

Received October 30, 2020, accepted November 8, 2020, date of publication November 11, 2020, date of current version November 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037292

An Efficient Degraded Deductive Fault Simulator for Small-Delay Defects

TIEQIAO LIU¹, TING YU¹, (Graduate Student Member, IEEE), SHUO WANG¹, AND SHUO CAI²

¹School of Information, Zhejiang University of Finance and Economics Dongfang College, Haining 314408, China

²School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

Corresponding authors: Ting Yu (yuting@zufedfc.edu.cn) and Tieqiao Liu (tieqiao120@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61804037 and Grant 62002314, in part by the Research Foundation of the Zhejiang University of Finance and Economics Dongfang College under Grant 2019dfy003, and in part by the Scientific Research Foundation of Hunan Provincial Education Department under Grant 18A137.

ABSTRACT An efficient degraded deductive simulator for small delay defects is proposed. The proposed method takes into account the conditions of re-convergence sensitization and hazard-based detection, providing fast and accurate simulation results for small delay defects. Separate simulation strategies for faults with different fault effects are proposed. For faults on fault effects re-convergent fan-out stems, the serial simulation technique is applied. For other faults, a deductive simulation technique is proposed to accelerate the simulation. Different from previous works, serial simulations are carried out no longer for all faults on fan-out re-convergent stems, but only for fault effects re-convergences, and the other faults are parallel simulated with the degraded deductive technique, which eliminates “AND” operation and the propagation of fault-list is simpler than conventional deductive ones. Experimental results demonstrate that the proposed simulator that can further accelerate the fault simulation in efficiency. It achieves a 28.3X speedup on average compared with the serial simulation method, and a 3.92X speedup on average compared with the critical path tracing based method.

INDEX TERMS Circuit faults, fault simulation, small-delay defects, deductive simulation, fault effects re-convergence.

I. INTRODUCTION

Integrated circuit (IC) testing is a necessary part of chip production. Empirical data proves that as the chip size shrinking, more and more failures are skewed towards smaller delays [1]. Furthermore, the presence of small delays will cause the chips easier to aging and can be taken as an indication of early life failures [2], [3]. Therefore, small delay defect (SDD) screening becomes increasingly important in IC test [4]–[6] or diagnosis [7], [8].

As an essential part of the test, fault simulation is the key to evaluate test quality, assist test generation and improve test efficiency. There are several techniques for fault simulation acceleration. Lee and Ha [9] proposed a word parallel technique called as HOPE for stuck-at fault simulation. Armstrong [10] proposed a deductive simulation technique, where the fault detection is deduced by the circuit structure when the fault-free simulation is carried out. In the concurrent

fault simulation [11], the event-driven method is integrated with fault simulation in the most effective way, and it can deal with various fault models. Different from other fault models, the SDD refers to a defect whose delay amount is smaller than the system clock cycle. SDD simulation needs to consider the timing information, which leads to a complex simulation.

Many approaches have been proposed for processing timing information. Path-delay based simulations [12]–[16] record the lengths of sensitized paths for timing expression, where the fault detection interval is determined by the maximum sensitized path length. To effectively assist test generation, every sensitized path has to be recorded [16], which results in a high amount of memory consumption and complex processing. Furthermore, sensitized paths are usually determined using robust and non-robust detection conditions, without taking re-convergent sensitization [13] or hazard-based sensitization [17] into account, which renders these approaches inaccurate. Full-waveform based fault simulations [18]–[21] accurately simulate the timing information. Liu *et al.* [18] proposed an improved Boolean

The associate editor coordinating the review of this manuscript and approving it for publication was Dazhong Ma.

process theory for small delay fault simulation. In [18], the timing information is contained in the Boolean expression, and the waveform is simulated by the Boolean process. Czutro *et al.* [19] proposed a new method for waveform expression, where a waveform is formally defined as a set of tuples, and a series of rules are used to handle waveform operations. Since waveforms are represented by Boolean expressions [18] or tuples [19], the operation of waveform becomes very complex with the number of signal transitions increasing. In [20], bitmap data is proposed for waveform expression. In the bitmap data, each bit represents the corresponding time and the logical value of the waveform, thus reducing the storage cost of the waveform recording and the complexity of waveform operation. However, the above methods are based on serial fault simulation. In [21], a critical path tracing technique is applied to small-delay fault simulation, which makes a fast judgment and significantly accelerates the process of simulation. However, the serial simulation needs to be performed on each re-convergence stem, which will result in the consumption of long simulation time. By exploiting multidimensional parallelism from gates, faults, patterns and circuit instances, Schneider *et al.* [22], [23] utilized the computing power of Graphics Processing Units (GPUs) for simulation acceleration.

In this paper, a new acceleration technique for SDD simulation is proposed. For the first time, the idea of deductive simulation is applied to SDD simulation, with the consideration of re-convergent sensitization and hazard-based detection. The contributions of the proposed method are as follows.

- 1) Hazard-based detection is adopted. Compared with the traditional simulator, the SDD simulator with the hazard-based detection consideration improves the detection quality and the result is more accurate.
- 2) A degraded deductive method for SDD simulation is proposed. Compared with traditional deductive simulation methods, the “AND” operation is eliminated, which makes it possible to apply traditional deductive techniques to the SDD simulation. Furthermore, the process of fault-list propagation is simpler than traditional ones.
- 3) The serial simulation application is further reduced. Serial simulation is performed only if there exists fault effects re-convergence. Faults are differentiated by their fault effects. For faults on fault effects re-convergent fan-out stems, the serial simulation technique is applied. For other faults, deductive simulation technique is applied to accelerate the simulation.

