A DELAY FAULT MODEL FOR AT-SPEED FAULT SIMULATION AND TEST GENERATION

Irith Pomeranz
School of Electrical & Computer Eng.
Purdue University
W. Lafayette, IN 47907
pomeranz@ecn.purdue.edu

Sudhakar M. Reddy
Electrical & Computer Eng. Dept.
University of Iowa
Iowa City, IA 52242
reddy@engineering.uiowa.edu

Abstract

We describe a transition fault model, which is easy to simulate under test sequences that are applied at-speed, and provides a target for the generation of at-speed test sequences. At-speed test application allows a circuit to be tested under its normal operation conditions. However, fault simulation and test generation for the existing fault models become significantly more complex due to the need to handle faulty signal-transitions that span multiple clock cycles. The proposed fault model alleviates this shortcoming by introducing unspecified values into the faulty circuit when fault effects may occur. Fault detection potentially occurs when an unspecified value reaches a primary output. Due to the uncertainty that an unspecified value propagated to a primary output will be different from the fault free value, an inherent requirement in this model is that a fault would be potentially detected multiple times in order to increase the likelihood of detection. Experimental results demonstrate that the model behaves as expected in terms of fault coverage and numbers of detections of target faults. A variation of an n-detection test generation procedure for stuck-at faults is used for generating test sequences under this model.

1. INTRODUCTION

Application of tests for delay faults in synchronous sequential circuits can be done in one of several ways. Scan can be used to apply two-pattern tests that start and end with scan operations [1]-[3]. Tests can also be applied using only the functional mode of operation of the circuit. In this case it is possible to use slow clock cycles for initialization and fault propagation and fast clock cycles for capturing fault effects [4]-[6]. Alternatively, a test sequence can be applied at-speed [7]-[11] using only fast clock cycles. At-speed test application has the advantage that the circuit is tested under its normal operation conditions. It has been shown that certain defects will only be detected if tests are applied at-speed. In addition, as demonstrated in [12], test application that deviates from normal operation can cause faulty behavior that would not show up during normal operation.

Transition faults are used for their simplicity in modeling spot defects that affect delays at inputs or outputs of gates. Under scan based tests transition faults are associated with an extra delay that is large enough to cause the delay of any path through the fault site to exceed the clock period. Beyond this assumption, the specific delay size is not important. When at-speed tests are used, a faulty line is considered under multiple consecutive fast clock cycles. In this case, it becomes necessary to explicitly consider defect sizes measured in numbers of clock cycles in order to determine the value of a faulty line in consecutive fast clock cycles [9]. Thus, it is necessary to consider each transition fault multiple times, associating with it a delay of size 1, 2, $\cdots$ cycles. This increases the complexity of fault simulation and test generation.

In this work we propose a new delay fault model similar to the transition fault model for use with at-speed tests. The model allows simulation of a given transition fault only once to determine whether it is detected by a given test sequence, and it allows test generation similar to the generation of n-detection test sequences for stuck-at faults. The model has the following features.

(1) The fault simulation process for the proposed model is similar in complexity to fault simulation of stuck-at faults.

(2) The conditions for fault activation are simple to compute. When a fault is activated, the value assigned to the fault site in the faulty circuit is unspecified (x). Unspecified values have several properties that make them suitable for at-speed fault simulation as discussed later.

(3) Each time unit where an unspecified value appears on a primary output is counted as a potential detection of the fault. Due to the uncertainty that the unspecified value will be different from the fault free value, a requirement in this model is that a fault would be detected multiple times.

ICCAD'06, November 5-9, 2006, San Jose, CA

(4) The model is shown to be significantly different from the stuck-at fault model by comparing stuck-at faults and transition faults that are *related* to each other [18]. Experimental results show many cases where a stuck-at fault is detected more times than the related transition fault. The opposite happens in fewer cases. The conclusion is that even n-detection test generation for stuck-at faults will not be sufficient for a good coverage of transition faults during at-speed test application.

We assume that at most one transition fault (slow-to-rise or slow-to-fall) will occur on each line. The model can be extended to deal with the case where both faults may be present on a line by introducing unspecified values for both types of transitions.

The paper is organized as follows. In Section 2 we introduce the proposed transition fault model. In Section 3 we present the results of fault simulation of test sequences generated for stuck-at faults. The results demonstrate that the model behaves as expected in terms of fault coverage and numbers of detections of target faults. In Section 4 we discuss test generation for the proposed fault model. We use a variation of n-detection test generation for stuck-at faults that takes into account the need to increase the numbers of detections of stuck-at faults that are related to undetected transition faults.

2. DELAY FAULT MODEL

In this section we describe the proposed delay fault model. The model is similar to the transition fault model, but it is more suitable for at-speed fault simulation and test generation. We refer to faults of the new fault model as *unspecified transition faults*.

