A New Framework for Static Timing Analysis, Incremental Timing Refinement, and Timing Simulation

Liang Chi Chen
Professor Melvin Breuer

Technical Report Number: CENG 00-01

Department of Electrical Engineering – Systems
University of Southern California
Los Angeles, California 90089-2562
213-740-4465

A New Framework for Static Timing Analysis, Incremental Timing Refinement, and Timing Simulation

Liang-Chi Chen, Sandeep K. Gupta, and Melvin A. Breuer

Department of Electrical Engineering - Systems

University of Southern California

Los Angeles, CA 90089-2562

{lichen, sandeep, mb}@poisson.usc.edu

In this paper we present a framework that enables the computation of ranges of arrival, transition, and required times - each transition (rising or falling) at each circuit line, for a given sequence of two partially specified vectors. At one extreme where the vectors are completely unspecified, this framework becomes identical to *static timing analysis (STA)*. At the other extreme, when the vectors are completely specified, this framework performs *timing simulation (TS)*. Our key motivation for developing this framework was to reduce the amount of search required by any test generator that uses timing information. During test generation for a target, values are specified incrementally and this framework enables refinement of timing. We have demonstrated elsewhere [1] that this significantly accelerates test generation. In this mode, the proposed framework is said to be performing *incremental timing refinement (ITR)*.

1. Introduction

Static timing analysis (STA) [2] as well as timing simulation (TS) are both integral parts of validation of timing of a circuit. In the former case, minimum and maximum values of transition, arrival, and required times are computed at each circuit line, for the universe of all possible sequences of two vectors. In the later case, computation is performed for a given sequence of two completely specified vectors and, if hazards are ignored, then each range becomes a point.

An intermediate timing analysis situation arises during timing-oriented test generation [3][4][5]. In these test generations, timing of events must satisfy certain conditions for a sequence of two/three vectors to be a test for the target. Due to this reason, timing of transitions are considered as an integral part of test generation.

Test generation for a target fault begins with a sequence of completely unspecified vectors. At this stage STA can be performed to determine the timing ranges of transitions. As test generation proceeds, specific logic values are successively assigned at circuit inputs. Since the application of specific logic values reduced the set of sequences that can be applied, a timing analysis method that can take advantage of this constraints improved by these logic value assignments can help narrow the timing ranges. We call such a generalized timing analysis approach *incremental timing refinement (ITR).* We have demonstrated elsewhere that incremental timing refinement helps significantly reduce the search required during timing-oriented test generation.

In the following, we present a new framework that encompasses STA, ITR, and TS. The development of such a framework requires delay models for gates that are more accurate than the pin-to-pin delay models used for STA (and expressed in the standard delay format - SDF [6]). In addition, the delay model should have certain characteristics that enable identification of the combinations of transition and arrival times at a gate's inputs that lead to a particular extreme for a timing range at its output. In the following we present an enhanced delay model that has these characteristics. This is followed by the development of a STA and an ITR approach that uses the new delay model.

The proposed delay model is compared with many existing models and is shown to accurately capture the effect of many delay phenomena over a wide range of device sizes, loads, transition times and skews (differences between arrival times) of transitions at gate inputs. This effectiveness of ITR is then demonstrated.

In Section 2, previous researches on delay models and static timing analysis are reviewed. In Section 3, the approach to develop our delay model is introduced with the assumptions validated. In Section 4, operations for static timing analysis is developed on our delay model. In Section 5, the developed operations are extended to incremental timing analysis. Experiment results are shown in Section 6.

2. Previous Researches

2.1 Delay Models

Simulators of digital circuits have been developed for different accuracy/computation cost trade-off. *Timing simulators* [7][8] generate voltage waveforms more efficiently (less time) than SPICE-like *circuit simulators* [9], but are less accurate. *Delay calculators* are very efficient in determining circuit delay.

Several approaches for delay calculation have been developed. *Resistance-capacitance (RC) based systems* [10] model a transistor as a resistance charging/discharging a capacitance. *Equation solving systems* [11] solve simplified circuit equations while simulating a circuit. *Analytical delay function systems* [12] substitute input parameters into pre-solved simplified delay equations. Due to the simplicity of these equations, analytical delay function systems developed to date do not provide a sufficient degree of accuracy. *Empirical delay based systems* pre-characterize primitive circuit elements and store the information in *lookup tables* [13][14][15][16] or using *empirical delay functions* [17].

To accurately model the effects of simultaneous input transitions [16][17][18][19], both input transition time and input skew must be considered. The transfer function for modeling transition delay through an inverter is relatively straightforward. To model the effects of multiple input transitions at a gate is quite complex. Often this problem is mapped into an "equivalent" inverter problem where the multi-input gate is modeled as an "equivalent" inverter, and the multiple input transitions are mapped into a single transition at the inverter's input. In some approaches researchers have obtained an equivalent inverter for a gate by replacing (collapsing) parallel transistors by a single transistor whose width is the sum of the widths of the transistors in parallel. In [17] and [18], the authors provided better models for finding an equivalent inverter, but their models can result in significant errors because *input transition time* and *input skew* are ignored in some situations.

A table lookup method considering simultaneous switches is proposed in [16]. In addition to the large amount of simulation effort required for building the tables, the input variables to the table are not independent of others.

2.2 Static Timing Analysis

In this paper we propose a new method for delay calculation to handle simultaneous to-controlling transitions. Considering the same input variables as in [16], our method finds more accurate empirical formulas than that in [17] and [18], and have no significant errors in any special cases. The model has been validated using arbitrary skews over a typical range of input transition times.

Calculating the timing requirements forward and backward, *static timing analysis (STA)* [2] provides vector-independent min-max timing range for rising and falling transitions on each line in a circuit. The accuracy of STA depends on the timing model used. Although traditional timing/circuit [7][8][9] simulators and delay calculators [10]~[19] provide various timing models, they can only handle fully specified input vectors. Unable to apply all possible fully specified vectors to these timing models, STA can not use them to provide more accurate timing information.

By identifying the worst case corner for each quantity computed, we developed STA's forward and backward calculation methods on our timing model, and so provide a more accurate STA system.

Since STA considers the input vectors as fully unspecified, realistic timing ranges should be more specific when the input vectors are partially/fully specified. *Incremental timing refinement (ITR)* [5] starts with the timing information provided by STA, and refines these data as input values become more specified. For each quantity to be computed, we classified all possible timing assumptions, and refined the timing calculation methods on STA. Our ITR provides refined min-max timing information for partially specified vectors.

3. Proposed Delay Model

A NAND gate with output Z and two inputs X and Y is used as the example for illustrating the definitions. Here Z represents the gate output and also the gate. The **controlling value** of a multi-input gate Z, CV^Z, is the value when applied to any of the gate's inputs, completely determines the value at its output. In the two-value logic system, the **non-controlling value** of a gate Z, $\overline{CV^Z}$, is the complement of its controlling value. The **to-controlling transition** at an input of Z is denoted as a sequence of values <$\overline{CV^Z}$, CV^Z>. If the to-controlling transitions occurs at one or more inputs of a gate, and the gate's non-controlling value is applied to its remaining inputs, then the transition at the gate output is called a **to-controlling response. To-non-controlling transition** and **response** are defined similarly. **Transition time ($T^X{}_{tr}$)** of a transition *tr*, where $tr \in \{R, F\}$, on line X is the time required for a rising transition (**R**) to go from 0.1Vdd to 0.9Vdd and from 0.9Vdd to 0.1Vdd for a falling transition (**F**). **Arrival time ($A^X{}_{tr}$)** of a transition *tr* on line X is the time when the voltage at the output reaches 0.5 V_{dd}. The **skew** ($\delta^{X,Y}$) between two transitions on line X and Y is $A^Y{}_{tr}$ - $A^X{}_{tr}$. The **to-controlling gate delay function $d^Z{}_{tr}$** , defined as $A^Z{}_{tr}$ - min($A^X{}_{\overline{tr}}$, $A^Y{}_{\overline{tr}}$), is the gate delay of Z, where the output transition $tr \in$ {R, F} is a to-controlling response and R = $\overline{F}$, and F = $\overline{R}$. **Pin-to-pin delay** from X to Z is the gate delay of Z when Y is steady at non-controlling value and a transition is applied on X. $\mathbf{d^{Z,\,X}{}_{tr}}$ is the pin-to-pin delay function from X, an input of gate Z, to Z, where the output transition is *tr*. The **to-non-controlling gate delay** is defined as $A^Z{}_{tr}$ - max($A^X{}_{\overline{tr}}$, $A^Y{}_{\overline{tr}}$), where $A^Z{}_{tr}$, the latest output arrival time computed through pin-to-pin delay, is max($A^X{}_{\overline{tr}}$ + $d^{Z,X}{}_{tr}$, $A^Y{}_{\overline{tr}}$ + $d^{Z,Y}{}_{tr}$). **To-controlling transition time function $t^Z{}_{tr}$** and **to-non-controlling transition time function $t^{Z,\,X}{}_{tr}$** are defined similarly.