The relevant techniques are described in Section 2. Section 3 shows the detailed fault simulation process. Experimental results are shown in Section 4. Section 5 concludes this paper.

II. RELEVANT CONCEPTS

The relevant concepts of small delay defect simulation and hazard detection are described in this section.

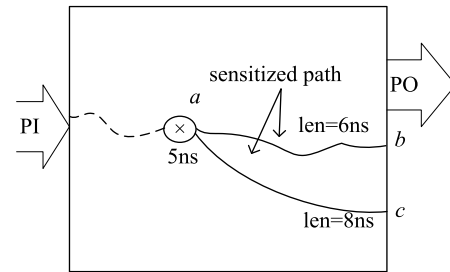


FIGURE 1. Example of SDD detection.

A. SMALL DELAY DEFECT SIMULATION

The small delay defect (SDD) refers to a defect in a circuit whose delay fault size is less than one system clock cycle. The SDD simulation aims to determine the detection interval in the timing domain.

As shown in Figure 1, assuming that the system clock cycle T_c is 20ns, and a delay defect on point a is activated at time 5ns. The fault effects can propagate to primary outputs through two sensitized paths (a, b) and (a, c), and the propagation lengths are 6ns and 8ns, respectively. The difference in the size of delay fault will lead to different results. Both output b and c will be affected when the fault size is greater than 9ns. When the fault size is greater than 7ns and less than 9ns, only the output c will be affected. Neither b nor c will be affected when the fault size is less than 7ns, and the defect on site a cannot be detected.

B. HAZARD-BASED DETECTION

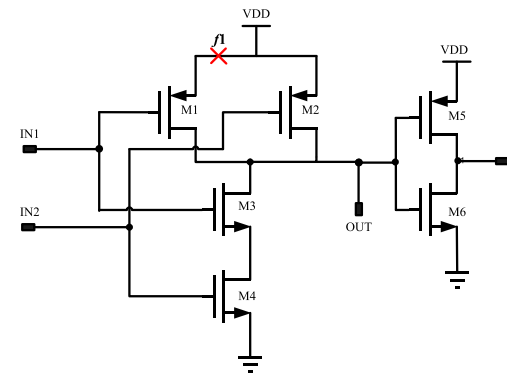
Hazard-based detection is studied based on the fact that hazards can activate delay faults [17]. It is assumed that a hazard/pulse $a' \rightarrow a \rightarrow a'$ ($a \in \{0, 1\}$) in test will activate the delay fault and delay the $a \rightarrow a'$ transition of the pulse.

Figure 2 shows an HSPICE simulation experiment for hazard activation. The experimental circuit consists of a 2-input NAND gate and an inverter as the output load is shown in Figure 2(a). Figure 2(b) shows the simulation results for different resistances (2M Ω and 10M Ω , corresponding to OUT₁ and OUT₂, respectively) inserted on point $f1$. The 0.18 μ m standard SPICE model based on TSMC is used and the supply voltage is set to 1.8V. As we can see, the resistive open defect on $f1$ leads to a slow rise of the hazard, and the amount of delay is increased with the size of the resistance (shown in OUT₁ and OUT₂).

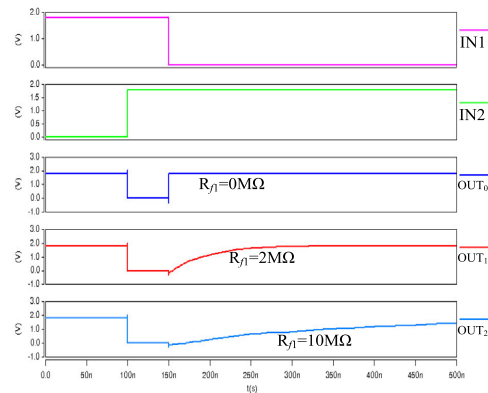
III. FAULTY CIRCUIT TIMING SIMULATION

The paper is the first time to apply the deductive simulation technique to the SDD simulation. The process of fault simulation is shown as follows.

Phase 1 (Circuit pre-Treatment): The pre-treatment includes the identification of re-convergent fan-out stems, fault-free timing simulation and fault activation marking. The unit delay model and the method of bitmap data [10] are adopted for waveform expression. The single fault model is employed. Faults are defined as slow-to-fall (STF) or



(a) Experimental circuit



(b) Resistance inserted on f1

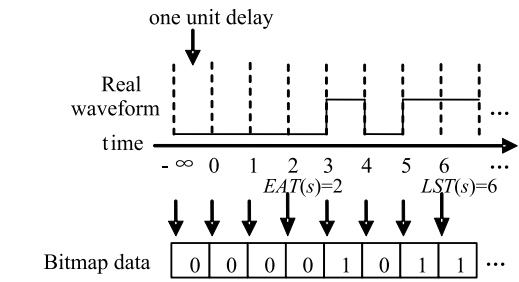
FIGURE 2. HSPICE simulation for resistive open defects.

slow-to-rise (STR) failures on signal lines. The method of key transition [7] is used to determine the activation of the fault. A key transition is defined as the last transition in the waveform. Hazards activation condition [6] is also considered. For each activated fault, the fault activation time and the length of the propagation path are recorded. It is also necessary to record the fault waveform if the fault is fault effects re-convergence.

