Test Generation in VLSI Circuits for Crosstalk Noise[1]

Weiyu Chen Sandeep K. Gupta Melvin A. Breuer

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2562

Abstract

This paper addresses the problem of efficiently and accurately generating two-vector tests for crosstalk induced effects, such as pulses, signal speedup and slowdown, in digital combinational circuits. These effects are becoming more prevalent due to short signal switching times and deep submicron circuitry. These noise effects can propagate through a circuit and create a logic error in a latch or at a primary output. We first present a new way for predicting the output waveform produced by an inverter due to a non-square wave pulse at its input. Our modeling technique captures such properties as the amplitude of a pulse and its rise/fall times and the delay through a device. To expedite the computation of the response of a logic gate to an input pulse, we have developed a novel way of modeling such gates by an equivalent inverter. We have developed a mixed-signal test generator that incorporates classical PODEM-like static values as well as dynamic signals such as transitions and pulses, and timing information such as signal arrival times, rise/fall times, and gate delay. We also present a new analog cost function that is used to guide the search process. Comparison of results with SPICE simulations confirms the accuracy of this approach. This paper focuses primarily on crosstalk induced pulses, but these results have been extended to deal with speedup and slowdown effects.

I. Introduction

The dramatic increase in signal switching speed and density of integrated circuits leads to challenging design and test problems. Interconnection lines that were once considered to be electrically isolated can now interfere with each other and have an important impact on system performance and correctness. One such interaction caused by parasitic coupling between wires is known as crosstalk. Crosstalk noise may cause undesirable effects including excessive overshoot, undershoot, glitches, additional signal delay and even a reduction in signal delay [11]. Crosstalk can produce logic errors in the circuit. Current trends in integrated circuit design indicate that signal noise and skew due to crosstalk create severe design and test problems.

These problems are further aggravated by variations in the fabrication process [2]. If the area and performance constraints for a circuit are not too stringent, then an error observed during validation can be eliminated by re-routing signals or redesign [3]. However, in designs with aggressive goals, it may not be possible to eliminate all noise effects at all worst case design and fabrication corners. In addition, because of the random nature of process defects and variations, careful design and validation cannot guarantee all manufactured parts to be free of error causing crosstalk effects. Thus, testing for severe process aggravated noise effects is necessary to ensure the correct functionality of fabricated chips.

Logic level crosstalk fault models and a PODEM based ATPG algorithms were presented in [1, 14]. These models characterize crosstalk effects as static hazards having a full voltage swing, and result in an overestimation of noise strength. In addition, since crosstalk is a finite energy transient effect, test vectors generated using these models may not be able to actually propagate the noise to POs or flip-flops because of the inertia inherent to gates. The ability to *efficiently* and *accurately create* a *large* crosstalk effect and *propagate* it with *minimal attenuation* has not been previously addressed.

In this paper a mixed-signal test generation process is proposed where characteristics of crosstalk induced noise are accurately modeled. Conditions are employed so that a crosstalk effect, E, is first generated, and then other constraints are employed so as to propagate the effect E to an output or a flip-flop such that it has maximum amplitude and width. The algorithm is PODEM-like, but to process noise effects a first order model based on LEVEL 1 MOS device equations is used. Because second order effects such as channel length modulation and body effect are ignored, some tests generated may overestimate the severity of the noise. Since the complexity of these high level models make pure analytic approach impractical, we use the first order equations to approximate the noise strength and generate the test vectors. Due to limitation of space, in this paper all derivations and examples focus on crosstalk pulse case, the conditions and analysis procedures for speedup and slowdown are similar.

The paper is organized as follows. In section II, the theoretical foundations for the proposed methodology

[1] This work was supported in part by the Defense Advanced Research Project Agency and monitored by the Department of the Army, Ft. Huachuca, under Contract No. DABT63-95-C-0042, and by Intel Corporation. The information reported here does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

[2] A logic error is either a binary error in a flip-flop or at an output.

are presented. These results are used to compute analog properties of noise. In section III the proposed methodology to generate test vectors is presented. In section IV we discuss experimental results. Finally, in section V we present our conclusions.

II. Theoretical foundations for proposed methodology

To accurately propagate noise through gates, we need to (1) characterize the noise waveform, (2) analyze gate transfer functions, and (3) compute output noise waveforms. Since many CMOS gates in a random logic circuit have different electrical characteristics, our approach is to first model CMOS logic gates as equivalent inverters and then calculate the output response of noise through this gate using the transfer function of the equivalent inverter. In Section 2.1 a new inverter model is presented that reduces the error found in other approaches caused by neglecting the short circuit current. In Section 2.2 we propose a method to determine the inverter that is equivalent to a given CMOS logic gate (NAND, NOR). This method can also be generalized to complex gates. In section 2.3 we characterize the noise waveform and calculate the propagated output noise waveform through the equivalent inverter.

2.1 A new inverter model

Accurate circuit models are important in CMOS design and validation. Several analytic models have been proposed for the transient response of CMOS inverters [4-6]. Although these models take into account the influence of the input waveform on the propagation delay, the short-circuit current is neglected. For current technology where the signal transition time is near 100ps and the gate load is in the range of 10-50fF, neglecting short-circuit current can result in a 20-50% error in the estimation of the propagation delay and output waveform. Since crosstalk noise is a finite energy transient phenomenon, we proposed an improved model for a CMOS inverter to take into account the short-circuit current so that the error in estimating the propagated noise can be significantly reduced.

The derivations assume a rising input transition. Similar results have been obtained for falling input transitions.