3.1 Delay Phenomena

3.1.1 Simultaneous switching

SDL [6] provides pin-to-pin delays, i.e., delay from one input pin to the output pin, assuming all side-inputs are steady and thus is not accurate for simultaneous transitions with small skew values. For a two-input NAND gate, the delay when a single input has a falling transition is larger than that when both inputs have simultaneous falling transitions, since in the latter case the output is charged via multiple PMOS transistors (Figure 1). A delay model is developed to capture this phenomenon. Given the input skews and transition times of a gate, our model computes gate delay and output transition time by formulating timing functions using empirical results.

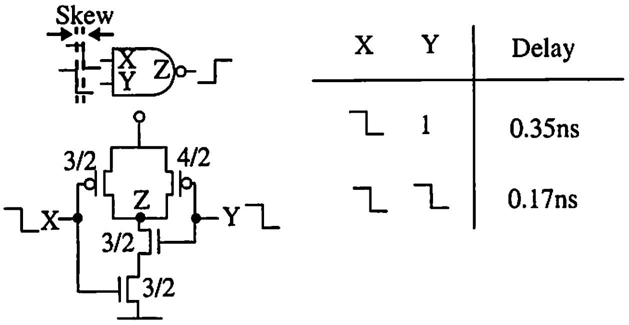


FIGURE 1. Single vs. multiple to-controlling-value transitions at gate inputs.

To explain the speed-up caused by simultaneous falling transitions in Figure 1, we plot the to-controlling gate delay as a function of $\delta^{X,Y}$ for some fixed $T^X{}_F$ and $T^Y{}_F$, where

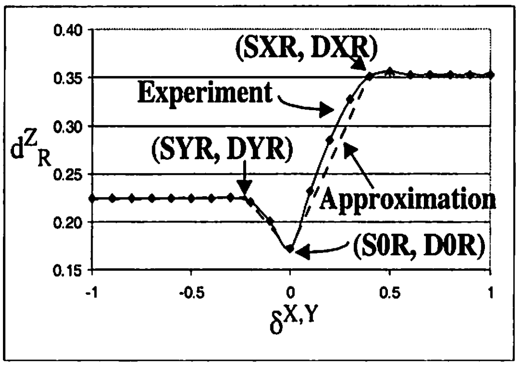


FIGURE 2. Rising delay of two-input NAND gate as a function of $\delta^{X,Y}$ and its linear approximation.

$\delta^{X,Y} = A^Y{}_F - A^X{}_F$ (Figure 2). The speed-up caused by the simultaneous switches is significant only when $|\delta^{X,Y}|$ is small. When $|\delta^{X,Y}|$ is very large, the delay is the same as pin-to-pin delay. A linear approximation is shown in Figure 2. Two transitions to the same value on X and Y are called **δ-simultaneous** if SYR ≤ $\delta^{X,Y}$ ≤ SXR. Output response is speeded up by multiple input transitions only if input transitions are δ-simultaneous.

3.1.2 Input positions

Let **n** be the number of inputs in the NAND gate and $\mathbf{p}^X$ the position of input X in the serial chain (Figure 3). The position of the input closest to the output is defined as 0. According to SPICE simulations, the pin-to-pin rise delay of 5-input NAND gate when a falling transition is applied to the input position 4 to the output may be 50% larger than that at input position 0. The reason is that the pull-up transistor also needs to charge the source/drain capacitances of the serial transistors in the pulldown, that are conducting to the output.

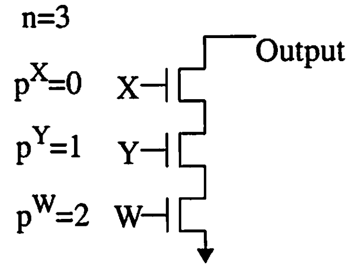


FIGURE 3. Number of inputs and input positions.

3.2 Timing functions (for a two-input NAND gate)

During test generation, all circuit parameters (e.g., device sizes and loads) remain fixed. In contrast, timing parameters (e.g., arrival times, transition times) may change. So the delay and transition times for a two-input NAND gate can be represented by functions of timing variables. As δ-simultaneous to-controlling transitions have a much greater impact on delay than δ-simultaneous to-non-controlling transitions, we hence enhanced the classical delay models for simultaneous to-controlling transitions and use pin-to-pin delay model for to-non-controlling transitions.

We hence characterized a gate's timing behavior via signal arrival times and transition times. Given the *arrival times* and *transition times* of transitions at a gate's inputs, we compute the *gate delay* and *output transition time*. The *output arrival time* of a gate is computed using the input arrival times and gate delay.

We only consider the cases where all inputs of a gate have either non-controlling values or transitions to the same value. The reason is that the non-trivial output response caused by input transitions to opposite values at gate inputs are either (a) two separate transitions in opposite directions that can be processed by separately considering the response of to-controlling transitions and to-non-controlling transitions on the inputs of a gate, or (b) glitches where gate delay and output transition time are usually not the parameters of concern [4]. The gate delay and output transition time of a two-input NAND gate is represented by timing functions defined below:

Fall delay function (from input pin X): $\mathbf{d}^{Z,X}_{F}(T^{X}_{R})$

Fall time function (from input pin X): $\mathbf{t}^{Z,X}_{F}(T^{X}_{R})$

Rise delay function for two simultaneous input switches: $\mathbf{d}^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y})$

Rise time function for two simultaneous input switches: $\mathbf{t}^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y})$

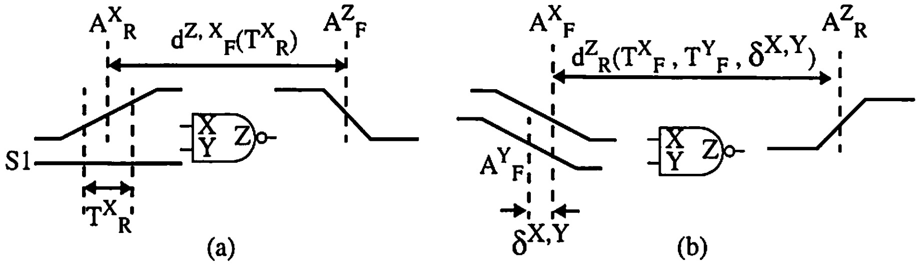


FIGURE 4. (a) Fall delay function and (b) rise delay function.

3.3 Trends with Respect to Single Variables

The relations between output variables and each input variable for a two-input NAND gate (Figure 1) is first explored in Figure 5.

For fixed $\delta^{X,Y}$ and T^{Y}_{tr}, the gate delay as a function of T^{X}_{tr} may be either (1) monotonically increasing or (2) bi-tonic (monotonically increasing and then monotonically decreasing in this case). In case (2), the pin-to-pin delay may become negative for large T^{X}_{tr} .This bi-tonicity is due to the fact that the input transition starts to pull up (down) output voltage before the input transition arrives, i.e., reaches 0.5Vdd. Effective βn/βp ratio determines which shapes the $T^{X}_{\overline{tr}}$ - $d^{Z,X}_{tr}$ curves take. Below we treat this relation as (2), because (1) is a special case of (2) with the curve's peak at infinity. Output transition time

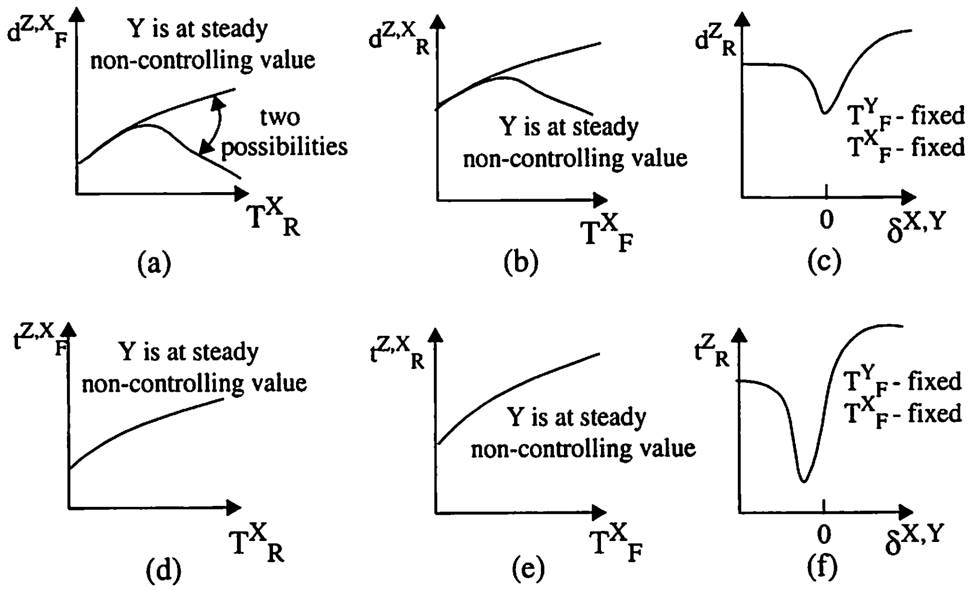


FIGURE 5. Timing functions v.s. input variables.

will always increase as $T^{X}\mathit{tr}$ increases. Delay and output transition times have similar shapes with respect to skew. The minimal delay point is always at $\delta^{X,Y} = 0$, but the minimal transition time point is not.