Phase 2 (Fault Simulation): The simulation of fault circuit is divided into two categories: degraded deductive simulation and serial simulation. The degraded deductive simulation is used to speed up the process of simulation, where all the fault detections are deduced by the fault-free waveform. When there exist fault effects re-convergence, the corresponding faults are deleted from the deductive fault list and sent to serial simulation. The process of serial simulation is from the fan-out re-convergent stem to fan-out-free nodes, dominators [9], or POs. After the serial simulation, the sensitive faults are rejoined to deductive simulation.

A. FAULT-FREE TIMING SIMULATION

A delay test pair consists of two patterns {V1, V2}. V1 is for initialization and V2 is for test. In this method, bitmap data is used for waveform expression. The unit delay model is employed [18], where the fault size is quantized as the times of a unit delay, for instance, the propagation delay of an inverter.



(a) Waveform expression

$$\begin{aligned}
 &EAT(a)=22 \\
 &Bitmap(a)=10B \\
 &EAT(b)=23 \\
 &Bitmap(b)=01B \\
 &EAT(c)=22+2=24 \\
 &Bitmap(c)=Bitmap(a) \text{ OR } \\
 &\quad (Bitmap(b) \gg (23-22)) \\
 &=1000000B \\
 &\quad ||00111111B \\
 &=10111111B
 \end{aligned}$$

(b) Fault-free timing simulation

FIGURE 3. Waveform expression and fault-free timing simulation.

Figure 3(a) shows an example of an actual waveform relative to a bitmap data element. In a bitmap data element, each bit represents the corresponding time and logical value. For example, the first bit of the bitmap data element represents the logic value of the real waveform is 0 at the initialization time (the final stable state of V1), and the second bit represents the logic value of the real waveform under V2 at time 0 is 0. The delay size of each gate can be manually specified or obtained from Standard Delay Format (SDF) files. As only the interval between the earliest arrival time (EAT) and the latest stabilization time (LST) needs to be considered [19], the actual waveform s can be expressed by $EAT(s) = 2$, and $Bitmap(s) = 001011B$, where the initial state (the final stable state of V1) is recorded in the first bit, the following bits record the states of V2 beginning from time $EAT(s) = 2$ to time $LST(s) = 6$.

Figure 3(b) shows another example of a fault-free simulation of an OR gate. The bitmap data 10B on the signal line a indicates that the initial value of the input line a is 1, and it jumps to 0 at the time $EAT(a) = 22$ under V2. In addition, since the recorded waveform value between $EAT(a)$ and $LST(a)$ is only one bit, it can be inferred that $EAT(a)$ of the input line a is equal to $LST(a)$. The output waveform of the OR gate is calculated as follows: $EAT(c)$ is equal to the minimum $EAT(i)$ value in the input of the OR gate plus the propagation delay of the OR gate (assumed to be 2). Since the difference between $EAT(b)$ and $EAT(a)$ is 1, the $Bitmap(c)$ is equal to the $Bitmap(a)$ OR $Bitmap(b)$ after 1-bit right shift. In order to facilitate the shift and calculation, the blank bits on the right are filled with the final stable value, that is, the state value of time $LST(s)$, assuming the word-length is 8 bits.

In the experiment, the waveforms are usually stored in integer data. As the limitation of word-length, the bounds of the

```

1 int source_wave_H, temp;
2 unsigned int source_wave_L;
3
4 temp = source_wave_H<<(32-AND_delay);
5 union_wave_H = source_wave_H>>AND_delay;
6 union_wave_L = (temp|source_wave_L)>>AND_delay;

```

FIGURE 4. Example of 2-word waveform simulation.

representable system clock cycle are limited. Concatenating multiple integers together can alleviate the problem. A signed integer and multiple unsigned integers are used to achieve the words binding, where the signed bit represents the initial state. Figure 4 shows an example of the realization of the shift of 2-word waveform simulation on a 32-bit system.

As can be seen from the above, the use of bitmap data for waveform representation has many advantages. First, the length of the waveform expression is not affected by the number of signal transitions and the storage overhead is lower compared to other waveform representations. Secondly, since the method only records the part of the waveform, the feasibility is higher compared to the previous bitmap waveform representation method [20]. Furthermore, there are corresponding AND-OR-NOT operations in computer instructions, which makes the high-level circuit waveform simulation run at the lower instruction level, resulting in a fast simulation.

B. SERIAL FAULT SIMULATION

Different from serial simulation on all re-convergent fan-out stem faults in [21], a fault proceeds to serial fault simulation if and only if it exists in more than one input fault-list of a gate. Rules for serial fault simulation are as follows.

Rule 1: For the signal line s which is affected by fault f , two bitmap data $GL(s)$ and $FL(s)$ are used for information recording, where $GL(s)$ represents the good waveform, and $FL(s)$ is for fault waveform marking and fault effect recording. For example, $GL(s)$ valued 0011B has the following means: as shown in Figure 3, the first bit means the initial state (the final stable state of $V1$, time $-\infty$) is 0. The second bit means the state of time $0 + EAT(s)$ under $V2$ is 0. A rising transition occurs at time 1. While $FL(s)$ valued 0011B means that a fault with the fault size of x occurs at time $1 + EAT(s)$, and the state will jump from 0 to 1 at time $1 + EAT(s) + x$. If $FL(s)$ valued all 0 or all 1, it means that no fault effect exists.