Consider the CMOS inverter in Fig. 1(a). We wish to determine the falling output waveform $V_o(t)$ due to a rising input ramp $V_{in}(t)$ with rise time t_r. Assume all circuit capacitance is lumped into one grounded load capacitance C at the inverter's output and all voltages have been normalized with respect to V_{DD}. The charging of the capacitance C can be expressed by

$$I_p = -I_n - C\frac{dV_o}{dt},$$

where

$$I_n = b_n\left[(V_{in}-v_{tn})V_o - V_o^{\,2}/2\right], \qquad for\ \ (V_{in}-v_{tn}) > V_o,\ and$$

$$= \frac{b_n}{2}(V_{in}-v_{tn})^2, \qquad for\ \ (V_{in}-v_{tn}) < V_o;$$

$$I_p = b_p\left[(V_{in}-1-v_{tp})(V_o-1)-(V_o-1)^2/2\right],\ for\ \ (V_{in}-v_{tp}) < V_o,\ and$$

$$= \frac{b_p}{2}(V_{in}-1-v_{tp})^2, \qquad for\ \ (V_{in}-v_{tp}) > V_o.$$

β_n (β_p) is the gain factor, and v_{tn} (v_{tp}) is the transistor threshold voltage normalized with respect to Vdd of the NMOS(PMOS) transistors.

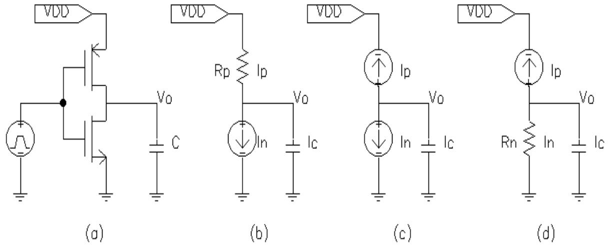


Fig.1 CMOS inverter and its corresponding model when N and P MOS transistors operate in different modes: (a) circuit, (b) PMOS in linear and NMOS in saturation, (c) both in saturation, and (d) NMOS in linear and PMOS in saturation.

When the input is first applied, the NMOS (PMOS) is in saturation (linear) region and can be modeled as shown in Fig. 1(b), where we replace the NMOS by a current source and the PMOS by a resistance. As long as the PMOS is in the linear region, the circuit can be characterized by the differential equation

$$\frac{1-V_o}{R_p} = C\frac{dV_o}{dt} + \frac{b_n V_{DD}}{2}(V_{in}-v_{tn})^2 .$$

With the initial condition $V_o = 1$ when $V_{in} = v_{tn}$, integration yields

$$V_o = P\cdot e^{\frac{-(t-t_r v_{tn})}{R_p C}} + A(V_{in}-v_{tn})^2 + B(V_{in}-v_{tn}) + D, \qquad (1)$$

where

$$K = b_n V_{DD}/2C,\ P = 2R_p^{\,3}C^3K/t_r^{\,2},\ A = -R_pCK,$$

$$B = 2R_p^{\,2}C^2K/t_r,\ \text{and}\ D = 1-2R_p^{\,3}C^3K/t_r^{\,2}.$$

However, the on-channel resistance R_p of the PMOS transistor in this model is not constant during the input transition. R_p is small (P-channel is fully ON) when the input is small and becomes very large when the PMOS transistor saturates to become a current source. Taking this non-constant property into account we modify the channel resistance as a function of input waveform, namely, we set

$$R_p = \frac{1}{\left|b_p V_{DD}\left(V_{in}-1-v_{tp}\right)\right|},$$

where $V_{in}(t) = t/t_r$.

When the input is rising and the output voltage drops to $(V_{in}-v_{tp})$, the PMOS transistor goes into saturation.

The circuit can now be modeled as shown in Fig. 1(c) and can be described by the equation

$$\frac{b_p V_{DD}}{2}\left(V_{in}-1-v_{tp}\right)^2=\frac{b_n V_{DD}}{2}\left(V_{in}-v_{tn}\right)^2+C\frac{dV_o}{dt}.$$

Integrating the above equation we obtain

$$V_o=\frac{b_p V_{DD} t_r}{6C}\left(V_{in}-1-v_{tp}\right)^3-\frac{b_n V_{DD} t_r}{6C}\left(V_{in}-v_{tn}\right)^3+M, \qquad (2)$$

where M is a constant and can be obtained by using the boundary condition ($V_o = V_{in}-v_{tp}$) in both equations (1) and (2).

As the output voltage continues to drop, the NMOS transistor will eventually operate in the linear region. The circuit can now be modeled as shown in Fig. 1(d). The equations characterizing this region are similar to the case in Fig. 1(b).

In [11] it was shown that for a specific case, when an affecting line has a transition with a 100ps rise time, the slope of the rising edge of the crosstalk noise on the affected line is about 250ps. Fig. 2 shows the result of our new model. The input waveform is assumed to be a ramp having a rise time of 250ps, and the load capacitance is 15fF. The results using our model match SPICE results very well except for the tail portion of the response. Note that the result based on ignoring the PMOS transistor has a significant error.

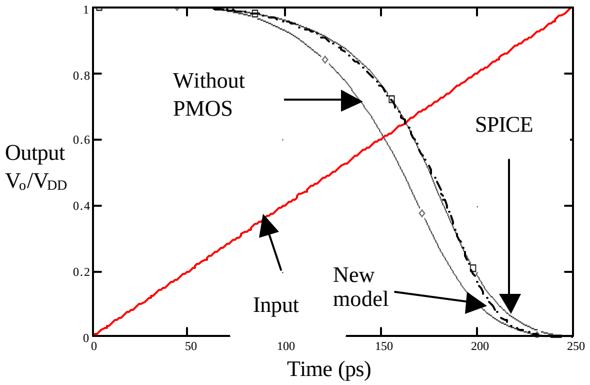


Fig. 2 Comparison of analytic result of proposed model and SPICE simulations.

2.2 A method to collapse CMOS gates

In this section we deal with the problem of propagating a pulse (noise) through a NAND or NOR gate. We set the side fan-in's to their non-controlling values. Our approach for computing the output noise for a general CMOS gate is to collapse the gate to an equivalent inverter and then apply the results in section 2.1. Collapsing techniques have been previously used for computing propagation delay [7-10]. The methods presented in [7] treat series transistors as series resistors and add the widths of parallel devices. This leads to an inaccurate estimate of delay. The approaches described in [8,9] need either pre-characterization or DC analysis to determine some necessary parameters, which is technology dependent and is not applicable in the ATPG process. Although the approach in [10] provides a good estimation of propagation delay, the predicted output waveforms do not match well with SPICE simulations. Since the propagation of the noise depends heavily on the gate's response, we have developed a new but simple approach to collapse CMOS gates into equivalent inverters.