3.4 Finding Empirical Formulas

$\mathbf{d^{Z,X}_F}$, $\mathbf{d^{Z,X}_R}$, $\mathbf{t^{Z,X}_F}$, and $\mathbf{t^{Z,X}_R}$ are derived as they are in SDL. For small skew, $\mathbf{d^Z_R}$ and $\mathbf{t^Z_R}$ are constructed via simulation and curve fitting. $\mathbf{d^Z_R}$ is constructed as a function of input skew by fixing other variables. This function is represented by a V-shape and the expressions for the three vertices of the triangle are determined in term of input transition times (see Figure 2). So $\mathbf{d^Z_R}$ finally becomes a function of input transition times and input skew. The general forms for the formulas of timing functions are found by curve fitting on empirical results.

The V-shape function has three important points (S0R, D0R), (SXR, DXR), and (SYR, DYR) shown in Figure 2. Here **S0R** = 0, **D0R** is the minimal delay caused by simultaneous transition T^X_F and T^Y_F. **SXR** is the minimum skew $\delta^{X,Y}$ such that transition on Y does not affect the gate delay for fixed T^X_F and T^Y_F. **DXR** is the delay caused by a single transition on X with transition time T^X_F. SYR and DYR are defined similarly. Here D0R and SXR are functions of T^X_F and T^Y_F . DXR is a function of T^X_F. We determined the general forms of DOR, SXR, and DXR from the experimental data, performed curve fitting, and validated the following expressions via simulation.

$DXR(T^X_F) = K_{10}*(T^X_F)^2 + K_{11}*T^X_F + K_{12}$,

$DOR(T^X_F, T^Y_F) = (K_{20}*(T^X_F)^{1/3}+K_{21})*(K_{22}*(T^Y_F)^{1/3}+K_{23})+K_{24}$, and

$SXR(T^X_F, T^Y_F) = K_{30}*(T^X_F)^2+K_{31}*(T^Y_F)^2+K_{32}*T^X_F*T^Y_F+K_{33}*T^X_F+K_{34}*T^Y_F +K_{35}$.

Here the gate delay is defined with respect to the arrival time of the earliest transition at the gate input. $\mathbf{t^Z_R}$ is constructed similarly with $\mathbf{d^Z_R}$ except that SOR for $\mathbf{t^Z_R}$ may be non-zero.

3.5 Validation of the Approximation

In previous sections, we stated that SOR = 0 and used the V-shape approximation for d^Z_R-$\delta^{X,Y}$ curves. Now we validate these two assumptions.

Theorem 1: The minimal delay point in function $d^Z_R(T^X_F , T^Y_F , \delta^{X,Y})$ for NAND gate Z is always at $\delta^{X,Y} = 0$ for all fixed T^X_F , T^Y_F .

Proof: The Proof of this and the subsequent results can be found in [20].

(a) $A^X_F < A^Y_F$:

Compare two cases of simultaneous switches. In case 1, the transition time and arrival time of the transition on X (TranX) and Y (TranY) are A^X_F and T^X_F and A^Y_F and T^Y_F , respectively. In case 2, the same transition, TranX, is applied to X. The transition on Y (TranY') has a smaller arrival time A^Y_F' and same transition time T^Y_F such that $A^X_F \leq A^Y_F' < A^Y_F$.

$A^X_F \leq A^Y_F' < A^Y_F \Rightarrow 0 \leq \delta^{X,Y'} < \delta^{X,Y}$, where $\delta^{X,Y'} = A^Y_F{}' - A^X_F$.

Since TranX is the earlier transition, delay is computed as A^{Z}_{R} - A^{X}_{F} for both cases. Because $A^{Y}_{F}{}'$ < A^{Y}_{F}, the transition TranY' causes the PMOS transistor with input Y to help pull the output high earlier than TranY (both have the same transition time). If output rising transition has not crossed 0.5Vdd when TranY' starts to pull up the output voltage, then delay of case 2 will be smaller than that in case 1. Otherwise, both TranY and TranY' do not affect gate delay and the delay for case 1 and case 2 will be identical.

So $d^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y'}) \leq d^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y})$ for $0 \leq \delta^{X,Y'} < \delta^{X,Y}$, $\forall\ \delta^{X,Y'}$

Since the statement above is true for arbitrary large $\delta^{X,Y}$, $d^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y})$ monotonically increases for $\forall\ \delta^{X,Y} \geq 0$.

(b) $A^{X}_{F} > A^{Y}_{F}$: Similarly, $d^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, 0) \leq d^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y})$ for $\forall\ \delta^{X,Y} \leq 0$.

(c) $A^{X}_{F} = A^{Y}_{F}$: $\delta^{X,Y} = 0$; $d^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y}) = d^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, 0)$.

By (a), (b), and (c), it is proven that the minimal delay points in $\delta^{X,Y}$-d^{Z}_{R} curve is always at $\delta^{X,Y} = 0$.

Theorem 2: The V-shape approximation in Figure 2 can capture the general shape of $d^{Z}_{R}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y})$ for all fixed values of T^{X}_{F} and T^{Y}_{F}.

Proof:

For an arbitrary fixed T^{Y}_{F}, we can always find T^{X1}_{F}, a value of T^{X}_{F}, such that

$DXR(T^{X1}_{F}) = (-0.5 - V_{thp}/V_{dd})\ T^{Y}_{F}$ (see Figure 6).

(T^{X1}_{F} is defined such that for $\delta^{X,Y} = 0$, when transition on Y reaches its threshold V_{dd} - V_{thp}, output has already risen to $0.5V_{dd}$, so the gate delay is not affected by the transition on Y. T^{X1}_{F} is usually quite large and $DXR(T^{X1}_{F}) < 0$.)

Three possible cases are shown in Figure 7.

(a) For the case $T^{X}_{F} \geq T^{X1}_{F}$

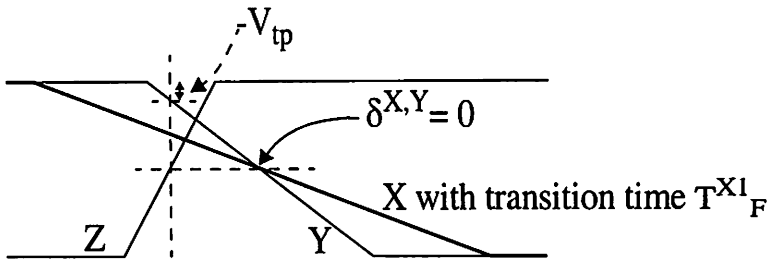
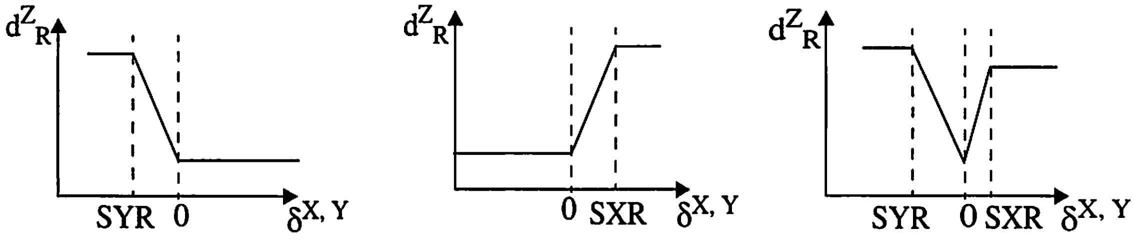


FIGURE 6. T^{X1}_F **such that** $DXR(T^{X1}_F) = (-0.5 - V_{thp}/V_{dd})\ T^Y_F$.