Similar to standard transition faults, we associate an unspecified transition fault with every line g and signal-transition $v \rightarrow v'$, for $v \in \{0,1\}$. The fault associated with line g and signal-transition $v \rightarrow v'$ is denoted by $g : v \rightarrow v'$. Similar to a standard transition fault, the unspecified transition fault $g : v \rightarrow v'$ is activated at time unit $u+1$ if $g = v$ at time unit u and $g = v'$ at time unit $u+1$. However, when the fault is activated we set the value of g in the faulty circuit to the unspecified value x instead of setting g to the value v. The unspecified value on g can then be propagated to the next time units until it eventually disappears. For every time unit where an unspecified value is propagated to a primary output we say that the fault is detected once. The detection is viewed as a potential detection to accommodate the following effects.

(1) The duration of the fault is unknown and different durations may result in different values. As a result, a fault of a certain duration may be detected while a fault of a different duration may not be detected by a given test sequence [9]-[10].

(2) Unmodeled delay effects may speed up a transition and cause the effects of a transition fault not to appear under certain conditions [13]-[17].

For an unspecified transition fault, the higher the number of potential detections, the more likely it is that a defect associated with the same site will actually be detected. Therefore, fault simulation and test generation procedures must consider the numbers of times faults are detected, similar to n-detection fault simulation and test generation procedures.

We illustrate the model and the fault simulation process for it by using the example circuit shown in Figure 1. The input sequence under consideration is 00 10 00 10 10 10 00 10. The circuit is assumed to be initialized to state 0 before the application of the input sequence. The fault under consideration is $g : 1 \rightarrow 0$. The values throughout the fault free and faulty circuits are shown in Figure 2. Y and z are combined in Figure 2 since they assume the same values at all the time units under the fault $g : 1 \rightarrow 0$. Values are shown in Figure 2 in the form fault-free/faulty. The following points should be noted.

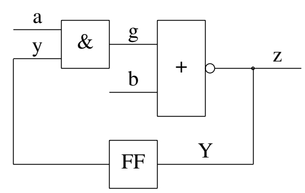

Figure 1: Example circuit

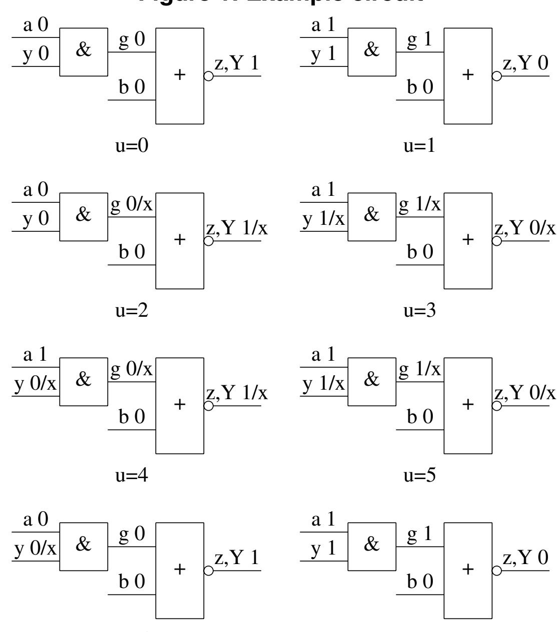

Figure 2: Example circuit with a fault

(1) Since $g = 1$ at time unit 1 and $g = 0$ at time unit 2, the fault is activated at time unit 2. This results in the value $g = 0/x$ at time unit 2.

(2) The fault effect is propagated to the output z at time unit 2. Thus, the fault is detected for the first time.

(3) At time unit 3, the fault effect continues to propagate. The value $1/x$ on g is a result of the value $1/x$ on y at time unit 3. The fault is detected a second time.

(4) If the fault effect had disappeared by time unit 3 and fault free values had been obtained at time units 3 and 4, the fault would have been injected again at time unit 4 (this would have been based on $g = 1$ at time unit 3 and $g = 0$ at time unit 4). The unspecified value on g at time unit 4 accommodates this case.

(5) In this example, for every time unit u where the fault is potentially detected since an x is propagated to the output, there is a transition on the output in the fault free circuit. The transition occurs between time units $u-1$ and u. In general, an x value can also be propagated to an output at time unit u when the fault free value of the output is the same at time units $u-1$ and u.

(6) The values $g = 1/x$ at time unit 5 and $g = 0$ at time unit 6 do not cause the fault to be activated again at time unit 6. This is a result of the fact that $g = x$ in the faulty circuit at time unit 5 may not support fault activation at time unit 6.

The importance of point 6 results from the following observations. Unspecified values have the property that injecting a new unspecified value cannot mask an unspecified value injected earlier. A new unspecified value can only increase the likelihood that an unspecified value will be propagated to an output. We introduce an unspecified value into the faulty circuit only when the faulty line has specified values in two consecutive time units. Since the effects of unspecified values add up, by not injecting unspecified values under certain conditions, we may in effect be reducing the number of time units where a fault will be considered potentially detected. As a result, we may be computing a pessimistic estimate of potential fault detections.