2.2.1 Series MOS

The effective transconductance, β_{eff}, of n series-connected transistors is traditionally approximated as β/n. This approximation is valid only when the input is a step function, all transistors operate in their linear regions, and they all have the same β value. Consider the pull-down NMOS chain of a CMOS NAND gate in Fig. 3(b), where the V_{DS} and/or V_{GS} of each MOSFET in the series-connected chain is smaller than that of the inverter (Fig. 3(a)). Assume that all devices have identical β values. Also assume that there are no more than 5 MOSFETs connected in series. When the input transition is applied, the switching MOS first operates in the saturation mode and then moves into the linear region. In addition, during the first part of the input transition all transistors above the switching MOSFET operate in saturation and all those below the switching MOSFET operate in the linear region. This results in the primary source of error in the use of the β/n approximation. Thus, to take this into account we need to estimate β_{eff} under various conditions of operations.

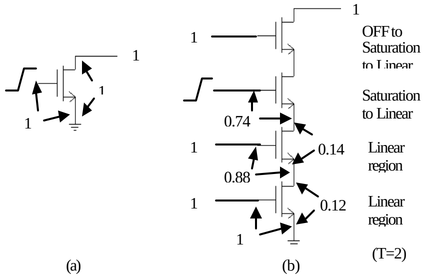


Fig. 3 Pull-Down NMOS chains; (a) single MOS, (b) series connected MOS, all values normalized w.r.t. Vdd.

When the input transition first occurs, both the MOS in Fig. 3(a) and the switching MOS in Fig. 3(b) are in the saturation region and thus V_{GS} determines the device current. For the single MOS in Fig. 3(a), assume $V_{GS} = v_{inv}$ is the input voltage at which $V_{out} = 0.5$ (i.e. $V_{DD}/2$). For the switching MOS in Fig. 3(b) to conduct the same amount of current so that V_{out} can drop to $V_{DD}/2$, the input voltage applied to the switching device must be $v_{inv}+\sum_i V_{DS}^{\,i}$, where i is over all transistors below the switching MOS. At the instant that the switching MOS moves from the saturation region into the linear region, the voltages across the T MOSFETs below the switching device are as indicated in Fig. 3(b)[3]. Hence the summation term is approximately equal to 0.14×T, and the estimated input voltage is $(v_{inv} + 0.14\times T)$.

Therefore when the switching MOS is in the saturation region and the PMOS transistors with their own effective β_p are in the linear region, is

$$b_{eff} = b_1 \cong b\left[\frac{V_{inv}}{(v_{inv} + T\times 0.14)}\right] .$$

As V_{out} continues to drop, the PMOS transistors in the pull-up network will go into their saturation region and change their effective β_p. To deal with this situation, one can either modify the effective β_p directly or continue to modify the β_{eff} of the pull-down network to compensate for the change in β_p. We chose the later approach because we can use interpolation to easily approximate the modification for β_{eff}.

Before developing the interpolation approach, consider the next region where the switching MOS goes into the linear region. Here the MOS can be modeled as an on-channel resistor except that its V_{DS} is not the whole output voltage drop and the devices above the switching MOS will move into the linear region one by one. Hence instead of using the traditional β_{eff} value of β/n, a correction term is needed. Thus, the effective transconductance when the switching MOS is in the linear region is approximated by $b_{eff} = b_2 = m\frac{b}{n}$ where m is a constant determined empirically. We have found that m = 0.75 works well when the number of devices below the switching MOS range from 0 to 5, which is usually the case for a NAND gate.

Returning to the region where both the complementary PMOS and the switching MOS are in the saturation region, by interpolation from the other two cases presented, we get

$$b_{eff} = b\,[(\frac{V_{DD} - V_{in}}{V_{DD} - v_{inv}})(\frac{b_1 - b_2}{b}) + \frac{m}{n}] .$$

Because the above approximation involves the input V_{in} and may lead to difficulty in finding closed-form solutions, it can be further approximated by

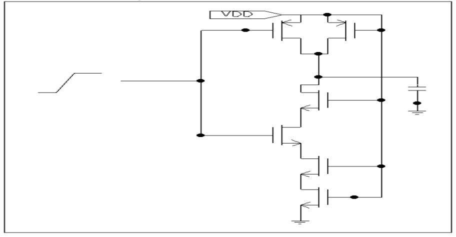
$$b_{eff} = ab_1 + (1-a)b_2 ,$$

where α is an experimental constant. We have found that α = 1/3 works well when the number of devices below the switching MOS range from 0 to 5.

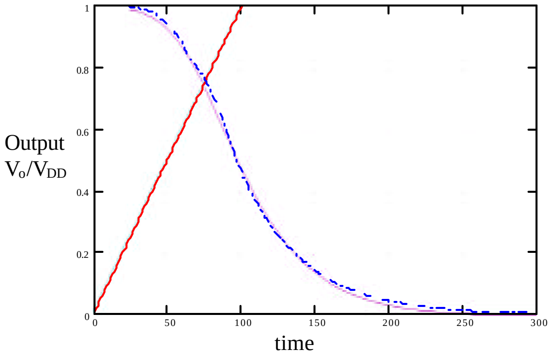
2.2.2 Parallel MOS

When propagating noise through a CMOS gate, since all side fan-in's have to be set to their non-controlling values, the parallel network is reduced to a single transistor whose gate is connected to the switching input.

Fig. 4 illustrates the collapsing technique. The input is a ramp of rise time of 100ps and the load is 20fF. All device sizes are (4u/0.8u) and we assume all capacitances are lumped into the output load. The dash curve is the output waveform of the equivalent inverter obtained using the collapsing technique, and the solid curve was obtained by SPICE simulation.



(a)



(b)

Fig. 4 (a) Circuit for collapsing NAND gate into an equivalent inverter, (b) model and SPICE simulation results.