(a) $T^X_F \ge T^{X1}_F$ (b) $T^X_F \le DYR(T^Y_F) * V_{dd}/ (-0.5-V_{thp})$ (c) else

FIGURE 7. $\delta^{X,Y}$ **-** d^Z_R **curves for three possible cases with fixed** T^X_F **and** T^Y_F.

When $\delta^{X,Y} \ge 0$, $d^Z_R(T^X_F , T^Y_F , \delta^{X,Y}) = DXR(T^X_F)$ because before the transition on Y crossed its threshold and start to pull up output Z, Z has already reached $0.5V_{dd}$, so the transition on Y does not help speed up the output transition at all.

When $\delta^{X,Y} < 0$, $d^Z_R(T^X_F , T^Y_F , \delta^{X,Y}) = A^Z{}_R - A^Y{}_F$ (transition on Y crosses $0.5V_{dd}$ earlier than that on X does).

If $\delta^{X,Y} << 0$, Z has reached $0.5V_{dd}$ before transition on X crosses its threshold, $d^Z_R(T^X_F , T^Y_F , \delta^{X,Y}) = DYR(T^Y_F)$ is a constant.

there exists SYR < 0 such that $d^Z_R(T^X_F , T^Y_F , SYR) = DYR(T^Y_F)$ and $d^Z_R(T^X_F , T^Y_F , SYR^+) < d^Z_R(T^X_F , T^Y_F , SYR)$ (See Figure 7 (a))

for ∀ SYR' that satisfies SYR < SYR' < 0, d^Z_R(TXF , TYF , SYR') monotonically decreases as SYR' increase because the earlier the transition on X arrives, the more it will help on pulling up the output Z, that decreases the delay defined as $A^Z{}_R - A^Y{}_F$.

So the delay function $d^Z{}_R(T^X{}_F , T^Y{}_F , \delta^{X,Y})$ can be approximated by two horizontal lines connected by a ramp.

(b) For the case $T^X{}_F \leq DXR(T^Y{}_F) * V_{dd}/ (-0.5 - V_{thp})$

The curve is similar to (a) as shown in (Figure 7 (b))

(c) For the case $DXR(T^Y{}_F) * V_{dd}/ (-0.5 - V_{thp}) < T^X{}_F < T^{X1}{}_F$ (Figure 7 (c))

We divide it into two cases:

<1> the flat segment for $\delta^{X,Y} \leq SYR$ and $\delta^{X,Y} \geq SXR$, and

<2> the monotosity of $d^Z{}_R(T^X{}_F , T^Y{}_F , \delta^{X,Y})$ for $SYR \leq \delta^{X,Y} \leq 0$ and $0 \leq \delta^{X,Y} \leq SXR$

Both cases can be obtained by methods similar to Theorem 1 and Theorem 2(a).

In all the three cases (a), (b), and (c), $d^Z{}_R(T^X{}_F, T^Y{}_F, \delta^{X,Y})$ can be captured by V-shape linear approximation.

3.6 Extended Model

3.6.1 Considering Input Positions

In the previous section, a model for a two-input NAND gate is presented. Here this model is expanded to deal with NAND gates with more than two-inputs.

To consider number of inputs and input positions in our model, **DXR** is redefined as a function of $T^X{}_F$, n, and p^X. Experimental results show the following form.

DXR$(T^X{}_F , n , p^X) = K_{40}*(T^X{}_F)^2 + K_{41}*T^X{}_F + K_{42}*n + K_{43}*p^X + K_{44}$.

A function **Teqv** is defined to convert the transition time at position p^X of an n-input NAND gate to its equivalent transition time that causes the same delay at position 0 of 2-input NAND gate (assume all transistors are of the minimum size.), i.e., **DXR**$(T^X{}_F , n , p^X) =$ **DXR**$(Teqv(T^X{}_F , n , p^X), 2, 0)$. We find

$\mathrm{Teqv}(T^X{}_F\,,\, n\,,\, p^X) = K_{50} - [(T^X{}_F)^2 + K_{51}{*}T^X{}_F + K_{52}{*}n + K_{53}{*}p^X + K_{54})]^{1/2}.$

We use $\mathrm{Teqv}(T^X{}_F\,,\, n\,,\, p^X)$ and $\mathrm{Teqv}(T^Y{}_F\,,\, n\,,\, p^Y)$ to substitute $T^X{}_F$ and $T^Y{}_F$ in formulas D0R and SXR. Then we perform curve fitting again to obtain the new coefficients. So number of inputs and input positions are incorporated into our model.

3.6.2 Considering More Simultaneous Transitions

Here the delay on simultaneous to-controlling transitions is interpreted as other simultaneous input transitions help reduce the delay caused by the *earliest input transition.* We calculate this speedup caused by each later transition using $\mathbf{d}^{\mathbf{Z}}{}_{\mathbf{R}}(T^X{}_F, T^Y{}_F, \delta^{X,Y})$, in the sequence according to their arrival time. In cases where multiple later input transitions participate the speedup, the actual speedup due to each transition is a fraction of speedup calculated above. This fraction is a function of (a) the pin-to-pin delay of the earliest input, and (b) the total speedup caused by the later input transitions considered previously.

4. Static Timing Analysis on Our Delay Model

Static timing analysis provides min-max timing ranges for each line in a circuit for both rising and falling transitions. The ranges are derived independent of input vectors, and represent bounds on minimum and maximum delay values over all pairs of vectors. In timing analysis (Figure 8) arrival times (**A**) and transition times (**T**) at a gate's output are calculated based on these values at gate inputs. These values are computed via a forward traversal starting at the primary inputs. The required time (**Q**) at each input of a gate is calculated based on that at the gate's output. The required times are hence computed via a backward traversal starting at primary outputs. If the arrival time range does not overlap with the required time range for the rising/falling transitions at a line, then the given timing requirements cannot be satisfied and a delay error is found. Delay transfer functions for forward and backward calculations in timing analysis are defined in the proposed model. The min-max ranges in the proposed timing analysis are due to the unspecified input values, pulses, as well as approximations that ignore data dependencies caused by fanouts and reconverges. In our current delay model that handles only transitions in the same direction at the inputs of each gate, pulses are ignored. Hence, if all input values are specified, timing ranges become points.

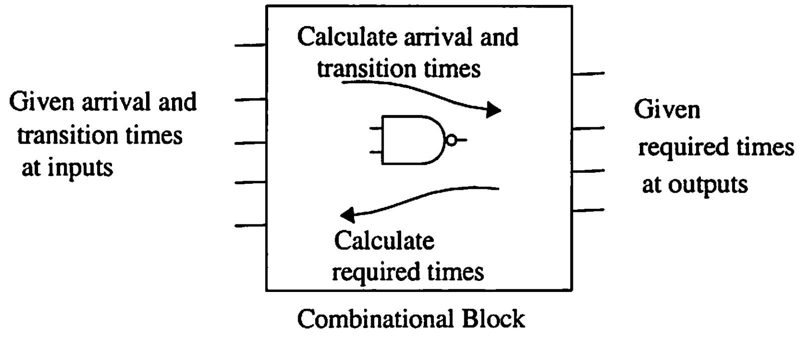

Calculate arrival and transition times
Given arrival and transition times at inputs
Given required times at outputs
Calculate required times
Combinational Block

FIGURE 8. Overall structure of timing analysis.

4.1 The Worst Case Corners for Min-max Ranges

We developed STA's delay transfer functions on a two-input NAND gate based on our delay model. In our min-max range representation, the timing windows in [5] are used and shown in Figure 9. The earliest/latest arrival times and the shortest/longest transition times of rise/fall transitions are recorded for calculating the timing information for next stage. The smallest (largest) arrival time of falling (rising) transition on line X is represented as $A^X_{F,S}$ ($A^X_{R,L}$). Transition and required times are represented similarly.

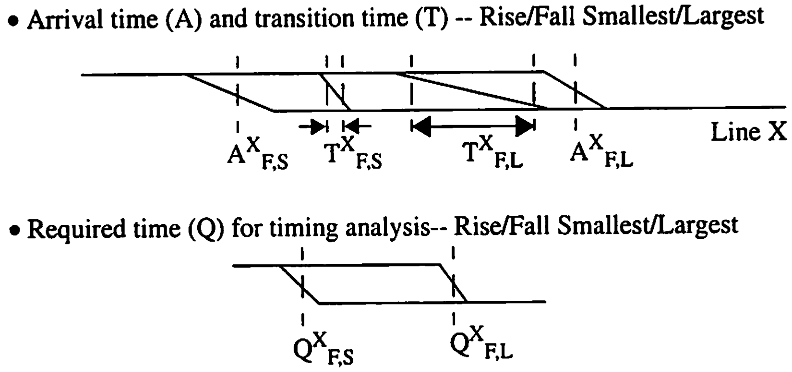

• Arrival time (A) and transition time (T) -- Rise/Fall Smallest/Largest
$A^X_{F,S}$
$T^X_{F,S}$
$T^X_{F,L}$
$A^X_{F,L}$
Line X
• Required time (Q) for timing analysis-- Rise/Fall Smallest/Largest
$Q^X_{F,S}$
$Q^X_{F,L}$

FIGURE 9. Timing information used in our method.