The use of unspecified values to mark fault activation and propagation has the following shortcoming. Simulation of unspecified values using three-value logic has an inherent loss of accuracy that may result in an output being unspecified even though more accurate simulation would indicate that the output can only be 0, or only be 1. However, this effect is small, and it is tolerated in most fault simulation and test generation procedures for synchronous sequential circuits that use three-value logic.

3. FAULT SIMULATION

We implemented a fault simulation procedure for the unspecified transition fault model. In this procedure we simulate a fault until it is detected n times, for a constant n. For comparison, we also performed n-detection fault simulation of stuck-at faults. In this process, a stuck-at fault is considered detected at a time unit u if there exists an output with different specified fault free and faulty values at time unit u. A fault is dropped from consideration after it is detected at n different time units. For further comparison we also simulated transition faults with an extra delay of a single clock cycle.

The test sequences we simulated are compacted deterministic test sequences generated for stuck-at faults. In every case the circuit is started from the all-zero state. This is done to eliminate unspecified values that occur due to the initial state of the circuit. It is possible to accommodate an unspecified initial state by starting the simulation of unspecified transition faults only after the fault free and faulty circuit states are specified, and ignoring fault activations and fault detections that occur earlier.

The results obtained using $n = 5$ are shown in Table 1. Under column $flts$ we show the number of faults (the number of uncollapsed single stuck-at faults, which is equal to the number of transition faults). Under column len we show the length of the test sequence simulated. Under column $model$ we show the fault model being simulated, either single stuck-at faults (row $s.a.$) or unspecified transition faults (row $xtrans$). Under column $f.c.$ we show the fault coverage obtained (this is the number of faults detected at least once as a percentage of the total number of faults). Under column ave we show the average number of times a fault is detected by the test sequence. Under column $d = d_0$, for $d_0 = 0,1, \cdots ,5$, we show the number of faults detected d_0 times. Under column $1trans$ we show the fault coverage of transition faults with an extra delay of a single clock cycle.

From Table 1 it can be seen that the coverage of unspecified transition faults is lower than the coverage of stuck-at faults, and that the numbers of detections of unspecified transition faults are lower as well. The fault coverage is typically somewhat higher than that obtained for transition faults with an extra delay of a single clock cycle. This is consistent with the fact that unspecified transition faults are meant to capture transition faults of different sizes. These results indicate that the unspecified transition fault model behaves as expected, and similar to other delay fault models.

For the purpose of test generation, it is also interesting to see the correlation between the number of detections of an unspecified transition fault $g:v \rightarrow v'$ and the number of detections of the related stuck-at fault g stuck-at v. We say that the faults are related since detection of both faults occurs when the fault changes the value of g from v' to a faulty value (a similar relationship exists between standard transition faults and stuck-at faults [18]). For example, we would like to know whether a high (low) number of detections for g stuck-at v implies a high (low) number of detections for $g:v \rightarrow v'$. If the correlation between the numbers of detections is high, then test generation for unspecified transition faults can be replaced with n-detection test generation for stuck-at faults.