2.2.3 Internal Capacitance

Internal capacitances are usually ignored when they are small compared to the load capacitance, but often this is not the case when a large number of transistors are connected in series. The easiest way to take into account the effects of internal capacitance is to add it to the load capacitance. But this results in an overestimation of the propagation delay and output transition time.

Hence our approach is to model MOS devices as ON-channel resistances and use the Elmore delay model to obtain the equivalent load capacitance at the gate output.

2.2.4 Multiple input transitions

Computing t_r (or t_f) for transition signals is complicated when more than one input of a gate switches. Consider a NAND gate with multiple (q) switching inputs. First we apply the method in section 2.2.3 to lump all internal capacitances to the output load. That is, all C_p's below the lowest switching MOSFET are discharged to "0" and, depending on the current state of the circuit, either all other internal C_p's or only those above the highest switching MODFET are added to the output load. Then, we re-order the series connected MOSFETs so that the number of "ON" transistors below the lowest switching

[3] Analysis done for 0.8um CMOS technology.

device remains the same as before, all q switching devices are then put in series, and finally the remaining "ON" MOSFETs are put on top. The next step is to merge all q switching MOSFETs into one equivalent switching device. This is accomplished by setting the effective β of these switching MOS to β/q. Let the switching inputs be V_{in}^1, $V_{in}^2,\ldots,V_{in}^q$. The effective input is selected as the input V_{in}^i to the MOS device such that $t_{ai} + t_{ri} \geq t_{aj} + t_{rj}$ for all j, where t_{ai} is the arrival time of transition i, and t_{ri} is the rise time of transition i.

Then the series connected MOS chain is reduced to the circuit model of Fig. 3(b).

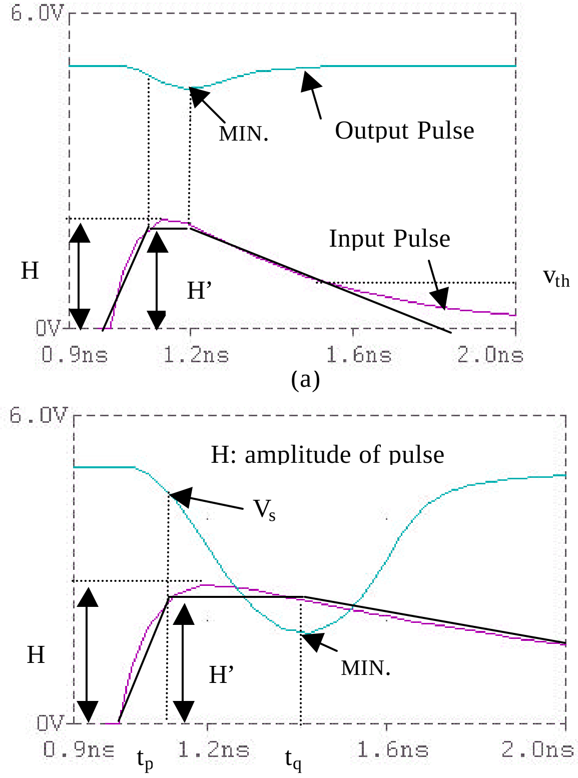
2.3 Noise propagation

When a crosstalk noise (a pulse) passes through a gate, it can be either attenuated or amplified depending on its amplitude H and width W. Fig. 5 shows a simulation result of crosstalk noise propagate through an inverter. In Fig. 5(a) the output noise is small because of its small input noise amplitude and width. On the other hand, the input noise in Fig. 5(b) is sufficient to produce a large output pulse. Note that the output reaches its minimum after the amplitude of input noise starts to decrease. In addition, the output pulse is almost symmetric with respect to t_q, the time it reaches its minimum value.

There are two obvious ways to obtain the output waveform as a function of the input waveform. The first is to use the crosstalk waveform equations developed in [11] convolved with the equations described in section 2.1. The second is to use a piece-wise linear model of the input noise and approximate the output response using the transformation developed in the previous sub-sections. The latter technique is preferred because it is both accurate and computationally efficient. Let the value of the input voltage be H' when the output reaches its minimum. There are two instants of time where the input has the value H', labeled t_p and t_q in Fig 5(b). We approximate the input pulse waveform by three linear segments, as shown in Fig 5, namely

1) a rising ramp from the start of the noise until the input reaches the value H' at time t_p,
2) a constant value of H', and
3) a falling ramp from H' at time t_q and going through the point where the input voltage drops to v_{th}.

Assume H' is a linear function of H, i.e., H' = ρH for 0≤ρ≤1. Experimental results show that when H is in the range of 2-5V, ρ is in the range from 0.85-0.87. By using the crosstalk pulse equations in [11], the slope and time period of each segment can be easily determined. With this piece-wise linear approximation of the noise waveform, we can apply it to the inverter model described in section 2.1 to obtain the output response.



(a)

(b)

Fig. 5 Crosstalk pulse passes through an inverter (a) a small input pulse, (b) a large input pulse.

To complete our model we need to set a critical voltage v_x such that a pulse with amplitude less than v_x will be attenuated and one larger than v_x will be amplified. This critical voltage v_x can be defined as the input voltage such that dV_{out}/dV_{in} = -1. Since this point resides in the region that is modeled by the circuit in Fig. 1(b), the following results are obtained from equation (1).

$$\frac{dV_{out}}{dV_{in}} = P \cdot e^{\frac{-t_r(V_{in}-v_{tn})}{R_pC}} \cdot \frac{-t_r}{R_pC} + A(V_{in} - v_{tn}) + B$$
$$= -1. \qquad (4)$$

Solving for V_{in} we can obtain the critical voltage v_x. An approximate value of v_x can be found by using a Taylor series expansion for the exponential term.