The key issue of min-max timing calculation is to identify the worst case corners for A^Z_{tr}, T^Z_{tr}, and $Q^X_{\overline{tr}}$, where $tr \in \{R, F\}$, based on the characteristics of the delay model, given the min-max timing ranges of X, Y, and Z. The relations between output variables and input variables in Section 3.3 help identify the worst case corners. How to use this information to identify the worst case corners is shown in the rest of Section 4 and Section 5.

4.2 Calculation of Arrival and Transition Times

4.2.1 for Output Falling Transition

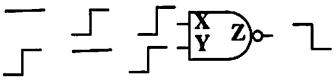


FIGURE 10. Possible input combinations for output falling transition.

Given the arrival and transition times at a gate's inputs, we calculate the corresponding quantities for the gate's outputs. The arrival time for output falling transition is associated with input rising transitions. Recall that pin-to-pin delay is used for to-non-controlling responses. The arrival times and the transition times for output falling transitions are:

$$A^Z_{F,S} = \min\,[A^X_{R,S} + \min\,[\mathbf{d}^{Z,X}_F(T^X_{R,S}),\, \mathbf{d}^{Z,X}_F(T^X_{R,L})],$$

$$A^Y_{R,S} + \min\,[\mathbf{d}^{Z,Y}_F(T^Y_{R,S}),\, \mathbf{d}^{Z,Y}_F(T^Y_{R,L})]].$$

$$A^Z_{F,L} = \max\,[A^X_{R,L} + \mathbf{d}^{Z,X}_F(T^{X'}_R),\, A^Y_{R,L} + \mathbf{d}^{Z,Y}_F(T^{Y'}_R)]$$

$$\text{where } T^{X'}_R = T^X_{R,\max}, \quad \text{if } T^X_{R,\max} \in (T^X_{R,S}, T^X_{R,L});$$

$$= T^X_{R,S}, \quad \text{else if } \mathbf{d}^{Z,X}_F(T^X_{R,S}) > \mathbf{d}^{Z,X}_F(T^X_{R,L});$$

$$= T^X_{R,L}, \quad \text{otherwise.}$$

Here, $\mathbf{T}^X_{\mathbf{R,\,max}}$ is the value of T^X_R that maximizes $\mathbf{d}^{Z,X}_F(T^X_R)$. $\mathbf{T}^Y_{\mathbf{R,\,max}}$ is defined similarly.

$$T^Z_{F,S} = \min\,[\mathbf{t}^{Z,X}_F(T^X_{R,S}),\, \mathbf{t}^{Z,Y}_F(T^Y_{R,S})].$$

$$T^Z_{F,L} = \max\,[\mathbf{t}^{Z,X}_F(T^X_{R,L}),\, \mathbf{t}^{Z,Y}_F(T^Y_{R,L})].$$

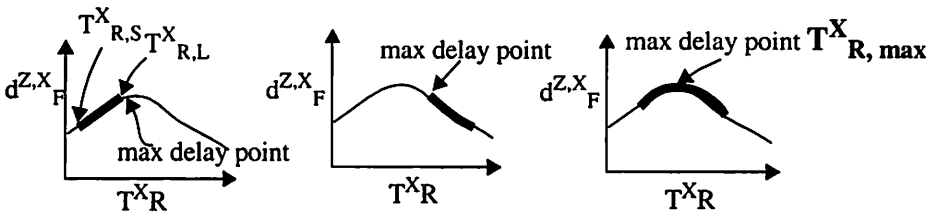


FIGURE 11. Possible transition time min-max range in T^i_R-$d^{Z,i}_F$ curve.

The gate delay from X to Z is minimized when T^X_R is either minimal or maximal (because d_F is bi-tonic with respect to T^X_R, as shown in Figure 5(a)). A^Z_F is smaller if A^X_R is smaller.

A^Z_F is maximum when both the input arrival times and the gate delay are maximal. The maximal gate delay may occur when the input transition times are (a) maximal, (b) minimal, or (c) at some values in between. These three scenarios correspond respectively to the three cases shown in Figure 11, where the min-max range is on the left of the peak, right of the peak, respectively.

The smallest output transition time, $T^Z_{F,S}$, is always caused by smallest input transition time, from either input. $T^Z_{F,L}$ is similar.

4.2.2 for Output Rising Transition

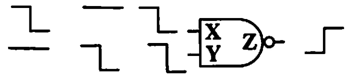


FIGURE 12. Possible input combinations for output rising transition.

$$A^Z_{R,S} = \min\,[A^X_{F,S}\,,\, A^Y_{F,S}] + \underset{\beta,\,\gamma \in \{S,\,L\}}{min}\,[\mathbf{d}^Z_{\mathbf{R}}(T^X_{F,\beta},\, T^Y_{F,\gamma},\, A^Y_{F,S} - A^X_{F,S})].$$

$$A^Z_{R,L} = \max\,[A^X_{F,L} + [\mathbf{d}^{Z,\,X}_{\mathbf{R}}(T^{X'}_F)],\, A^Y_{F,L} + [\mathbf{d}^{Z,\,Y}_{\mathbf{R}}(T^{Y'}_F)]]$$

$$\text{where } T^{X'}_F = T^X_{F,\,max}\,, \qquad \text{if } T^X_{F,\,max} \in (T^X_{F,S},\, T^X_{F,L});$$

$$= T^X_{F,S}\,, \qquad \text{else if } \mathbf{d}^{Z,\,X}_{\mathbf{R}}(T^X_{F,S}) > \mathbf{d}^{Z,\,X}_{\mathbf{R}}(T^X_{F,L});$$

$$= T^{X}_{F,L}, \qquad \text{otherwise.}$$

Here, $\mathbf{T^{X}_{F,max}}$ is the value of T^{X}_{F} that maximizes $\mathbf{d^{Z,X}_{R}}(T^{X}_{F})$. $T^{Y'}_{F}$ is defined similarly.

$$T^{Z}_{R,S} = \mathbf{t^{Z}_{R}}(T^{X}_{F,S}, T^{Y}_{F,S}, SK_{t,R,min})$$

$$\text{if } (A^{X}_{F,S} + SK_{t,R,min}, A^{X}_{F,L} + SK_{t,R,min}) \cap (A^{Y}_{F,S}, A^{Y}_{F,L}) \neq \varnothing;$$

$$= \min[\mathbf{t^{Z}_{R}}(T^{X}_{F,S}, T^{Y}_{F,S}, A^{Y}_{F,S} - A^{X}_{F,L}), \mathbf{t^{Z}_{R}}(T^{X}_{F,S}, T^{Y}_{F,S}, A^{Y}_{F,L} - A^{X}_{F,S})]$$

$$\text{otherwise.}$$

Hence, $SK_{t,R,min}$ is the value of skew $\delta^{X,Y}$ that minimizes $\mathbf{t^{Z}_{R}}(T^{X}_{F}, T^{Y}_{F}, \delta^{X,Y})$ for given T^{X}_{F}, T^{Y}_{F}. Similarly, $T^{Z}_{R,L} = \max[\mathbf{t^{Z,X}_{R}}(T^{X}_{F,L}), \mathbf{t^{Z,X}_{R}}(T^{Y}_{F,L})]$.

For output rising transition to arrive as early as possible, we expect all inputs to arrive as early as possible because an earlier arrival transition helps pull up the output earlier compared with a later one, assuming both have the same transition times. Since the shortest delay may be caused by the shortest or longest transition time according to Figure 11 (but not at any other time in between), A^{Z}_{R} is minimum when both inputs have either the smallest or the longest transition times.

Since simultaneous to-controlling transitions may speed up the output transition, the arrival time of Z is maximized when only one transition occurs. In this situation, the transition time of lagging input does not affect A^{Z}_{R} at all. Considered as a function of input transition time, maximal delay may occur in one of the three cases similar to the ones shown in Figure 11.