Table 1: Results of simulation

circuit	flts	len	model	f.c.	ave	d=0	d=1	d=2	d=3	d=4	d=5	1trans
s208	416	105	s.a.	70.19	2.32	124	72	33	36	35	116	51.68
			xtrans	51.68	1.47	201	74	41	14	17	69	
s298	596	117	s.a.	89.60	4.20	62	10	30	3	29	462	69.97
			xtrans	69.97	3.05	179	32	26	21	22	316	
s344	688	57	s.a.	97.38	4.69	18	7	12	19	19	613	85.47
			xtrans	85.47	3.99	100	18	17	16	39	498	
s382	764	416	s.a.	96.60	4.66	26	11	16	12	11	688	73.43
			xtrans	74.08	3.18	198	61	30	18	34	423	
s386	772	121	s.a.	90.16	3.89	76	55	48	39	34	520	67.49
			xtrans	67.49	2.73	251	69	56	16	22	358	
s400	800	611	s.a.	95.38	4.67	37	7	11	4	13	728	72.38
			xtrans	72.38	3.18	221	39	41	23	23	453	
s420	840	108	s.a.	46.79	1.76	447	71	32	41	26	223	27.86
			xtrans	27.98	0.91	605	83	19	8	6	119	
s510	258	1020	s.a.	100.00	4.95	0	3	2	11	9	995	85.39
			xtrans	85.39	4.17	149	5	1	29	17	819	
s526	1052	1006	s.a.	86.88	4.29	138	4	1	9	16	884	59.03
			xtrans	59.13	2.71	430	11	51	19	20	521	
s641	1280	101	s.a.	88.12	4.07	152	63	37	30	10	988	75.16
			xtrans	75.16	3.32	318	71	46	55	27	763	
s820	1640	491	s.a.	96.34	4.44	60	67	61	60	52	1340	74.33
			xtrans	74.51	3.43	418	55	50	47	27	1043	
s953	267	1906	s.a.	99.37	4.56	12	72	92	78	55	1597	88.20
			xtrans	88.20	3.72	225	159	125	120	71	1206	
s1196	2392	238	s.a.	99.87	3.82	3	275	368	202	196	1348	81.44
			xtrans	81.56	2.54	441	466	409	225	137	714	
s1423	2846	1024	s.a.	96.94	4.34	87	182	137	78	140	2222	81.66
			xtrans	83.24	4.05	477	34	31	39	9	2256	
s5378	10590	646	s.a.	80.34	3.81	2082	291	206	164	101	7746	71.67
			xtrans	71.72	3.37	2995	286	208	215	105	6781	
s35932	71864	150	s.a.	89.78	4.48	7344	18	43	108	43	64308	86.50
			xtrans	86.50	4.29	9704	379	183	198	146	61254	
b03	768	130	s.a.	74.22	3.42	198	9	35	31	21	474	54.17
			xtrans	54.17	2.38	352	10	42	24	42	298	
b04	2284	168	s.a.	88.66	4.13	259	76	68	69	49	1763	71.94
			xtrans	72.42	2.66	630	191	279	230	126	828	
b09	678	269	s.a.	84.81	3.92	103	34	20	9	2	510	64.31
			xtrans	63.72	2.92	246	36	5	5	10	376	
b10	870	190	s.a.	93.33	4.38	58	25	43	10	0	734	70.00
			xtrans	70.57	3.21	256	37	35	8	4	530	
b11	1830	675	s.a.	92.40	4.57	139	7	13	5	7	1659	75.57
			xtrans	75.68	3.71	445	8	24	5	13	1335	

In Table 2 we show detailed information about numbers of detections for $s298$. For every pair (g,v), let $n_{xtrans}(g,v)$ be the number of times the unspecified transition fault $g:v \rightarrow v'$ is detected, and let $n_{sa}(g,v)$ be the number of times the stuck-at fault g stuck-at v is detected. The pair (g,v) contributes to the entry in row $n_{sa}(g,v)$ and column $n_{xtrans}(g,v)$ of Table 2. Thus, the entry in row i and column j of Table 2 provides the number of pairs (g,v) such that $n_{sa}(g,v)=i$ and $n_{xtrans}(g,v)=j$.

From Table 2 it can be seen that there are cases where $n_{sa}(g,v) < n$ and $n_{xtrans}(g,v) < n$. In these cases, n-detection test generation for stuck-at faults may help increase the numbers of detections of unspecified transition faults as well. However, there are also cases where $n_{sa}(g,v) = n$ and $n_{xtrans}(g,v) < n$. In these cases, it may be necessary to increase the numbers of detections of stuck-at faults that are already detected n times in order to potentially increase the numbers of detections of the related unspecified transition faults. We investigate a variation of an n-detection test generation procedure for stuck-at faults as a way to increase the numbers of detections of unspecified transition faults in the next section.

Table 2: Numbers of detections

n_{sa} \ n_{xtrans}	0	1	2	3	4	5
0	62	0	0	0	0	0
1	2	8	0	0	0	0
2	17	2	7	0	1	3
3	2	0	0	1	0	0
4	11	6	1	9	1	1
5	85	16	18	11	20	312

4. TEST GENERATION

In this section we describe a test generation procedure for unspecified transition faults. The procedure attempts to increase the numbers of detections of unspecified transition faults, which are detected fewer than n times by a given test sequence, for a constant n.

Based on the discussion of the previous section, we start from a test sequence T for stuck-at faults. We increase the numbers of detections of stuck-at faults by adding test subsequences to T in order to indirectly increase the numbers of detections of unspecified transition faults.

For $d = 0,1, \cdots ,n-1$, we consider every unspecified transition fault $g:v \rightarrow v'$ such that

$n_{sa}(g,v) > 0$ and $n_{xtrans}(g,v) = d$. The reason for requiring $n_{sa}(g,v) > 0$ is that we will use a subsequence that detects the stuck-at fault g stuck-at v in order to generate a subsequence that potentially detects the related unspecified transition fault $g:v \rightarrow v'$.

For every value of d we consider all the faults with $n_{sa}(g,v) > 0$ and $n_{xtrans}(g,v) = d$. For $d > 0$ we then consider all the faults with $n_{sa}(g,v) > 0$ and $n_{xtrans}(g,v) = d$ again. This is done in case a fault with $n_{xtrans}(g,v) = d-1$ is accidentally detected, and a test subsequences for it may be generated if it is targeted again directly.