If H is smaller than v_x, the circuit model in Fig. 1(b) is used to determine the output response. First we apply the first segment of the noise waveform, i.e. the rising ramp, to the model in section 2.1 and obtain the output voltage drop to V_s at time t_p as shown in Fig. 5(b). Then the second segment, a level voltage of value H' is applied to continuing discharge the output. Similar to the process in section 2.1 except the input is now held constant at H', we obtain the output response as

$$V_{out} = Pe^{-t/R_pC} + MR_pC \ , \qquad (5)$$

where $R_p = \frac{1}{V_{DD}\boldsymbol{b}_p(H'-1-v_{tp})Coef1}$,

$$P = e^{t_p/R_pC}(V_s - MR_pC),$$

$$M = \frac{1}{R_pC} - \frac{V_{DD}\boldsymbol{b}_n}{2C}(H'-v_{tn})^2,$$

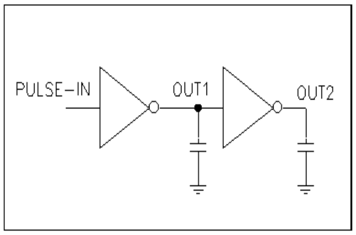
and Coef1 is an experimental correction term and is a function of H.

Hence the minimum output voltage is obtained as $V_{out}(t_q)$.

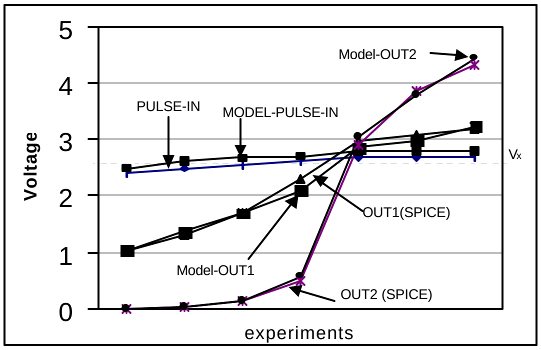
For values of $V_{in} > v_x$, a change, dV_{in}, in the input voltage will cause a change, dV_o, in the output voltage such that dV_o will be greater than dV_{in}, i.e. the circuit is in the amplification mode. The circuit model in Fig. 1(c) will be used to determine the output response. Similar to the above process for the case of a small pulse, we obtain the output response as $V_{out} = Z \cdot t + (V_s - Z \cdot t_p)$, where

$$Z = \left[\frac{\boldsymbol{b}_pV_{DD}}{2C}(H'-1-v_{tp})^2 - \frac{\boldsymbol{b}_nV_{DD}}{2C}(H'-v_{tn})^2\right].$$

If the output voltage continues to drop, the NMOS transistor will pull out of saturation and move into the linear region, and the inverter will no longer operate in the amplification mode. The circuit model in Fig. 1(d) is then used to calculate the output response. The resulting equations for the output response are similar to equation (5), except the roles of the NMOS and the PMOS transistors are interchanged and the coefficients are different. Again the corresponding minimum output voltage is $V_{out}(t_q)$.



(a)



(b)

Fig. 6 (a) Circuit for measurement for input and output pulses (b) Comparison of the model and SPICE results.

After the output reaches its minimum voltage, the third segment of the model, the falling ramp, is applied to the inverter model to obtain the recovery portion of the output waveform. This is the reverse of the previous processes in obtaining the discharging waveform. However, since we already observed that the output waveform is almost symmetric around t_q, another approach to obtain the recovery portion of the output waveform is just to reflect the discharging part of the waveform with respect to the axis t_q. The error caused by this "reflection" method is mainly in the tail portion of the output waveform. Since the variance in the tail portion is less than the device threshold voltage (v_{tn} or v_{tp}), this approximation has a negligible effect on the results.

Propagation of this output pulse through the next level of gates is done in a similar way. Fig. 6 shows a comparison of this approach with SPICE results. Here we see that for an input height equal to about v_x – 0.2V, the pulse at OUT1 is about 1V and is essentially zero at OUT2. For an input of about v_x + 0.1V= V^*, the pulse at OUT1 is more than V^*, and that at OUT2 is almost 5V.

III. Test Generator

In this section we present a modified PODEM algorithm to generate tests for crosstalk noise. This algorithm not only considers effects such as speedup, slowdown and pulses as new logic values, but also takes into consideration information such as finite noise energy, and input arrival skews to accurately characterize the noise strength. For a specific crosstalk coupling in a circuit, the objective of this test generator is to generate a pair of vectors that creates a crosstalk effect (pulse) and either a logic error or the maximum noise effect at a primary output. The symbols and value system shown in Table 1 will be used.

3.1 Conditions and cost functions for maximizing crosstalk noise

3.1.1 Conditions of exciting worst case crosstalk noise

Conditions that a two-pattern test must satisfy to generate a crosstalk effect of maximal severity were derived using the expressions developed in [11]. There are three objectives in creating a crosstalk effect of large severity: a weak driver on the affected line (objective 1), a fast signal transition on the affecting line (objective 2), and a propagation path that maintains/amplifies the noise effect until it reaches a PO or a flip-flop (objective 3). These objectives are used to determine conditions to be satisfied for maximizing the observed crosstalk noise. In Table 2 we list the conditions for each objective for a NAND gate. Similar conditions are established for other gate types. The objective line (affecting line, affected line, …) is assume to be fed by the gate of the type shown in the third column. Conditions in Table 2 are used by the backtrace process to select PI assignments that maximize crosstalk noise.

Note that for the propagation of a pulse (objective 3), only constant values are allowed at side fan-in's. This is because a transition aligned with a noise pulse will significantly decrease the amplitude and width of the pulse. Since each signal has an arrival time t_a and transition time $t_r(t_f)$ associated with it, the algorithm can determine whether a signal transition occurs before, after, or at the same time as a pulse. That is, a transition occurring long before (after) a pulse can be modeled as the final (initial) value of that transition with respect to a pulse.

3.1.2 Cost functions

Since the objective of this TG is to create the maximum noise at a primary output, in addition to the conditions in Table 2 we need a cost function that can guide the search for PI assignments as well a path from the source of the noise to a PO.