Rule 2: For the signal line s which is fault unaffected, the waveform interval $[LST(s), +\infty]$ only needs to be recorded, where $LST(s)$ represents the latest stabilization time. Since all transitions occurring before time $LST(s)$ will not affect the output value of the circuit at time T_c (the system clock cycle) [19]. A bitmap data, $GL'(s)$, is used to record the modified $GL(s)$.

Figure 5 shows an example of the serial fault simulation. Assuming $GL(a) = 10001111B$ and a slow-to-rise fault f is activated by the last key transition (hazard activation is considered). Since signal line a is affected by fault f , according to Rule 1, $FL(a)$ is used to record its fault effect. Since only the

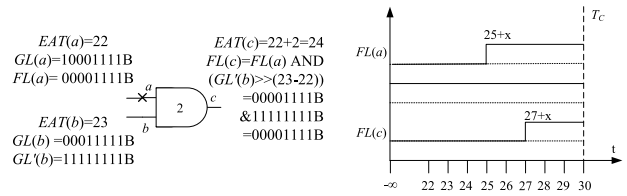


FIGURE 5. Example of fault injection and simulation.

Gate Type	Input a	Input b	Output c	Output Fault List
AND	0	0	0	F_c
	0	1	0	$L_a \cup F_c$
	1	0	0	$L_b \cup F_c$
	1	1	1	$[L_a \cup L_b] \cup F_c$
OR	0	0	0	$[L_a \cup L_b] \cup F_c$
	0	1	1	$L_b \cup F_c$
	1	0	1	$L_a \cup F_c$
	1	1	1	F_c
NOT	0	-	1	$L_a \cup F_c$
	1	-	0	$L_a \cup F_c$

FIGURE 6. Degraded deductive simulation fault propagation rules.

last key transition needs to be focused on, $FL(a)$ is assigned to 00001111B, meaning that the slow rising transition will occur at time $EAT(a) + 3 + x$. For signal line b unaffected by f , Rule 2 is applied. As only the final stable value at time $LST(b)$ needs to be considered, the original waveform $GL(b)$ is modified to $GL'(b) = 11111111B$. The calculations of the values of the output line c are carried out as shown in Figure 5. The serial simulation continues forward until reaching the PO / PPO, fan-out free nodes or its dominator. The propagation effect of fault f can be obtained by combining $EAT(s)$ and $FL(s)$. Assuming that c is PO and $T_c = 30$, the detection interval of fault f is [27], [30], which means that if the size of the fault is greater than 3 units, it will be detected.

C. DEGRADED DEDUCTIVE SIMULATION

The proposed degraded deductive simulation method is based on the elimination of re-convergent fan-out fault effects. A fault proceeds to serial fault simulation when it exists in more than one input fault-list of the gate. After the elimination of the re-convergent fault effect, there is no ‘‘AND’’ operation. The degraded deductive simulation fault propagation rules (under V_2) are shown in Figure 6. L_s represents the fault list on the signal line s , F_s represents the activated delay defect (STF or STR) on the signal line s . The propagation rules of multiple-input gates and other types of gates can be extended by these rules. Figure 7 shows the propagation rules for the traditional stuck-at fault deductive simulation, where L_s represents the fault list on the signal line s and c_i represents the activated s-a-0 ($i = 0$) and s-a-1 ($i = 1$) faults on the signal line c . Compared with the traditional deductive simulation method, it can be seen that the proposed degraded deductive simulation method eliminated ‘‘AND’’ operation is much simpler to deal with the fault list.

Gate Type	Input a	Input b	Output c	Output Fault List
AND	0	0	0	$[L_a \cap L_b] \cup c_1$
	0	1	0	$[L_a \cap L_b] \cup c_1$
	1	0	0	$[L_a \cap L_b] \cup c_1$
	1	1	1	$[L_a \cup L_b] \cup c_0$
OR	0	0	0	$[L_a \cup L_b] \cup c_1$
	0	1	1	$[L_a \cap L_b] \cup c_0$
	1	0	1	$[L_a \cap L_b] \cup c_0$
	1	1	1	$[L_a \cap L_b] \cup c_0$
NOT	0	-	1	$L_a \cup c_0$
	1	-	0	$L_a \cup c_1$

FIGURE 7. Traditional stuck-at deductive simulation fault propagation rules.

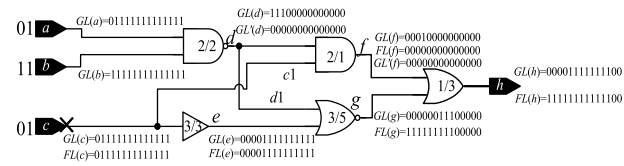


FIGURE 8. Example of comparison for fault simulation.

IV. EXAMPLE AND COMPARISONS

To further elucidate the superiority of the proposed method, two example comparisons with previous methods are given in this section.

A. SIMPLE AND FAST CALCULATION IN DEGRADED DEDUCTIVE SIMULATION

Figure 8 shows an example derived from [19], [21] for fault simulation. Assuming that the test pair is $\langle 010, 111 \rangle$. The value on each gate is shown as a rising delay/falling delay of each gate. In the previous works [19], [20], each fault is handled at a time. Especially for the slow-to-rise fault STR_c , as it is on the re-convergent fan-out stem, serial fault simulation needs to be applied. GL' and FL calculated using rules 1 and 2 are shown on each line.