When we consider $g:v \rightarrow v'$, we obtain a new test subsequence $\hat{T}$ that detects g stuck-at v and concatenate it to T. The new test subsequence $\hat{T}$ for g stuck-at v is obtained from T as follows.

We simulate g stuck-at v under the test sequence T starting from the all-unspecified state. If g stuck-at v is not detected by T, we do not consider it further. Otherwise, we find the smallest time unit u_e where g stuck-at v is detected by T. Denoting the subsequence of T that starts at time unit u_s and ends at time unit u_e by $T[u_s,u_e]$, we find in this step a subsequence $T[0,u_e]$ that detects g stuck-at v starting from the all-unspecified state.

We then consider decreasing values of u_s, $u_s = u_e, u_{e-1}, \cdots, 0$. We simulate the subsequence $T[u_s,u_e]$ of T that starts at time unit u_s and ends at time unit u_e starting from the all-unspecified state. We stop with the highest value of u_s such that $T[u_s,u_e]$ detects g stuck-at v. Since $T[u_s,u_e]$ detects g stuck-at v starting from the all-unspecified state, concatenating $T[u_s,u_e]$ to T is guaranteed to increase the number of detections of g stuck-at v. To ensure that different test subsequences are obtained when g stuck-at v is considered multiple times, we add the following two steps.

Considering the time units of $T[u_s,u_e]$ in a random order, we attempt to omit each test vector from $T[u_s,u_e]$. When time unit u is considered, we omit the vector $T[u]$ at time unit u of $T[u_s,u_e]$. If g stuck-at v continues to be detected, we accept the omission. Otherwise, we restore $T[u]$ into $T[u_s,u_e]$.

After the vector omission step is complete, we randomly decide whether or not to try and change every bit b of $T[u_s,u_e]$. When bit b is considered, if the decision is to try and change it, we complement the bit and simulate g stuck-at v. If the fault is not detected, we complement the bit again to restore its initial value. Otherwise, we leave the bit complemented.

For illustration we consider the test sequence of $s27$ shown in Table 3 under column *initial*. We consider a stuck-at fault f that is detected once by T, while the related unspecified transition fault is not detected by T. We find that f is detected by T at time unit $u_e = 8$. Considering $u_s = 8,7,\cdots$, we find that $T[8,8] = 0000$ does not detect f, $T[7,8] = 0000\ 0000$ does not detect f, and so on, until $T[5,8] = 1011\ 1001\ 0000\ 0000$ detects f. We set $\hat{T} = T[5,8] = 1011\ 1001\ 0000\ 0000$.

We randomly order the time units of $\hat{T}$ in the order <1,3,0,2>. We find that the vector 1001 at time unit 1 of $\hat{T}$ can be omitted, but the remaining vectors cannot be omitted. The resulting test subsequence is $\hat{T} = 1011\ 0000\ 0000$.

We randomly decide to try and complement bits 0, 1, 2 and 3 at time unit 0 of $\hat{T}$, bit 0 at time unit 1 of $\hat{T}$, and bits 0, 1 and 2 at time unit 2 of $\hat{T}$. We find that bits 0, 1 and 3 at time unit 0 of $\hat{T}$, bit 0 at time unit 1 of $\hat{T}$, and bits 0 and 2 at time unit 2 of $\hat{T}$ can be complemented. The resulting test subsequence is $\hat{T} = 0110\ 1000\ 1010$. After concatenating $\hat{T}$ to T we obtain the test sequence shown in Table 3 under column *extended*.

Table 3: Test sequence for $s27$

	$T[u]$	
u	initial	extended
0	0111	0111
1	1001	1001
2	0111	0111
3	1001	1001
4	0100	0100
5	1011	1011
6	1001	1001
7	0000	0000
8	0000	0000
9	1011	1011
10		0110
11		1000
12		1010

The second time the same fault g stuck-at v is considered, the test subsequence $T[5,8] = 1011\ 1001\ 0000\ 0000$ is found again. This time, the order of omission is set to <2,3,0,1>. Again, only the vector at time unit 1 is omitted to obtain $\hat{T} = 1011\ 0000\ 0000$. We randomly decide to try and complement bits 2 and 3 at time unit 0, bits 1 and 3 at time unit 1, and bits 2 and 3 at time unit 2. Bit 3 at time unit 0, bits 1 and 3 at time unit 1, and bits 2 and 3 at time unit 2 are complemented without losing the detection of the fault. The resulting test subsequence is $\hat{T} = 1010\ 0101\ 0011$, and it is concatenated to T. This test subsequence is different from the one extracted before for the same fault.

We note that even if the same test subsequence $\hat{T}$ is extracted and added to T, the state before the application of $\hat{T}$ may be different for the two appearances of $\hat{T}$ in T, contributing to different detection conditions of the fault. Experiments reported in [19] indicate that counting detections of a fault as different if they occur in different time units is as effective as using more complex definitions that require stricter conditions.