The cost function contains a digital and an analog part. The digital part deals with controllability and observability measures [12], and is used to break ties. The analog part of the cost function is a measurement of the gate's capability to propagate noise and is dependent on the gate's strength, i.e. effective β, load capacitance, and gate type such as static, dynamic, domino or latch. Consider a simple static gate such as the inverter in Fig. 1(a). When a positive pulse is applied to this inverter, the circuit model in Fig. 1(b) and/or (c) are used to obtain the output response. Since the PMOS current reaches its maximum when the transistor enters the saturation region, the influence of the PMOS in Fig. 1(c) is greater than that in Fig. 1(b). Re-arranging the differential equation for the circuit in Fig. 1(c) gives

$$C\frac{dV_o}{dt}=\frac{V_{DD}}{2}\left(V_{in}-v_{tn}\right)^2\left[\beta_n\left(1-\frac{\beta_p}{\beta_n}\left(\frac{V_{in}-1-v_{tp}}{V_{in}-v_{tn}}\right)^2\right)\right]$$

$$=\frac{V_{DD}}{2}\left(V_{in}-v_{tn}\right)^2\beta_{neff}\ .$$

Thus the effect β_n as a function of input V_{in} is

$$\beta_{eff}=\beta_n\left(1-\frac{\beta_p}{\beta_n}\left(\frac{V_{in}-1-v_{tp}}{V_{in}-v_{tn}}\right)^2\right).$$

The input V_{in} can be any value between V_{-1}, defined as the point in the DC characteristic where dV_{out}/dV_{in} = -1, and v_{inv} so that β_{eff} becomes a constant value and can be used as an index to define the analog cost function. Using the circuit model in Fig. 1(c) the analog cost function is defined to be:

$$Cost=\frac{C_{load}}{\beta_n[1-\frac{\beta_p}{\beta_n}\left(\frac{V_{-1}-1-v_{tp}}{V_{-1}-v_{tn}}\right)^2]},$$

where C_{load} is proportional to the number of inputs and fanouts of the gate.

We have defined an analog cost for different types of gates in the same manner.

This cost function quantifies the difficulty with which a pulse penetrates through a gate. For instance, the load capacitance serves as a charge pool to mitigate the noise, therefore the larger the output capacitance the smaller the output pulse. On the other hand, the larger the β_{eff} the stronger the pull-down strength. Hence a small pulse can easily discharge the output.

After the analog cost of each gate is obtained, the cost of a path can be obtained by combining these cost values in a manner similar to calculation of observability costs [12]. The computation of the analog cost of a path starts from the primary outputs (i.e. the last level of the gates) and then the circuit is traversed backward to accumulate the cost of each gate. Thus, to propagate a noise effect we can select a path whose cost is the lowest, i.e. propagates the noise with maximum severity. If two paths have the same analog costs, then the digital observability costs are used to break ties.

Table 1 Symbols and parameters used for test generation.

Symbols	Associated parameters	Description	Description for parameters
1	-	constant 1	-
0	-	constant 0	-
P_p	t_a, H, t_H, t_p, t_q	positive pulse	T_a: arrival time; H: amplitude; t_H: peak time; t_p, t_q: as in section 2.3
P_n	t_a, H, t_H, t_p, t_q	negative pulse	
T_u	t_a, t_r	rising transition	t_a: arrival time t_r/t_f: rise/fall time
T_d	t_a, t_f	falling transition	
X	-	Unknown	-

Table 2 Conditions for achieving three objectives.

Objective	Target value	Gate type	Necessary condition	Preferred condition on side fan-in	Sufficient condition on side fan-in
1	0	NAND	All inputs are 1	-	All 1
1	1	NAND	0 at one input	All 1	1 or T_u or T_d or 0
2	T_u	NAND	T_d at one input	All T_d	T_d or1
2	T_d	NAND	T_u at one input	All 1	1 or T_u
3	$P_p(P_n)$	NAND	$P_n(P_p)$ at one input	All 1	1 when $P_n(P_p)$ arrives

3.2 Test generation algorithm

The algorithm consists of three major steps: (1) set the affected line to a constant value (0 or 1); (2) set the affecting line to a transition (T_u or T_d); and (3) propagate the crosstalk effect. We have found that crosstalk noise is more sensitive to driver strength of the affected line than to signal transition time on the affecting line. This occurs because the crosstalk amplitude is inversely proportional to the affected line driver strength, but the logarithm of the amplitude is inversely proportional to the transition time on the affecting line. Hence, if a choice must be made, we prefer to establish a weak driver condition. The flowchart of the algorithm is shown in Fig. 7.

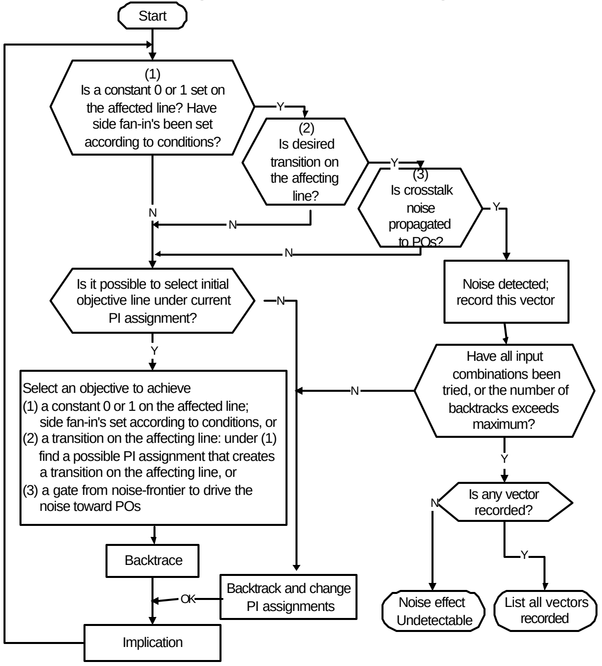

Start
(1)
Is a constant 0 or 1 set on the affected line? Have side fan-in's been set according to conditions?
Y
(2)
Is desired transition on the affecting line?
Y
(3)
Is crosstalk noise propagated to POs?
Y
N
N
N
Noise detected; record this vector
Is it possible to select initial objective line under current PI assignment?
N
Y
Select an objective to achieve
(1) a constant 0 or 1 on the affected line; side fan-in's set according to conditions, or
(2) a transition on the affecting line: under (1) find a possible PI assignment that creates a transition on the affecting line, or
(3) a gate from noise-frontier to drive the noise toward POs
N
Have all input combinations been tried, or the number of backtracks exceeds maximum?
Y
Is any vector recorded?
N
Y
Backtrace
Backtrack and change PI assignments
OK
Noise effect Undetectable
List all vectors recorded
Implication

Fig. 7 Flowchart of the algorithm.

