significantly if the correct fault model is used.

We described a specific implementation of this preprocessing step based on the unique output responses of a fault model. Considering two fault models, the unique output response of one model consists of the output vectors that faults of this model can produce, while faults of the other model cannot produce. To select a fault model for an observed response, we found the number of output vectors in the observed response that match the unique response of each fault model. The fault model with the higher number of matches was selected to perform fault diagnosis.

We applied this preprocessing step to the diagnosis of multiple stuck-at faults, selecting between single and double stuck-at faults as the fault model for diagnosis.

REFERENCES

[1] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press, 1995.

[2] J. A. Waicukauski and E. Lindbloom, "Failure Diagnosis of Structured VLSI", IEEE Design and Test of Computers, Aug. 1989, pp. 49-60.

[3] T. Bartenstein, D. Heaberlin, L. Huisman and D. Sliwinski, "Diagnosing Combinational Logic Designs Using the Single Location At-A-Time (SLAT) Paradigm", in Proc. Intl. Test Conf., Oct. 2001, pp. 287-296.

[4] M. Abramovici and M. A. Breuer, "Multiple Fault Diagnosis in Combinational Circuits Based on an Effect-Cause Analysis", IEEE Trans. on Computers, June 1980, pp. 451-460.

TABLE 6
RESULTS OF FAULT DIAGNOSIS
(FINITE-STATE MACHINES)

		ssa		ssa&dsa		ssa|dsa			
circuit	m	perf	cand	perf	cand	ssa	dsa	perf	cand
bbsse	1	100	1.02	100	22.30	100	0	100	1.02
bbsse	2	99	1.07	100	4.35	66	34	100	1.10
bbsse	3	98	1.08	100	3.08	49	51	100	1.10
bbsse	4	94	1.08	100	1.21	22	78	99	1.17
bbsse	5	90	1.09	100	1.20	9	91	100	1.19
cse	1	100	1.07	100	15.22	100	0	100	1.07
cse	2	100	1.13	100	6.44	74	26	100	1.14
cse	3	99	1.14	100	3.31	49	51	100	1.24
cse	4	95	1.15	100	2.36	39	61	99	1.34
cse	5	96	1.10	99	3.57	17	83	99	1.54
dk16	1	100	1.01	100	15.97	100	0	100	1.01
dk16	2	100	1.08	100	9.17	80	20	100	1.10
dk16	3	100	1.19	100	3.30	55	45	100	1.22
dk16	4	100	1.10	100	1.24	31	69	100	1.16
dk16	5	100	1.05	100	1.27	18	82	100	1.13
donfile	1	100	1.26	100	23.21	100	0	100	1.26
donfile	2	99	1.23	100	4.82	84	16	99	1.24
donfile	3	97	1.27	100	2.77	61	39	98	1.47
donfile	4	98	1.12	100	1.66	41	59	98	1.41
donfile	5	96	1.11	100	2.57	26	74	98	1.35
dvram	1	100	1.00	100	30.93	100	0	100	1.00
dvram	2	100	1.12	100	6.96	68	32	100	1.17
dvram	3	100	1.02	100	3.58	45	55	100	1.13
dvram	4	99	1.05	100	2.96	23	77	99	1.16
dvram	5	98	1.00	100	1.31	11	89	99	1.24
ex2	1	100	1.05	100	24.64	100	0	100	1.05
ex2	2	100	1.09	100	6.70	72	28	100	1.09
ex2	3	100	1.10	100	2.07	43	57	100	1.18
ex2	4	99	1.13	100	1.26	31	69	99	1.25
ex2	5	94	1.08	100	1.25	11	89	100	1.24
ex6	1	100	1.13	100	13.12	100	0	100	1.13
ex6	2	99	1.13	100	3.04	65	35	100	1.13
ex6	3	96	1.16	100	2.15	44	56	99	1.29
ex6	4	94	1.16	100	1.48	27	73	97	1.37
ex6	5	93	1.13	100	1.35	11	89	100	1.32
fetch	1	100	1.08	100	17.57	100	0	100	1.08
fetch	2	98	1.07	100	5.23	55	45	99	1.13
fetch	3	97	1.08	100	3.32	33	67	100	1.17
fetch	4	97	1.05	100	1.17	16	84	100	1.13
fetch	5	98	1.06	100	1.14	10	90	100	1.12
log	1	100	1.02	100	12.54	100	0	100	1.02
log	2	99	1.07	100	4.28	60	40	100	1.12
log	3	99	1.03	100	1.53	31	69	100	1.20
log	4	96	1.07	100	1.20	21	79	100	1.16
log	5	95	1.05	99	1.23	14	86	99	1.18
nucpwr	1	100	1.00	100	36.44	100	0	100	1.00
nucpwr	2	99	1.07	100	6.72	60	40	99	1.08
nucpwr	3	100	1.07	100	2.52	36	64	100	1.14
nucpwr	4	97	1.02	100	2.53	17	83	100	1.09
nucpwr	5	97	1.05	100	2.82	10	90	100	1.48
rie	1	100	1.09	100	18.13	100	0	100	1.09
rie	2	98	1.03	100	6.15	56	44	99	1.06
rie	3	100	1.03	100	4.27	27	73	100	1.11
rie	4	100	1.01	100	1.05	12	88	100	1.04
rie	5	97	1.02	100	1.08	6	94	100	1.03

[5] H. Cox and J. Rajski, "A Method of Fault Analysis for Test Generation and Fault Diagnosis" IEEE Trans. on Computer-Aided Design, July 1988, pp. 813-833.