After concatenating a test subsequence $\hat{T}$ for the fault g stuck-at v to T, we simulate the unspecified transition fault $g:v \rightarrow v'$. If the number of detections of $g:v \rightarrow v'$ does not increase, we remove $\hat{T}$ from T in order not to increase the length of T unnecessarily.

Results of test generation for the circuits of Table 1 are reported in Table 4. For circuits that are synchronizable using three-value logic, a test subsequence $\hat{T}$ is extracted such that it would detect the target fault starting from the all-unspecified initial state. For other circuits ($s510$ and $s953$), a test subsequence is extracted such that it would detect the target fault starting from the final state of the current test sequence T.

We show in Table 4 the following parameters before test generation (subcolumn *init*) and after test generation (subcolumn *tg*). The test length is shown under column *len*. The coverage of stuck-at faults is shown under column *f.c. s.a.*. The coverage of unspecified transition faults is shown under column *f.c. xtrans*. The average numbers of detections of stuck-at faults is shown under column *ave s.a.*. The average numbers of detections of unspecified transition faults is shown under column *ave xtrans*.

Increases in the stuck-at fault coverage due to test generation are possible in Table 4 since the initial test sequences used were generated assuming an unknown initial state, while we assume that the circuit is initialized to the all-0 state before the test sequence is applied.

In Table 5 we show the numbers of detections of unspecified transition faults before and after test generation. For every $d = d_0$, where $d_0 = 0,1, \cdots ,5$, we show the number of unspecified transition faults that are detected d times. The first row for every circuit shows the numbers of detections before test generation, and the second row shows the numbers of detections after test generation.

From Table 4, test generation increases the coverage of detected unspecified transition faults, and their average numbers of detections. From Table 5, relatively few unspecified transition faults remain, that are detected between one and four times.

Table 4: Results of test generation

	len		f.c. s.a.		f.c. xtrans		ave s.a.		ave xtrans	
circuit	init	tg	init	tg	init	tg	init	tg	init	tg
s208	105	364	70.19	70.19	51.68	53.37	2.32	3.42	1.47	2.53
s298	117	387	89.60	89.77	69.97	71.81	4.20	4.46	3.05	3.53
s344	57	186	97.38	97.38	85.47	89.68	4.69	4.87	3.99	4.43
s382	516	1352	96.60	96.86	74.08	75.65	4.66	4.78	3.18	3.64
s386	121	500	90.16	90.16	67.49	74.48	3.89	4.43	2.73	3.54
s400	611	1813	95.38	95.50	72.38	74.25	4.67	4.74	3.18	3.63
s420	108	368	46.79	46.79	27.98	28.21	1.76	2.29	0.91	1.33
s510	258	343	100.00	100.00	85.39	86.08	4.95	4.99	4.17	4.29
s526	1006	3068	86.88	86.88	59.13	61.60	4.29	4.33	2.71	3.05
s641	101	309	88.12	88.12	75.16	78.67	4.07	4.29	3.32	3.73
s820	491	1516	96.34	96.34	74.51	80.61	4.44	4.73	3.43	3.94
s953	267	1066	99.37	99.37	88.20	93.60	4.56	4.96	3.72	4.64
s1196	238	906	99.87	99.87	81.56	96.95	3.82	4.87	2.54	4.53
s1423	1024	3122	96.94	97.22	83.24	85.07	4.34	4.73	4.05	4.21
s5378	646	2489	80.34	80.34	71.72	73.00	3.81	3.95	3.37	3.57
s35932	150	1447	89.78	89.78	86.50	87.19	4.48	4.49	4.29	4.36
b03	130	380	74.22	74.22	54.17	54.69	3.42	3.62	2.38	2.70
b04	168	917	88.66	88.66	72.42	78.72	4.13	4.41	2.66	3.70
b09	269	1338	84.81	84.81	63.72	65.63	3.92	4.14	2.92	3.27
b10	190	550	93.33	93.33	70.57	76.78	4.38	4.64	3.21	3.77
b11	675	3641	92.40	92.46	75.68	77.49	4.57	4.62	3.71	3.86

Faults that remain uncovered ($d = 0$) are expected to be undetectable, while faults with five detections may actually have much higher numbers of detections, and defects at these sites are expected to be detected. The small numbers of faults that remain with one to four detections can be targeted directly. The approximately five time increase in test length is consistent with the requirement to detect each fault five times ($n = 5$).

To further demonstrate that n-detection test generation for stuck-at faults is not sufficient for ensuring the detection of unspecified transition faults, we compare in Table 6 the numbers of detections of unspecified transition faults before test generation, after n-detection test generation for stuck-at faults, and after test generation as proposed here. We consider several circuits for this comparison. The numbers of detections before test generation are shown in row *init*. The numbers of detections after test generation for stuck-at faults are shown in row *sa*. The numbers of detections after test generation as proposed here are shown in row *xtr*.