The outline of the algorithm is explained as follows.

1) Upon START, affected and affecting lines are selected. If the signal values are undecided, then the initial objectives are to set up desired values on these lines and lead to the creation of crosstalk condition at the fault site. Conditions in Table 2 and analog cost functions are used to guide the backtracing direction so that a weak affected line driver and a fast transition on the affecting line can be satisfied.
2) Once these assignments are made, forward imply and evaluate the actual transition rate on the affecting line. The evaluation procedure first converts a CMOS gate into an equivalent inverter (Section 2.2) and then uses the inverter model (Section 2.1) to obtain the output response. After the output waveform is obtained, the delay time and rise/fall time can be determined. After a transition is set on the affecting line, the crosstalk noise pulse can be computed according to equations in [11], and P_p or P_n is set on the affected line.
3) We utilize the analog cost function to select a path to propagate the noise to a PO. The cost of a path is obtained by accumulating the cost of gates started from POs. Each time the noise is to be propagated, we select a path from the current site with the minimum cost.
4) When driving the crosstalk noise through a gate, we compute the amplitude of the output noise using the method described in section 2.3. If the amplitude is less than v_x, then the noise is ignored. If the amplitude is greater than some value, say 4V, then the noise will be amplified due to the gain of CMOS logic gates and become a logic hazard. Hence if the amplitude is larger than 4V, we model the pulse as a static hazard and propagate it without computing its strength.
5) If the noise effect has not reached a PO, then one of the internal signal lines which has a "noise signal" value such as P_p or P_n is used as an objective. If the noise effect reached a PO, then the PI assignments are recorded as the test vector together with the corresponding noise amplitude and width.
6) Because we desire a test that creates the maximum noise at an output, we continue to backtrack so that all possible PI assignments are explored. Each time we find a vector pair that creates and propagates the noise to a PO, we record it and the corresponding noise amplitude.
7) The signal value 0, 1, T_u, or T_d can be assigned to primary inputs. Whenever a PI is set to a value the implication procedure is performed and the analog timing/strength information of some signals may have to be re-computed.

3.2.1 Utilization of conditions

In section 3.1 we described conditions to help create a large value of crosstalk noise. These conditions are used together with the backtrace function to achieve certain objectives. For example, assume that we want to set a "0" on the affected line that is fed by a NOR gate. This can be achieved by setting a "1" on any of the gate's inputs. Instead of stopping at this point and continuing to the next objective, from the preferred conditions we know that setting other inputs to "0" will make this NOR gate weaker in terms of its driving strength and the noise will become larger. Hence setting all other inputs to the non-controlling value is added to the initial objective list. However it is not necessary to satisfy all these objectives.

Similar steps are performed for conditions in creating a faster transition on the affecting line.

If no tests are recorded when the algorithm terminates, then either the initial crosstalk conditions could not be excited, or a severe noise effect could not be propagated to a PO. If one or more tests were recorded then based on their amplitude, width and values of their time constants, the worst one can be identified. These tests can then be simulated using SPICE to get a more accurate measure of the noise.

IV. Example and Discussion

The test generation algorithm described in the forgoing was implemented in C language and applied to several benchmark circuits described at the gate level to generate corresponding crosstalk test patterns. The program was run on a Pentium II 266 MHz desktop.

ISCAS ’85 benchmark circuits were used for the experiments are. Since no other circuit information such as crosstalk fault locations, polarity of transitions causing crosstalk fault, coupling capacitance, and layout information is available currently to us, the affecting and affected lines’ driver strength and coupling capacitance value are assumed to be sufficient to excite a significant crosstalk noise at a fault site. We assume all devices are 0.8um, the affecting line is driven by a large driver (64um PMOS/16um NMOS), the affected line is driven by a small driver (16um PMOS/4um NMOS), and they run in parallel for 1000um distance. All other gates and wires are assumed to have default device sizes and load capacitances.

Two sets of experiments are performed. In the first experiment a single crosstalk fault is targeted and the proposed algorithm is used to generate all possible tests for the target fault. Test vectors associated with corresponding pulses at POs are recorded so that the test creating the worst case pulse at a PO can be identified. The experimental results are shown in Table 3. In Table 3 PO denotes primary output (number is the node number), first_p_amp is the height of the pulse at the fault site, and amplitudes at the end of each line is the amplitude of the pulse at the corresponding output. Pulse amplitudes are normalized w.r.t. Vdd. The output statistics characterizing the output pulses into voltage ranges. The results correlate well with SPICE simulations. Due to limitation of space, the result is shown only for a small circuit. Similar experiments have been performed on other ISCAS circuits with large number of nodes.

In the second experiment, for each circuit, 500 pairs of affecting and affected lines are selected at random without considering the circuit structure. The proposed algorithm is applied to generate one test for each fault. Since a thorough search for test patterns for these many faults may require many backtracks, the maximum number of backtracks per fault is limited to 1000. Results of the experiments are shown in Table 4.

In Table 4, Column 2 shows the number of faults for which tests can be successfully generated. Column 3 shows the number of faults for which an appropriate test does not exist to propagate a crosstalk fault to a PO with significant amplitude (i.e. >0.2Vdd), and Column 4 shows the number faults for which the number of backtracks exceeds the maximum setting and the TG process was aborted. Column 5 indicates the TG efficiency (Column 2 plus Column 3 divided by 500), and Column 6 is the CPU time to generate test patterns, expressed in seconds.

Table 3 Results of experiment 1: all tests for a single fault.