Although minimal gate delay always occurs when $\delta^{X,Y} = 0$, minimal output rising transition time may occur when $\delta^{X,Y} = SK_{t,R,min} \neq 0$. $\delta^{X,Y}$ may be equal to $SK_{t,R,min}$ if $(A^{X}_{F,S} + SK_{t,R,min}, A^{X}_{F,L} + SK_{t,R,min}) \cap (A^{Y}_{F,S}, A^{Y}_{F,L}) \neq \varnothing$. Otherwise, either minimal or maximal $\delta^{X,Y}$ closest to $SK_{t,R,min}$ will cause minimal output transition time. Minimal output transition time occurs when both input transition times are minimal since it monotonically increases respect to T^{X}_{F} and T^{Y}_{F}.

The method for calculating the required times for STA can be found in [20].

4.3 Calculation of Required Times

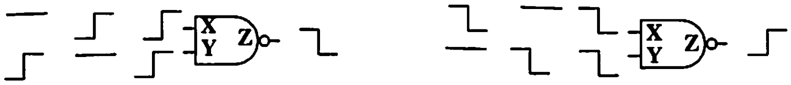


FIGURE 13. Possible input combinations for output falling/rising transition.

Given the required arrival time (Q) at a gate's output, and the minimal/maximal arrival and transition times at its inputs, we calculate Q for each input. There is no required transition time in our analysis. If input arrival time plus the maximal gate delay for this transition is smaller than the minimal output required time, then this timing requirement is not satisfied.

$$Q^X_{R,S} = Q^Z_{F,S} - \mathbf{d}^{Z,\,X}_F(T^{X'}_R),$$

$$\begin{array}{lll} \text{where } T^{X'}_R = & T^X_{R,\,max}, & \text{if } T^X_{R,\,max} \in (T^X_{R,S}, T^X_{R,L}); \\ & T^X_{R,S}, & \text{else if } \mathbf{d}^{Z,\,X}_F(T^X_{R,S}) > \mathbf{d}^{Z,\,X}_F(T^X_{R,L}); \\ & T^X_{R,L}, & \text{otherwise.} \end{array}$$

$\mathbf{T}^{\mathbf{X}}_{\mathbf{R,\,max}}$: T^X_R that maximizes $\mathbf{d}^{Z,\,X}_F(T^X_R)$.

$$Q^X_{R,L} = Q^Z_{F,L} - \min\,[\mathbf{d}^{Z,\,X}_F(T^X_{R,S}), \mathbf{d}^{Z,\,X}_F(T^X_{R,L})].$$

$$Q^X_{F,S} = Q^Z_{R,S} - \mathbf{d}^{Z,\,X}_R(T^{X'}_F),$$

$$\begin{array}{lll} \text{where } T^{X'}_F = & T^X_{F,\,max}, & \text{if } T^X_{F,\,max} \in (T^X_{F,S}, T^X_{F,L}); \\ & T^X_{F,S}, & \text{else if } \mathbf{d}^{Z,\,X}_R(T^X_{F,S}) > \mathbf{d}^{Z,\,X}_R(T^X_{F,L}); \\ & T^X_{F,L}, & \text{otherwise.} \end{array}$$

$\mathbf{T}^{\mathbf{X}}_{\mathbf{F,\,max}}$: T^X_F that maximizes $\mathbf{d}^{Z,\,X}_R(T^X_F)$.

$$Q^X_{F,L} = Q^Z_{R,L} - \min_{\beta,\gamma \in \{S,L\}} [\mathbf{d}^Z_{\mathbf{R}}(T^X_{F,\beta}, T^Y_{F,\gamma}, SK_{d,R,min})]$$

$$\text{if } (A^X_{F,S} + SK_{d,R,min}, A^X_{F,L} + SK_{d,R,min}) \cap (A^Y_{F,S}, A^Y_{F,L}) \neq \varnothing;$$

$$Q^Z_{R,L} - \min_{\beta,\gamma \in \{S,L\}} [\mathbf{d}^Z_{\mathbf{R}}(T^X_{F,\beta}, T^Y_{F,\gamma}, A^Y_{F,S} - A^X_{F,L}),$$

$$\mathbf{d}^Z_{\mathbf{R}}(T^X_{F,\beta}, T^Y_{F,\gamma}, A^Y_{F,L} - A^X_{F,S})] \qquad \text{otherwise.}$$

$\mathbf{SK_{d,R,min}}$: $\delta^{X,Y}$ that minimizes $\mathbf{d}^Z_{\mathbf{R}}(T^X_{F,\beta}, T^Y_{F,\gamma}, \delta^{X,Y})$; $\forall\ \beta, \gamma \in \{S, L\}$.

5. Incremental Timing Refinement on Our Delay Model

STA provides vector-independent min-max timing ranges for rising and falling transitions on each line. It can be used as initial timing information for test generation since a test generator starts with all unknown values. But as more specific values are assigned during test generation process, the min-max ranges will become narrower due to (1) the increased specificity of the input vector pair, and (2) the logic and timing dependencies between lines ignored in STA appear as more input values are specified. The worst case corners obtained from STA may be invalid after some input values are specified. We have developed a timing mechanism called **incremental timing refinement (ITR)** for identifying new worst case corners and computing new A, T, and Q in these corners.

Incremental timing refinement uses the min-max timing ranges computed from static timing analysis as the initial timing information. At each test generation step, a more specific value is assigned to one or more circuit lines. The min-max ranges for timing parameters shrink due to re-calculation of arrival, transition, and required times. The shrinking of timing ranges helps timing oriented test generator eliminate choices.

5.1 Logic Value System

For timing simulations and test generations, two-pattern tests are needed to create transitions carrying timing information. In addition to the timing information, a two-frame value (v_1, v_2) is used to record the logic information for each line, The values in each frame could be 0, 1, or x, where **x** represents the unspecified value for a primary input, and the unknown value for any other line. As the value at a line is further specified, the forward and backward logic implications may refine the values at other lines. The required implication procedure can be obtained by extending a basic implication method ([21]) to two timeframes.

Among the nine logic values, {00, 01, 0x, 10, 11, 1x, x0, x1, xx}, for two-frame logic, 01 certainly carries a rising transition. 0x, x1, and xx potentially carry a rising transition. Other logic values certainly do not carry a rising transition.

After performing the logic simulation, we may find that a line may definitely not (or definitely) carry a rising (falling) transition. This fact may invalidate the worst case corner analysis in STA which assumes a rising transition is possible at this line. The essential information from logic values for ITR is whether a line carries a transition or not. According to the analysis for transitions on the nine logic values, we define the **status** of a transition *tr* on line Z, S^Z_{tr} as below:

$S^Z_{tr} = 1$, if line Z certainly carries a transition *tr*;

$= 0$, if line Z potentially carries a transition *tr*;

$= -1$, if line Z certainly does not carry a transition *tr*

S^Z_{tr} can be computed according to the logic value on Z, where $tr \in \{R, F\}$. When S^Z_{tr} is -1, none of the timing values about the transition *tr* at line Z is meaningful; each timing value may hence be left undefined. Verifying the status at a line before accessing this line's timing values will avoid these values from being used incorrectly. In the other two cases (S^Z_{tr} = 1 or 0), the timing fields are those in STA. STA is a special case of ITR where $S_{tr} = 0$ for every line. A method to calculate the timing values at each line is illustrated next.

5.2 Calculation of Arrival and Transition Times

Again a NAND gate with output Z and two inputs X and Y is used for illustrating ITR. An **optimization target** (**$OPT^{Z}_{tr,\ extreme}$**) is an *OPT* on line Z whose *extreme* value is desired for transition *tr*, where $OPT \in \{T, A\}$, $tr \in \{R, F\}$ and $extreme \in \{S, L\}$. Since S^{Z}_{tr} is only related to $S^{X}_{\overline{tr}}$ and $S^{Y}_{\overline{tr}}$ where $\overline{tr}$ is R(F) when *tr* is F(R), below this mapping is always assumed if we do not specify the transition direction.

To find the extreme value for an optimization target on Z, we need to decide (1) this extreme value will occur when more or fewer transitions on X and Y (at least one transition at an input is needed, for the output to have a transition), (2) given current logic values on X and Y (we lose some choices on X if $S^X = 1$ or -1, similarly for Y), we prefer to have transitions on X and Y or not, if $S^X = 0$ ($S^Y = 0$), and (3) for the inputs with transition, how do we pick their arrival times and transition times (minimal, maximal, or peak) to excite the extreme value on the optimization target.

For the extreme value of an optimization target, the line with potential transitions ($S^X = 0$) will be treated as either the transition occurs ($S^X = 1$) or it does not occur ($S^X = -1$), depending on the optimization target. That zero value S^X should be set to 1 or -1 depends on S^Y. The five rules on setting the zero value S^X for minimal arrival time at Z are shown below:

Case 1. $S^Y = -1$: set S^X to 1 for creating a transition at Z.