Now, the method of degraded deductive simulation is employed. Referring to the rules of Figure 6, the fault list on each line is calculated as follows.

$L_a = \{STR_a\}$, $L_b = \{\Phi\}$. As the inputs of the NAND gate d is (1,1), the fault list of d is calculated as:

$$L_d = \{L_a \cup L_b \cup STF_d\} = \{STR_a, STF_d\}.$$

$L_c = L_{c1} = \{STR_c\}$. As the inputs of the AND gate f is (0,1), the fault list of f is calculated as:

$$L_f = \{L_d \cup STF_f\} = \{L_d \cup STF_f\} = \{STR_a, STF_d, STF_f\}.$$

$$\text{Similarly, } L_e = \{L_c \cup L_e\} = \{STR_c, STR_e\}.$$

$$L_{d1} = \{\Phi\}, L_g = \{L_e \cup STF_g\} = \{STR_c, STR_e, STF_g\}.$$

$$L_h = \{L_f \cup L_g \cup STF_h\} = \{STR_a, STF_d, STF_f, STR_c, STR_e, STF_g, STF_h\}.$$

For simplicity, the length of the propagation path of each fault is not shown. From the above it can be seen that the degraded deductive simulation can handle all the faults at once. As no fault both exists in more than one input fault-list

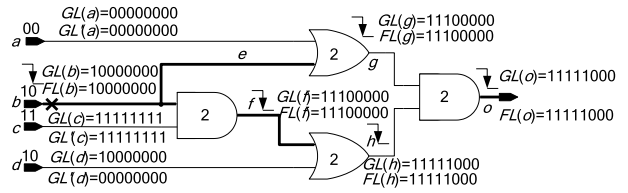


FIGURE 9. Example for re-convergent sensitization.

of the gate, serial fault simulation can be canceled in this example.

B. RE-CONVERGENT SENSITIZATION SIMULATION

A detailed example of fan-out re-convergent sensitization is shown in Figure 9. The propagation delay is shown on each logic gate. Assuming that the test pair is $\langle 0111, 0010 \rangle$.

The circuit first proceeds to fault-free timing simulation. $GL(s)$ represents the fault-free timing simulation result of each signal line s , with all $EAT(s)$ s assumed to be 0, for simplicity. The information of each activated fault can be derived from $GL(s)$. For example, as $GL(b) = 10000000$, it denotes that a slow-to-fall fault will be activated on line b at time 0 under V_2 . Similarly, $GL(h) = 11111000$ denotes that a slow-to-fall fault will be activated on line h at time 4.

Then, the degraded deductive simulation is applied. Referring to rules in Figure 6, the fault list on each line is calculated as follows.

$$L_a = L_c = L_d = \{\Phi\}, L_b = \{STF_b\}.$$

As the inputs of the AND gate f is (0,1), the fault list of f is calculated as:

$$L_f = \{L_b \cup STF_f\} = \{STF_b, STF_f\}$$

Similarly, the fault list of g is calculated as:

$$L_g = \{L_a \cup L_b \cup STF_g\} = \{STF_b, STF_g\}.$$

Similarly,

$$L_h = \{L_f \cup L_d \cup STF_h\} = \{STF_b, STF_f, STF_h\}.$$

As the fault STF_b exists both in the input fault lists of gate o (L_g and L_h), STF_b should be deleted from the degraded deductive simulation fault list, then, L_g and L_h turn into:

$$L_g = \{STF_g\}, \text{ and } L_h = \{STF_f, STF_h\}.$$

As the inputs of the AND gate o is (0,0), the fault list of i is calculated as:

$L_o = \{STF_o\}$, which means only STF_o can be detected in this degraded deductive simulation.

Meanwhile, STF_b is carried out to serial fault simulation from its fan-out re-convergent stem. $GL'(s)$ and $FL(s)$ calculated according to rules 1 and 2 are shown on each signal line. The calculation process is as follows.

As signal line b is affected by STF_b , according to Rule 1, $FL(b)$ is recorded. $FL(b) = 10000000$ means a delay fault occurs at time 0.

As signal line c is not affected by STF_b , according to Rule 2, only the waveform interval $[LST(s), +\infty)$ needs to be recorded, and $GL(c)$ is modified to $GL'(c)$. $GL'(c) = 11111111$, where 1 is the state value of time $LST(c)$.

$FL(f)$ is calculated as the result of $FL(b)$ OR $GL'(c)$ after 2 bit right shift.

TABLE 1. Small-delay fault simulation results of 1000 random LOC test pairs.

Circuit	Re-con.		Proposed				[20]		[21]	
	Dom	NoD	FC	Mem(M)	D%	CPU(s)	CPU(s)	X	CPU(s)	X
S5378	400	277	40.1	4.78	2.11	4.75	20.3	4.27	7.52	1.58
S9234	443	294	24.3	6.17	7.23	3.74	24.6	6.58	5.59	1.49
S13207	338	338	14.2	7.87	0.03	6.56	77.6	11.83	13.98	2.13
S15850	471	336	13.0	8.84	18.71	11.75	101.7	8.66	17.45	1.49
S35932	3222	47	43.5	13.91	20.31	70.92	2552.6	35.99	330.01	4.65
S38417	1927	864	28.1	16.93	20.27	37.65	1919.9	50.99	196.66	5.22
S38584	987	885	18.6	15.52	3.37	28.63	2288.6	79.94	180.16	6.29
B20	969	1948	22.7	16.65	7.11	65.46	N/A	N/A	385.26	5.89
B22	1286	3064	18.9	22.13	6.99	123.22	N/A	N/A	805.58	6.54
Average					9.57			28.32		3.92

The calculations of other signal lines are carried out in the same way. $FL(g)$ is calculated as the result of $FL(e)$ OR $GL'(a)$ after 2-bit right shift. $FL(h)$ is calculated as the result of $FL(f)$ OR $GL'(d)$ after 2 bit right shift. $FL(o)$ is calculated as the result of $FL(g)$ AND $FL(h)$ after 2 bit right shift.