Circuit c17.i
Affecting node 16 with rising transition,
Victim node 10 with value 0
Total 3 set of vectors: 12 out of 1024 combinations
1T_d1T_uX first_p_amp=0.676 PO=22 type=P_n amp=0.926
1 T_d 10X first_p_amp=0.541 PO=22 type=P_n amp=0.402
111T_uX first_p_amp=0.546 PO=22 type=P_n amp=0.477
Output statistics

0.2-0.4Vdd	0.4-0.6Vdd	0.6-0.8Vdd	>0.8Vdd
0	2	0	1

Total CPU run_time = 1 seconds

Table 4 Result of experiment 2: (one test for each fault)
Number of faults: 500.

Circuit name	Successful TG		TG Abort-ed	ATPG Effici-ency	TG time (s)
	Detect -ed	Undetect -able			
C432	146	70	284	43.2%	952
C880	270	32	198	60.4%	707
C1908	256	48	196	60.8%	2298
C2610	308	22	170	66.0%	1090
C3540	123	99	278	44.4%	6182
C5315	368	49	83	83.4%	1671
C7552	291	38	191	65.8%	3552

Although in the above experiments the device sizes, coupling capacitance, and related information are artificially inserted, the results in Table 4 demonstrate that the proposed algorithm can generate tests for circuits of reasonable sizes (such as a single functional block) within acceptable amount of time. That is, if all appropriate circuit and layout information is available, our algorithm can identify whether a significant crosstalk fault can be created and propagated to POs and generate an appropriate test.

While the ATPG efficiency is high for some circuits, for other circuits it is low. The main reason for this is the nature of PODEM which pursues only one objective at a time, even when several conditions that must all be satisfied for the detection of a fault are known. This causes unnecessary backtracks and decreases ATPG efficiency, especially when the limit on the number of backtracks is low. We are currently modifying our ATPG to incorporate multiple-backtrace and other feature from FAN [15]. We believe that this will greatly increase the ATPG efficiency.

Since the execution of the proposed algorithm requires a certain amount of calculation time, test pattern generation for all signal pairs of a complex circuit is not practical. Therefore, only critical pairs of lines should be targeted. The selection of these critical lines should be based on the circuit configuration, manufacturing process information, layout, designer's knowledge and other relevant information. This information is typically known in advance to the TG process, and should enable exclusion of some faults that cannot possibly cause errors at outputs.

V. Conclusion

In this work an algorithm to generate tests for crosstalk effects is proposed. This algorithm not only considers noise effects such as speedup, slowdown and pulses as new logic values, but also takes into consideration information such as finite noise energy and input arrival skews to accurately characterize noise strength. Several new techniques have been developed so that tests can be efficiently and accurately generated for what is essentially an analog effect, namely crosstalk noise. These techniques include new models for a CMOS inverter, methods to calculate inverter output response for pulse inputs, a method for collapsing CMOS gates into equivalent inverters, and a piece-wise linear model for pulses. These techniques were integrated into a test generation framework that takes into account several attributes such as noise strengths and signal arrival times and identifies test patterns that maximize crosstalk noise at POs while satisfying a given set of Boolean constraints. Conditions that help create the maximum crosstalk noise were identified and an analog cost function was proposed to select a preferred noise sensitive path. An analog approach is presented that estimates the noise size at each step to ensure the worst case noise is generated. Because crosstalk noise is a finite energy transient effect, it may be filtered if propagated though gates that have significant inertial delay. The proposed algorithm can generate tests that direct the noise through noise sensitive paths toward POs. The proposed algorithm can also generate all tests for a crosstalk effect so that a matching with functional tests can be performed to determine whether the functional tests cover the tests for the crosstalk effects. In this first version of our TG algorithm several factors have been ignored, such as complex gates and multiple crosstalk effects. These will be incorporated in future versions.

References

[1] A. Rubio, N. Itazaki, X. Xu and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol.13, pp.387-394, March 1994.

[2] M. A. Breuer and S. K. Gupta, " Process aggravated noise (PAN) : new validation and test problems", Proc. Int'l Test Conf., pp. 914-923, 1996.

[3] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 16, pp. 290-298, March 1997.

[4] N. Hedebstierna and K. O. Jeppson, "CMOS circuit speed and buffer optimization", IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, Vol. 6, pp.270-281, March 1987.

[5] A. I. Kayssi, K. A. Sakallah, and T. M. Burks, " Analytical transient response of CMOS inverters", Trans. Briefs, IEEE Trans. on Circuit and Systems, Vol. 39, pp.43-45, January 1992.

[6] T. Sakurai and A. R. Newton, "Alpha-power law MODFET model and its applications to CMOS inverter delay and other formulas", IEEE Journal of Solid-State Circuits, Vol. 25, pp. 584-593, April 1990.

[7] Y. H Jun, K. Jun, and S. B. Park, "An accurate and efficient delay time modeling for MOS logic circuits using polynomial approximation", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 8, pp.1027-1032, September 1989.

[8] T. Sakurai and A. R. Newton, "Delay analysis of series-connected MOSFET circuits", IEEE Journal of Solid-State Circuits, Vol. 26, pp. 122-130, February 1991.

[9] J. T. Kong and D. Overhauser, "Methods to improve digital MOS macromodel accuracy", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, pp. 868-881, July 1995.

[10] A. Nabavi-Lishi and N. C. Rumin, "Inverter models of CMOS gates for supply current and delay evaluation", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 13, pp. 1271-1279, October 1994.

[11] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Analytic models for crosstalk delay and pulse analysis for non-ideal inputs", Proc. Int'l Test Conf., pp. 809-818, 1997.

[12] M. Abramovici, M. A. Breuer, and A. D. Friedman, **Digital Systems Testing and Testable Designs**, IEEE Press, 1990.

[13] The national technology roadmap for semiconductors, 1997. (See the web page http:// www.sematech.org).

[14] K. T. Lee, C. Nordquist, and J. A. Abraham "Automatic test pattern generation for crosstalk glitches in digital circuits", Proc. VTS, 1998.

[15] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms", IEEE Trans. on Computers, Vol. C-32, No. 12, pp.1137-1144, December 1983.