Case 2. $S^Y = 1$ and to-controlling transition occurs at Y: set S^X to 1 because additional input transition may speed up the output transition.

Case 3. $S^Y = 1$ and to-non-controlling transition occurs at Y: set S^X to -1 because additional input transition may slow down the output transition.

Case 4. $S^Y = 0$ and possible to-controlling transition at Y: set (S^X, S^Y) to (1, 1), because simultaneous switches in this direction speed up the output transition.

Case 5. $S^Y = 0$ and possible to-non-controlling transition at Y: try two possibilities that $(S^X, S^Y) = (1, -1)$ and $(-1, 1)$, because simultaneous switches are not desired but at least one input transition is required to create a transition at the output.

The setting of zero value S^X_{tr} and S^Y_{tr} for extreme values on the given optimization target is shown in Table 1 according to the five rules above. Only the non-trivial cases where at least one of S^X_R and S^Y_R is 0 are shown. The extreme values for the optimization targets are calculated below with all the zero-value S set as shown in Table 1.

original input states		Optimization target															
		$A^Z_{F,S}$		$A^Z_{F,L}$		$A^Z_{R,S}$		$A^Z_{R,L}$		$T^Z_{F,S}$		$T^Z_{F,L}$		$T^Z_{R,S}$		$T^Z_{R,L}$	
S^X	S^Y	S^X_R	S^Y_R	S^X_R	S^Y_R	S^X_F	S^Y_F	S^X_F	S^Y_F	S^X_F	S^Y_F	S^X_F	S^Y_F	S^X_F	S^Y_F	S^X_F	S^Y_F
-1	0	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1
0	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
0	0	-1	1	1	1	1	1	-1	1	1	1	-1	1	1	1	-1	1
		1	-1					1	-1			1	-1			1	-1
0	1	-1	1	1	1	1	1	-1	1	1	1	-1	1	1	1	-1	1
1	0	1	-1	1	1	1	1	1	-1	1	1	1	-1	1	1	1	-1

TABLE 1. The implied values of S for obtaining the extreme cases for optimization target.

5.2.1 for Output Falling Transition

For Smallest Output Arrival Time:

Case 1. $S^X_R = 0$ and $S^Y_R = 0$: Use the formulas for STA.

Case 2. $S^X_R = 1$ and $S^Y_R = 1$: Both inputs have rising transitions.

Case 3. $S^Y_R = -1$ and $S^X_R = 0$ or 1: Only rising transition from X may cause falling transition on Z.

Case 4. $S^X_R = 1$ and $S^Y_R = 0$: Extreme case occurs when only rising transition on X (same formulas as that in case 3).

Case 5. $S^X_R = -1$ and $S^Y_R = 0$ or 1: similar to case 3.

Case 6. $S^{Y}_{R} = 1$ and $S^{X}_{R} = 0$: same as case 5.

Because case 1 uses the formulas for STA, and cases 4, 5, 6 are same or similar to case 3, only case 2 and 3 are shown here. In the pin-to-pin delay model, to-non-controlling response is determined by last input to-non-controlling transition.

$$A^{Z}_{F,S} = \max\left(A^{X}_{R,S} + \min\left[d^{Z,X}_{F}(T^{X}_{R,S}),\ d^{Z,X}_{F}(T^{X}_{R,L}) \right],\right.$$

$$\left. A^{Y}_{R,S} + \min\left[d^{Z,Y}_{F}(T^{Y}_{R,S}),\ d^{Z,Y}_{F}(T^{Y}_{R,L}) \right] \right),$$

for $S^{Y}_{R} = 1$, or $S^{X}_{R} = 1$.

$$A^{Z}_{F,S} = A^{X}_{R,S} + \min\left[d^{Z,X}_{F}(T^{X}_{R,S}),\ d^{Z,X}_{F}(T^{X}_{R,L}) \right],$$

for $S^{Y}_{R} = -1$, or $S^{X}_{R} = 1$ and $S^{Y}_{R} = 0$.

For Largest Output Arrival Time:

Case 1. $S^{X}_{R} \neq -1$ and $S^{Y}_{R} \neq -1$: Use the formulas for STA.

Case 2. $S^{Y}_{R} = -1$: only rising transition from X may cause falling transition on Z.

Case 3. $S^{X}_{R} = -1$: similar to case 2.

$A^{Z}_{F,L} = A^{X}_{R,L} + d^{Z,X}_{F}(T^{X'}_{R})$,for $S^{Y}_{R} = -1$,

where $T^{X'}_{R}$ is defined in $A^{Z}_{F,L}$ computation for STA.

For Smallest and Largest Output Transition Time:

Both cases can be analyzed in the same manner as three cases for largest output arrival time.

$T^{Z}_{F,S} = t^{Z,X}_{F}(T^{X}_{R,S})$, for $S^{Y}_{R} = -1$, or $S^{X}_{R} = 1$ and $S^{Y}_{R} = 0$.

$T^{Z}_{F,L} = t^{Z,X}_{F}(T^{X}_{R,L})$,for $S^{Y}_{R} = -1$, or $S^{X}_{R} = 1$ and $S^{Y}_{R} = 0$.

5.2.2 for Output Rising Transition

For Smallest Output Arrival Time:

Case 1. $S^X_F \neq -1$ and $S^Y_F \neq -1$: Use the formulas for STA.

Case 2. $S^Y_F = -1$: only falling transition from X may cause rising transition on Z.

Case 3. $S^X_F = -1$: similar to case 2.

$A^Z_{R,S} = A^X_{F,S} + \min [d^{Z,X}_R(T^X_{F,S}) , d^{Z,X}_R(T^X_{F,L})]$, for $S^Y_F = -1$.

For Largest Output Arrival Time:

Case 1. $S^X_F = 0$ and $S^Y_F = 0$: Use the formulas for STA.

Case 2. $S^X_F = 1$ and $S^Y_F = 1$: Both inputs have falling transitions.

Case 3. $S^Y_F = -1$ and $S^X_F = 0$ or 1: Only the falling transition on X may cause a rising transition on Z.

Case 4. $S^X_F = 1$ and $S^Y_F = 0$: Same formulas as that in case 3.

Case 5. $S^X_F = -1$ and $S^Y_F = 0$ or 1: similar to case 3.

Case 6. $S^Y_F = 1$ and $S^X_F = 0$: same as case 5.

$A^Z_{R,L} = \min (A^X_{F,L} , A^Y_{F,L}) + d^Z_R(T^{X'}_F , T^{Y'}_F , A^Y_{F,L} - A^X_{F,L})$,

for $S^Y_F = S^X_F = 1$.

$A^Z_{R,L} = A^X_{F,L} + d^{Z,X}_R(T^{X'}_F)$,

for $S^Y_F = -1$ and $S^X_F = 0$ or 1, or $S^X_F = 1$ and $S^Y_F = 0$.

where $T^{X'}_F$ is as defined in $A^Z_{R,L}$ computation for STA.

When $S^Y_F = S^X_F = 1$, the extreme value occurs when both inputs transitions arrive as late as possible. The proper T^X and T^Y for maximizing the delay need to be explored. Here $T^{X'}_F$ and $T^{Y'}_F$ have three possible values as that for $A^Z_{R,L}$ in STA. But there $T^X_{F,max}$ maximizes $d^Z_R(T^X_F , T^Y_F , A^Y_{F,L} - A^X_{F,L})$ only for fixed T^Y_F and $\delta^{X,Y}$. We use numerical method to calculate $T^X{}_{F,\,max}$ & $T^Y{}_{F,\,max}$ iteratively until the pair ($T^X{}_{F,\,max}$, $T^Y{}_{F,\,max}$) that maximizes the delay function is found for fixed $\delta^{X,Y}$.

For Smallest Output Transition Time:

Both can be analyzed using the same three cases for largest output arrival time.

Case 1. $S^X{}_F \neq -1$ and $S^Y{}_F \neq -1$: Use the formulas for STA.

Case 2. $S^Y{}_F = -1$: only falling transition from X may cause rising transition on Z.

Case 3. $S^X{}_F = -1$: similar to case 2.

$T^Z{}_{R,\,S} = t^{Z,\,X}{}_R\,(T^X{}_{F,\,S})$, for $S^Y{}_F = -1$.

For Largest Output Transition Time:

Case 1. $S^X{}_F = 0$ and $S^Y{}_F = 0$: Use the formulas for STA.

Case 2. $S^X{}_F = 1$ and $S^Y{}_F = 1$: Both inputs have falling transitions.