Table 5: Numbers of detections after test generation

circuit		d=0	d=1	d=2	d=3	d=4	d=5
s208	init	201	74	41	14	17	69
	tg	194	10	3	4	1	204
s298	init	179	32	26	21	22	316
	tg	168	5	3	2	2	416
s344	init	100	18	17	16	39	498
	tg	71	5	3	0	6	603
s382	init	198	61	30	18	34	423
	tg	186	7	23	4	5	539
s386	init	251	69	56	16	22	358
	tg	197	19	7	16	11	522
s400	init	221	39	41	23	23	453
	tg	206	8	9	0	6	571
s420	init	605	83	19	8	6	119
	tg	603	5	11	7	1	213
s510	init	149	5	1	29	17	819
	tg	142	2	1	0	0	875
s526	init	430	11	51	19	20	521
	tg	404	7	0	2	1	638
s641	init	318	71	46	55	27	763
	tg	273	31	25	25	13	913
s820	init	418	55	50	47	27	1043
	tg	318	18	10	14	12	1268
s953	init	225	159	125	120	71	1206
	tg	122	5	9	9	13	1748
s1196	init	441	466	409	225	137	714
	tg	73	58	97	71	88	2005
s1423	init	477	34	31	39	9	2256
	tg	425	11	10	14	8	2378
s5378	init	2995	286	208	215	105	6781
	tg	2859	114	91	55	44	7427
s35932	init	9704	379	183	198	146	61254
	tg	9205	3	3	1	5	62647
b03	init	352	10	42	24	42	298
	tg	348	0	3	4	8	405
b04	init	630	191	279	230	126	828
	tg	486	41	57	63	79	1558
b09	init	246	36	5	5	10	376
	tg	233	0	1	1	0	443
b10	init	256	37	35	8	4	530
	tg	202	6	6	7	7	642
b11	init	445	8	24	5	13	1335
	tg	412	1	3	9	0	1405

Table 6: Comparison with stuck-at test generation

circuit		len	ave	d=0	d=1	d=2	d=3	d=4	d=5
s298	init	117	3.05	179	32	26	21	22	316
	sa	406	3.49	168	6	1	12	8	401
	xtr	387	3.53	168	5	3	2	2	416
s344	init	57	3.99	100	18	17	16	39	498
	sa	116	4.38	75	5	4	7	6	591
	xtr	186	4.43	71	5	3	0	6	603
s382	init	516	3.18	198	61	30	18	34	423
	sa	1219	3.58	187	19	12	13	10	523
	xtr	1352	3.64	186	7	23	4	5	539
s400	init	611	3.18	221	39	41	23	23	453
	sa	1120	3.36	211	37	20	22	4	506
	xtr	1813	3.63	206	8	9	0	6	571
s526	init	1006	2.71	430	11	51	19	20	521
	sa	2020	2.95	419	2	2	16	17	596
	xtr	3068	3.05	404	7	0	2	1	638
s641	init	101	3.32	318	71	46	55	27	763
	sa	370	3.86	253	18	15	20	40	934
	xtr	309	3.73	273	31	25	25	13	913
s820	init	491	3.43	418	55	50	47	27	1043
	sa	1832	3.89	317	23	17	31	29	1223
	xtr	1516	3.94	318	18	10	14	12	1268
s1196	init	238	2.54	441	466	409	225	137	714
	sa	1019	4.32	73	79	162	137	177	1764
	xtr	906	4.53	73	58	97	71	88	2005
s1423	init	1024	4.05	477	34	31	39	9	2256
	sa	5394	4.23	407	10	22	21	9	2377
	xtr	3122	4.21	425	11	10	14	8	2378
b03	init	130	2.38	352	10	42	24	42	298
	sa	470	2.73	347	0	2	1	4	414
	xtr	380	2.70	348	0	3	4	8	405
b04	init	168	2.66	630	191	279	230	126	828
	sa	787	3.69	488	33	65	50	122	1526
	xtr	917	3.70	486	41	57	63	79	1558
b09	init	269	2.92	246	36	5	5	10	376
	sa	1357	3.26	230	3	3	2	3	437
	xtr	1338	3.27	233	0	1	1	0	443
b10	init	190	3.21	256	37	35	8	4	530
	sa	463	3.64	217	7	17	6	7	616
	xtr	550	3.77	202	6	6	7	7	642
b11	init	675	3.71	445	8	24	5	13	1335
	sa	1556	3.83	425	4	0	4	1	1396
	xtr	3641	3.86	412	1	3	9	0	1405

From Table 6 it can be seen that the proposed test generation procedure typically results in higher numbers of detections of unspecified transition faults. The test length is sometimes lower under the proposed procedure than if n-detection test generation is carried out for stuck-at faults.