Fault detection interval can be derived from L_o or $FL(o)$, which can be determined by the fault activation time and the length of the sensitized path. For example, from $L_o = \{STF_o\}$, the detection interval of fault STF_o can be determined as $[Tc-4, Tc]$, since the activation time of STF_o is 4, derived from $GL(o) = 1111000$, and the length of sensitized path is 0. Likewise, from $FL(o) = 11111000$, it can get the detection interval of fault STF_b is $[Tc-4, Tc]$, which means if the size of fault STF_b is greater than 4 units, it will be detected. However, in the traditional path based simulation methods, the fault STF_b is undetectable for this test pair. Because there is no strong path or non-robust path between lines b and o [13]. In fact, the signal transition at the output o is the result of the combined operation of the sensitized path $b \rightarrow f \rightarrow h \rightarrow o$ and the sensitized path $b \rightarrow e \rightarrow g \rightarrow o$, where the time of signal transition is determined by the short sensitized path $b \rightarrow e \rightarrow g \rightarrow o$.

C. THEORETICAL ANALYSIS

By extracting the “AND” operations for serial simulation, the deductive fault simulation for SDD is realized for the first time. The results mentioned above are consistent with [20] and [21]. Investigating its principle, $GL(s)$ realizes accurate and fault-free timing simulation. $FL(s)$ takes into account the re-convergent sensitization and hazard-based detection, and can accurately describe the fault effect propagation. The applications of $GL'(s)$ and degraded deductive simulation are actually equivalent to non-robust conditions of conventional path simulation. As the serial simulation is only carried out for re-convergent fault effects, and the degraded deductive technique can handle faults in a single simulation, the proposed method is simple, accurate and efficient.

V. EXPERIMENTAL RESULTS

The proposed small delay fault simulator is implemented in C programming language on a 3.0GHz Pentium Pro PC with 2.0GB RAM. Experiments are carried out on ISCAS'89

and ITC'99 benchmark circuits. The calculations of detection intervals and fault coverage in [20], [24] are used.

The system clock cycle is assigned to the longest path delay plus a safety margin of 20%. Assuming that the propagation delay of the NOT gate is 1 unit, and the propagation delay of the AND / NAND / OR / NOR gate is 2 units, and the XOR / XNOR gate delay is 4 units. Taking gate inertia effect into account, a pulse at gate inputs will be filtered if the pulse width is less than the gate's inertia delay [19].

Table 1 shows the experimental results of the proposed method with other compared methods on large ISCAS'89 and IWLS'05 benchmark circuits. Column 1 shows the circuit name. Column 2 and column 3 show the number of re-convergent fan-out stems of each circuit, representing the number of re-convergent fan-out stems with a dominator and without a dominator, respectively. Column 4 to column 7 shows the results of the proposed method with 1000 random test pairs under LOC (launch-off-capture) test simulation. Column 4 and column 5 show the fault coverage and memory overhead, respectively. Column 6 shows the percentage of the fault effect re-convergence, which means the ratio of the fault effect of re-convergent fan-out stem propagating to re-convergent nodes. Column 7 shows the computation time (in seconds) with no fault dropping. Column 8 and column 10 show the results of computation time for [20] and [21], respectively. Column 9 and column 11 show the acceleration ratios of the proposed method relative to [20] and [21], respectively. They all use the same unit delay model and the same hardware platform.

As can be seen from Table 1, the proposed method shows higher efficiency in fault simulation. Compared with the serial simulation method [20], as the modified waveform expression and the fast determination technology are adopted, the average speedup is 28.32. Compared with the critical path tracing based method [21], the simulation speed is further improved because of the method of serial simulation only for re-convergent fault effects and the deductive simulation method are adopted. Actually, as column 6 shown, the ratio of the activated fault effect of the re-convergent fan-out stem propagating to the re-convergent nodes is very low. The speed up is more obvious with the decrease of the fault effect re-convergence ratio and the increase of the circuit size, such as circuit S38584 and B22.

TABLE 2. Statistics of discontinuous detection intervals.

Circuit	Fault_Way_Num	Fault_Num	Ratio(%)
S344	3926	0	0
S832	51	1	0.001000
S1423	7154	43	0.017000
S1488	203	2	0.001000
S5378	4153	25	0.004000
S9234	33147	235	0.021000
S13207	38316	399	0.026000
S15850	106344	135	0.007000
S35932	69371	0	0.000000
S38417	352594	237	0.004765
S38584	267711	222	0.003624
B07	5116	0	0
B11	5242	0	0
B12	13875	33	0.006974
B13	881	0	0
B14	152851	115	0.002862
B20	332113	548	0.006799
B22	515717	1153	0.009624
Average	103289	178	0.006203

The proposed method can well assist the test generation. Table 2 shows a statistical result for fault waveforms with 1000 random test pairs under LOC test simulation. Column 1 shows the circuit name. Column 2 shows the total number of fault waveforms that exist discontinuous detection intervals on each output under each test pair. Column 3 shows the result of the number of faults that still exist discontinuous detection intervals after the union of the detection intervals when the test is finished. The ratio of the number of faults with discontinuous detection intervals to the number of the total faults, expressed as a percentage, is shown in column 4. The averages are shown in the last row.