[6] N. Yanagida, H. Takahashi and Y. Takamatsu, "Efficiency Improvements for Multiple Fault Diagnosis of Combinational Circuits", in Proc. Asian Test Symp., Nov. 1994, pp. 82-87.

[7] Z. Wang, M. Marek-Sadowska, K.-H. Tsai and J. Rajski, "Multiple Fault Diagnosis Using n-Detection Tests" in Proc. Intl. Conf. on Computer Design, Oct. 2003, pp. 198-201.

[8] S. D. Millman, E. J. McCluskey and J. M. Acken, "Diagnosing CMOS Bridging Faults with Stuck-At Fault Dictionaries", in Proc. Intl. Test Conf., 1990, pp. 860-870.

[9] S. Chakravarty and Y. Gong, "An Algorithm for Diagnosing Two-Line Bridging Faults in Combinational Circuits", in Proc. Design Autom. Conf., June 1993, pp. 520-524.

[10] B. Chess, D. B. Lavo, F. J. Ferguson and T. Larrabee, "Diagnosis of Realistic Bridging Faults with Single Stuck-At Information", in Proc. Intl. Conf. on Computer-Aided Design, Nov. 1995, pp. 185-192.

[11] R. C. Aitken and P. C. Maxwell, "Better Models or Better Algorithms ? Techniques to Improve Fault Diagnosis", HP Journal, Feb. 1995, pp. 110-116.

[12] R. C. Aitken, "A Comparison of Defect Models for Fault Location with I_{ddq} Measurements", in Proc. Intl. Test Conf., Sept. 1992, pp. 778-787.

[13] S. Chakravarty and M. Liu, "Algorithms for Current Monitor Based Diagnosis of Bridging and Leakage Faults", in Proc. Design Autom. Conf., 1992, pp. 353-356.

[14] J. Saxena, K. M. Butler, H. Balachandran, D. B. Lavo, B. Chess, T. Larrabee and F. J. Ferguson, "On Applying Non-Classical Defect Models to Automated Diagnosis", in Proc. Intl. Test Conf., Oct. 1998, pp. 748-757.

[15] S. Venkataraman and S. B. Drummonds, "A Technique for Logic Fault Diagnosis of Interconnect Open Defects", in Proc. VLSI Test Symp., April 2000, pp. 313-318.

[16] J. Davies Lesser and J. J. Shedletsky, "An Experimental Delay Test Generator for LSI Logic", IEEE Trans. on Computers, March 1980, pp. 235-248.

[17] P. Girard, C. Landrault and S. Pravossoudovitch, "A Novel Approach to Delay-Fault Diagnosis", in Proc. Design Autom. Conf., 1992, pp. 357-360.

[18] M. Sivaraman and A. J. Strojwas, "A Diagnosability Metric for Parametric Path Delay Faults", in Proc. VLSI Test Symp., April 1996, pp. 316-322.

[19] Y.-C. Hsu and S. K. Gupta, "A New Path-Oriented Effect-Cause Methodology to Diagnose Delay Failures", in Proc. Intl. Test Conf., 1998, pp. 758-767.

[20] J. Richman and K. R. Bowden, "The Modern Fault Dictionary", in Proc. Intl. Test Conf., 1985, pp. 696-702.

[21] P. G. Ryan, S. Rawat and W. K. Fuchs, "Two-Stage Fault Location", in Proc. Intl. Test Conf., 1991, pp. 963-968.

[22] D. B. Lavo, B. Chess, T. Larrabee and I. Hartanto, "Probabilistic Mixed-Model Fault Diagnosis", In Proc. Intl. Test Conference, Oct. 1998, pp. 1084-1093.

[23] R. D. Blanton, J. T. Chen, R. Desineni, K. N. Dwarakanath, W. Maly and T. J. Vogels, "Fault Tuples in Diagnosis of Deep-Submicron Circuits", in Proc. Intl. Test Conf., 2002, pp. 233-241.

[24] S. Venkataraman and S. B. Drummonds, "POIROT: A Logic Fault Diagnosis Tool and its Applications", in Proc. Intl. Test Conf., 2000, pp. 253-262.

[25] S. Holst and H.-J. Wunderlich, "Adaptive Debug and Diagnosis without Fault Dictionaries", in Proc. European Test Symp., 2007, pp. 7-12.