5. CONCLUDING REMARKS

We defined a transition fault model for use with at-speed test sequences. The model was referred to as the unspecified transition fault model since it introduces unspecified values into the faulty circuit when fault effects may occur. Fault detection potentially occurs when an unspecified value reaches a primary output. Due to the uncertainty that the unspecified value will be different from the fault free value, a requirement of this model is that a fault would be detected multiple times. Experimental results demonstrated that the model behaves as expected in terms of fault coverage and numbers of detections of target faults. Moreover, an unspecified transition fault may have a significantly smaller number of detections than the related stuck-at fault. Thus, the model provides a target for the generation of at-speed test sequences, which is more effective than n-detection test generation for stuck-at faults. A variation of an n-detection test generation procedure for stuck-at faults was used for generating test sequences under this model.

ACKNOWLEDGMENTS

The work was supported in part by SRC Grant No. 2004-TJ-1244 (IP) and SRC Grant No. 2004-TJ-1243 (SMR).

REFERENCES

[1] S. Dasgupta, R. G. Walther, T. W. Williams and E. B. Eichelberger, "An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability and Serviceability", in Proc. 11th Fault-Tolerant Computing Symp., 1981, pp. 880-885.

[2] J. Savir and S. Patil, "Scan-Based Transition Test", IEEE Trans. on Computer-Aided Design, Aug. 1993, pp. 1232-1241.

[3] J. Savir and S. Patil, "Broad-Side Delay Test", IEEE Trans. on Computer-Aided Design, Aug. 1994, pp. 1057-1064.

[4] S. Devadas, "Delay Test Generation for Synchronous Sequential Circuits", in Proc. Intl. Test Conf., Aug. 1989, pp. 144-152.

[5] T. J. Chakraborty, V. D. Agrawal and M. L. Bushnell, "Delay Fault Models and Test Generation for Random Logic Sequential Circuits", in Proc. Design Autom. Conf., June 1992, pp. 165-172.

[6] I. Pomeranz and S. M. Reddy, "SPADES-ACE: A Simulator for Path Delay Faults in Sequential Circuits with Extensions to Arbitrary Clocking Schemes", IEEE Trans. on Computer-Aided Design, Feb. 1994, pp. 251-263.

[7] I. Pomeranz and S. M. Reddy, "At-Speed Delay Testing of Synchronous Sequential Circuits", in Proc. 29th Design Autom. Conf., June 1992, pp. 177-181.

[8] S. Bose, P. Agrawal and V. D. Agrawal, "Path Delay Fault Simulation of Sequential Circuits", IEEE Trans. on VLSI Systems, Dec. 1993, pp. 453-461.

[9] K.-T. Cheng, "Transition Fault Testing for Sequential Circuits", IEEE Trans. on Computer-Aided Design, Dec. 1993, pp. 1971-1983.

[10] D. Brand and V. S. Iyengar, "Identification of Redundant Delay Faults", IEEE Trans. on Computer-Aided Design, May 1994, pp. 553-565.

[11] W.-C. Lai, A. Krstic and K.-T. Cheng, "On Testing the Path Delay Faults of a Microprocessor Using Its Instruction Set", in Proc. VLSI Test Symp., Apr. 2000, pp. 15-20.

[12] J. Rearick and R. Rodgers, "Calibrating Clock Stretch During AC Scan Testing", Intl. Test Conf., Nov. 2005, pp. 266-273.

[13] P. Franco and E. J. McCluskey "Three-Pattern Tests for Delay Faults", in Proc. 12th VLSI Test Symp., April 1994, pp. 452-456.

[14] A. Pierzynska and S. Pilarski, "Quality Considerations in Delay Fault Testing", in Proc. EURO-DAC 1995, Sept. 1995.

[15] A. Pierzynska and S. Pilarski, "Non-Robust versus Robust", in Proc. 1995 Intl. Test Conf., Oct. 1995, pp. 123-131.

[16] L.-C. Wang, J.-J. Liou and K.-T. Cheng, "Critical Path Selection for Delay Fault Testing Based Upon a Statistical Timing Model", IEEE Trans. on Computer-Aided Design, Nov. 2004, pp. 1550-1565.

[17] B. Seshadri, I. Pomeranz, S. M. Reddy and S. Kundu, "Path-Oriented Transition Fault Test Generation Considering Operating Conditions", in Proc. Europ. Test Symp., May 2005, pp. 54-59.

[18] J. Waicukauski, E. Lindbloom, B. Rosen and V. Iyengar, "Transition Fault Simulation", IEEE Design & Test, April 1987, pp. 32-38.

[19] I. Pomeranz and S. M. Reddy, "On n-Detection Test Sequences for Synchronous Sequential Circuits", in Proc. 15th VLSI Test Symp., April 1997, pp. 336-342.