As it can be seen from Table 2, as the effects of fault propagation paths convergences, the fault waveform on output may have multiple detection intervals. Although the ratio of the number of faults with discontinuous detection intervals to the number of total faults is very low (with an average of 0.006203%) after detection intervals unions, the fault waveform is easy to exist discontinuous detection intervals under random test pairs, as column 2 shown.

When it is considered the aids of fault simulation to test generation, the test pair that causes multiple detection intervals is not recommended. As multiple detection intervals would mean fault masking [25], i.e., small delay defects can be detected, but large defects may escape the test. To ensure the quality of the test, the test generation program is required to regenerate the test or take the output mask technique [26] to process the output of multiple detection intervals.

Figure 10 shows another experiment of fault detection analysis on sensitized path length with 5000 random LOC test pairs. The sensitized path length of each detected fault for B13 and B14 is shown in Figure 10 in descending order, respectively. When considering applying a faster-than-at-speed test, a suitable solution can be found from the statistical curves, so that a high fault coverage can be obtained with the least number of test clocks. From the statistical curves it can be also learned that compared with B13, the path distribution

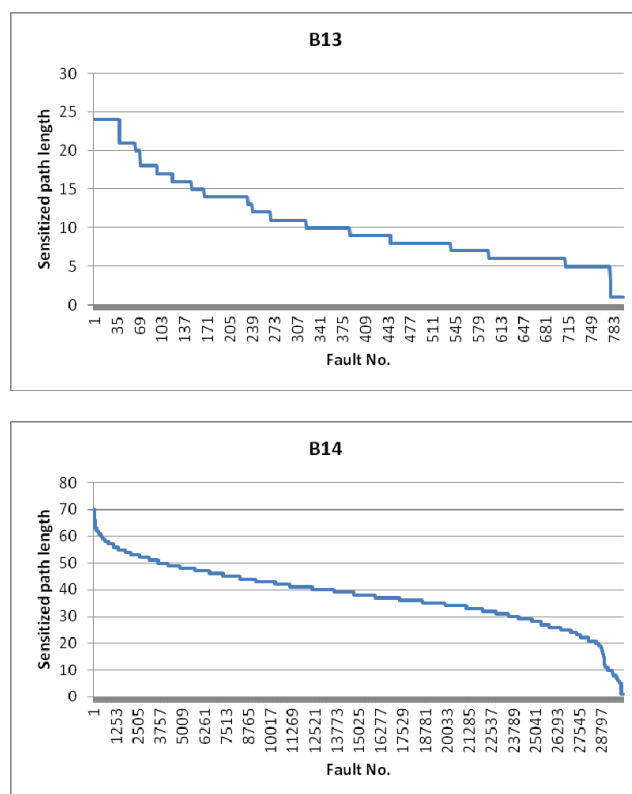


FIGURE 10. Sensitized path length statistics for B13 and B14.

of B14 is more even, which can provide guidance for circuit design.

VI. CONCLUSION

In order to accelerate the speed of SDD simulation, an efficient small delay fault simulator is proposed. The method takes into account re-convergent sensitization and hazard-based detection, aiming to provide a fast and accurate simulation result for SDD screening. Separate simulation strategies for faults of different fault effects are proposed. For faults of fault effects re-convergence, the serial simulation technique is applied. For other faults, the deductive simulation technique is applied to accelerate the simulation.

Compared with previous path-based SDD simulation, the proposed method enables the consideration of re-convergent sensitization and hazard-based simulation, to provide a more accurate simulation result for SDD screening. Different from previous waveform-based SDD simulation works, the proposed method no longer focuses on fault sites but fault effects when it carries out the serial simulation, thus the serial fault simulation time is greatly reduced. By eliminating the “AND” operation, the proposed method has first applied traditional deductive techniques to the SDD simulation. Compared with previous waveform-based SDD simulation works, the proposed method carries out serial simulation only for re-convergent fault effects, and the degraded deductive technique can handle other faults in a single simulation, leading to an efficient acceleration in the fault simulation.

REFERENCES

- [1] P. Nigh and A. Gattiker, "Test method evaluation experiments and data," in *Proc. Int. Test Conf.*, Atlantic City, NJ, USA, Oct. 2000, pp. 454–463.
- [2] Y. M. Kim, Y. Kameda, H. Kim, M. Mizuno, and S. Mitra, "Low-cost gate-oxide early-life failure detection in robust systems," in *Proc. Symp. VLSI Circuits*, Honolulu, HI, USA, Jun. 2010, pp. 125–126.
- [3] S. Cai, W. Wang, F. Yu, and B. He, "Single event transient propagation probabilities analysis for nanometer CMOS circuits," *J. Electron. Test.*, vol. 35, no. 2, pp. 163–172, Apr. 2019.
- [4] I. Polian, B. Becker, S. Hellebrand, H.-J. Wunderlich, and P. Maxwell, "Towards variation-aware test methods," in *Proc. 16th IEEE Eur. Test Symp.*, Trondheim, Norway, May 2011, pp. 219–225.
- [5] M. Sauer, A. Czuto, I. Polian, and B. Becker, "Small-delay-fault ATPG with waveform accuracy," in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, San Jose, CA, USA, 2012, pp. 30–36.
- [6] S. Eggersglüß and R. Drechsler, *High Quality Test Pattern Generation and Boolean Satisfiability*. New York, NY, USA: Springer, 2012, pp. 2–20.
- [7] X. Qian and A. D. Singh, "Distinguishing resistive small delay defects from random parameter variations," in *Proc. 19th IEEE Asian Test Symp.*, Shanghai, China, Dec. 2010, pp. 325–330.
- [8] M. H. Tehranipoor, K. Peng, and K. Chakrabarty, *Test and Diagnosis for Small-Delay Defects*. New York, NY, USA: Springer, 2011, pp. 2–22.
- [9] H. Ki Lee and D. Sam Ha, "HOPE: An efficient parallel fault simulator for synchronous sequential circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 15, no. 9, pp. 1048–1058, Sep. 1996.
- [10] D. B. Armstrong, "A deductive method for simulating faults in logic circuits," *IEEE Trans. Comput.*, vol. C-21, no. 5, pp. 464–471, May 1972.
- [11] E. G. Ulrich and T. Baker, "Concurrent simulation of nearly identical digital networks," *Computer*, vol. 7, no. 4, pp. 39–44, Apr. 1974.
- [12] B. N. Lee, L.-C. Wang, and M. S. Abadir, "Reducing pattern delay variations for screening frequency dependent defects," in *Proc. 23rd IEEE VLSI Test Symp. (VTS)*, Palm Springs, California, May 2005, pp. 153–160.
- [13] X. Lin, K.-H. Tsai, C. Wang, M. Kassab, J. Rajsiki, T. Kobayashi, R. Klingenberg, Y. Sato, S. Hamada, and T. Aikyo, "Timing-aware ATPG for high quality at-speed testing of small delay defects," in *Proc. 15th Asian Test Symp.*, Fukuoka, Japan, Nov. 2006, pp. 139–146.
- [14] S. K. Goel, N. Devta-Prasanna, and R. P. Turakhia, "Effective and efficient test pattern generation for small delay defect," in *Proc. 27th IEEE VLSI Test Symp.*, Santa Cruz, CA, USA, May 2009, pp. 111–116.
- [15] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "Test-pattern grading and pattern selection for small-delay defects," in *Proc. 26th IEEE VLSI Test Symp. (VTS)*, San Diego, CA, USA, Apr. 2008, pp. 233–239.
- [16] K. Peng, M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "Crosstalk- and process variations-aware high-quality tests for small-delay defects," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 6, pp. 1129–1142, Jun. 2013.
- [17] T. Liu, Y. Zhou, Y. Liu, and S. Cai, "Harzard-based ATPG for improving delay test quality," *J. Electron. Test.*, vol. 31, no. 1, pp. 27–34, Feb. 2015.
- [18] L. Liu, J. Kuang, and H. Li, "Small delay fault simulation for sequential circuits," in *Proc. 15th IEEE Pacific Rim Int. Symp. Dependable Comput.*, Shanghai, China, Nov. 2009, pp. 63–68.
- [19] A. Czuto, N. Houarche, P. Engelke, I. Polian, M. Comte, M. Renovell, and B. Becker, "A simulator of small-delay faults caused by resistive-open defects," in *Proc. 13th Eur. Test Symp.*, Verbania, Italy, May 2008, pp. 113–118.
- [20] L. Zhou, J. Kuang, and S. Cai, "An efficient fault simulation method for small-delay faults," *J. Chin. Comput. Syst.*, vol. 33, no. 6, pp. 1363–1366, 2012.
- [21] T. Liu, J. Kuang, S. Cai, and Z. You, "An efficient small-delay faults simulator based on critical path tracing," *Int. J. Circuit Theory Appl.*, vol. 43, no. 8, pp. 1015–1023, 2015.
- [22] E. Schneider, S. Holst, M. A. Kochte, X. Wen, and H.-J. Wunderlich, "GPU-accelerated small delay fault simulation," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Grenoble, France, Mar. 2015, pp. 1174–1179.
- [23] E. Schneider, M. A. Kochte, S. Holst, X. Wen, and H.-J. Wunderlich, "GPU-accelerated simulation of small delay faults," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 5, pp. 829–841, May 2017.
- [24] Y. Sato, S. Hamada, T. Maeda, A. Takatori, and S. Kajihara, "Evaluation of the statistical delay quality model," in *Proc. Conf. Asia South Pacific Design Autom. (ASP-DAC)*, Shanghai, China, Jan. 2005, pp. 305–310.
- [25] B. Kruseman, A. K. Majhi, G. Gronthoud, and S. Eichenberger, "On hazard-free patterns for fine-delay fault testing," in *Proc. Int. Conf. Test*, Charlotte, NC, USA, Oct. 2004, pp. 213–222.
- [26] T. Yoneda, K. Hori, M. Inoue, and H. Fujiwara, "Faster-than-at-speed test for increased test quality and in-field reliability," in *Proc. IEEE Int. Test Conf.*, Anaheim, CA, USA, Sep. 2011, pp. 1–9.

• • •