Case 3. $S^Y{}_F = -1$ and $S^X{}_F = 0$ or 1: only falling transition from X may cause rising transition on Z.

Case 4. $S^X{}_F = 1$ and $S^Y{}_F = 0$: same formulas as that in case 3.

Case 5. $S^X{}_F = -1$ and $S^Y{}_F = 0$ or 1: similar to case 3.

Case 6. $S^Y{}_F = 1$ and $S^X{}_F = 0$: same as case 5.

When $S^X{}_F = 1$ & $S^Y{}_F = 1$, the output transition time is maximized when T^X and T^Y are maximized and the absolute value of $\delta^{X,Y}$ is maximized.

$T^Z{}_{R,\,L} = \max\,[t^Z{}_R(\,T^X{}_{F,L}\,,\,T^Y{}_{F,L}\,,\,A^Y{}_{F,S} - A^X{}_{F,L}),\,t^Z{}_R(\,T^X{}_{F,L}\,,\,T^Y{}_{F,L}\,,\,A^Y{}_{F,L} - A^X{}_{F,S})]$,

for $S^X{}_F = 1$ & $S^Y{}_F = 1$.

$T^Z{}_{R,\,L} = t^{Z,\,X}{}_R\,(T^X{}_{F,\,L})$, for $S^Y{}_F = -1$, or ($S^X{}_F = 1$ & $S^Y{}_F = 0$).

5.3 Calculation of Required Time:

For STA, minimal required time at input X = minimal required time at output Z - maximal delay from X to Z. Maximal required time is defined similarly. The required time for $Q^{X}_{R,S}$ and $Q^{X}_{R,L}$ are the same as that in STA because the same pin-to-pin delay from X to Z are used for computing these two Qs.

For the required time $Q^{X}_{F,S}$ and $Q^{X}_{F,L}$, the cases where $S^{X}_{R} = -1$ are not of interested since no required time is needed for X. Observe the other three cases for maximal rising delay used for $A^{Z}_{R,L}$, we can find

$Q^{X}_{F,S} = d^{Z}_{R}(T^{X'}_{F} , T^{Y'}_{F} , A^{Y}_{F,L} - A^{X}_{F,L})$,for $S^{Y}_{F} = S^{X}_{F} = 1$;

is the same as that in STA, else.

Here $T^{X'}_{F}$ and $T^{Y'}_{F}$ are defined in $A^{Z}_{R,L}$ for STA. $Q^{X}_{F,L}$ is found similarly.

$Q^{X}_{F,L}$ is the same as that in STA, for $S^{Y}_{F} \neq -1$;

$= Q^{X}_{R,L} - \min [d^{Z,X}_{R}(T^{X}_{F,S}) , d^{Z,X}_{R}(T^{X}_{F,L})]$, for $S^{Y}_{F} = -1$.

5.4 Considering gates with more than two inputs

For worst case delays and output transition times on a gate *Z* with inputs X_1 , X_2 , ..., X_n, there are three possible cases on setting S^{X_j} to 1 or -1, for $\forall$ j $\in$ {1, 2, ..., n} and S^{X_j}= 0. The three cases are (1) set all such S^{X_j} to 1, (2) set all of them to -1, and (3) set one of them to 1 and all others to -1. The worst case corners can be found and A, T, and Q can be calculated by directly extending the above techniques.

When the effect of input position is considered or transistors of different sizes are used in the parallel transistor, to-controlling delay should be redefined respect to dominating input. Here **dominating input,** $\mathbf{DI}^{Z}_{tr}$, of gate *Z* with inputs X_1 , X_2 , ... X_n , is the gate input X_j that the transition on X_j minimizes (maximizes) $(A^{X_j}_{tr1} + d^{Z,X_j}_{tr})$ for j $\in$ {1, 2, ... n} when *tr* is a to-controlling (to-non-controlling) response. Here tr1 equals tr for noninvertering gates, and $\overline{tr}$ for invertering gates. And the output arrival time becomes the sum of dominating input arrival time plus gate delay.

6. Experiments and Comparisons

6.1 Delay Model

The proposed delay model has been implemented and compared with HSPICE and with two inverter-collapsing methods Jun [17] and Nabavi [18]. The improved input mapping method for simultaneous switches at more than two inputs proposed in [19] is also integrated with Jun's approach. Empirical data are obtained from HSPICE simulation using SPICE LEVEL 3 model and 0.5 μm technology. NAND gates with minimum-size transistors are used for comparison. Each gate drives a minimum-size inverter as the load. To-controlling transitions are applied to some inputs of NAND gates. Non-controlling value is applied to remaining inputs.

Figure 14 shows the pin-to-pin delay at position 4 of a five-input NAND gate. Since current inverter-collapsing methods do not consider input position, the error rate may be high even for a single input transition. The results of Jun and Nabavi overlap because the same empirical inverter model is used. When the same transition is applied at the position 0 of a five-input NAND gate, all these approaches match HSPICE results.

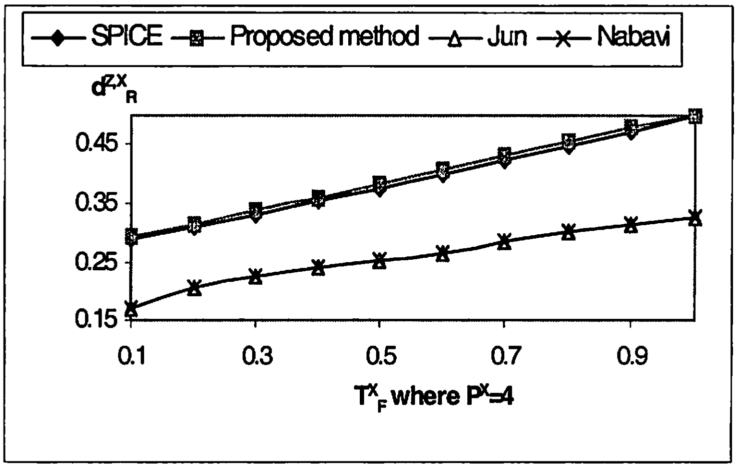


FIGURE 14. Single transition on position 4 of NAND5.

Figure 15 shows the result of simultaneous transitions at the NAND gate in Figure 1 when $\delta^{X,Y} = 0$ and $T^X_F = 0.5$ns. The proposed method and that due to Jun perform well but that due Nabavi performs well only when the transition times of the two inputs are close to each other. The reason is that this approach mainly considers the simultaneous transitions with the same start time, but the formulas obtained in that case does not extend to general cases.

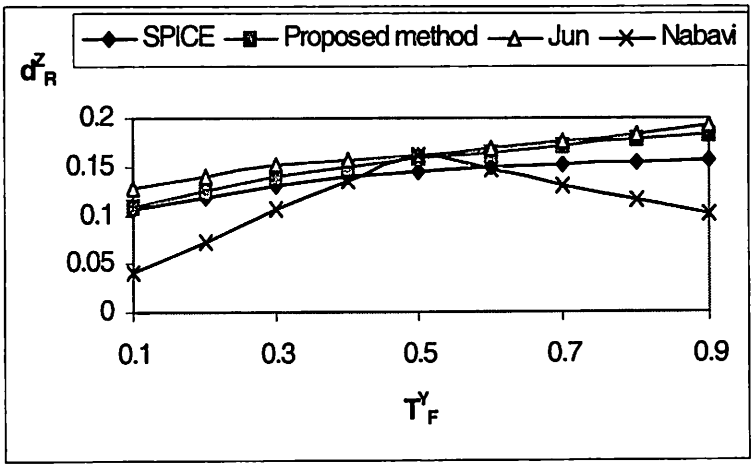


FIGURE 15. Simultaneous switch on NAND2 with single input transition time change.

For a two-input NAND gate with fixed T^X_F and T^Y_F , Figure 16 shows the delay as $\delta^{X,Y}$ changes. Our approach matches with HSPICE. Jun's approach fails to capture the delay for large skew. Nabavi's approach has significant error.

When identical transitions are applied to all inputs X, Y, and W of a three-input NAND gate, all the approaches under comparison perform well. When the transition times on the three inputs are different (Figure 17 where $T^X_F = 0.3$ns, $T^Y_F = 0.7$ns, and $\delta^{X,Y} = \delta^{X,W} = 0$), Nabavi's approach has significant error because of the reason stated earlier; the other two approaches preform reasonably well. Our approach performs well for three simultaneous transitions when one of the skew value is changed (Figure 18 where $T^X_F = 0.3$ns, $T^Y_F = 0.7$ns, $T^W_F = 0.2$ns, and $\delta^{X,Y} = 0$), while inaccuracies of the type discussed above are observed for the other two approaches.

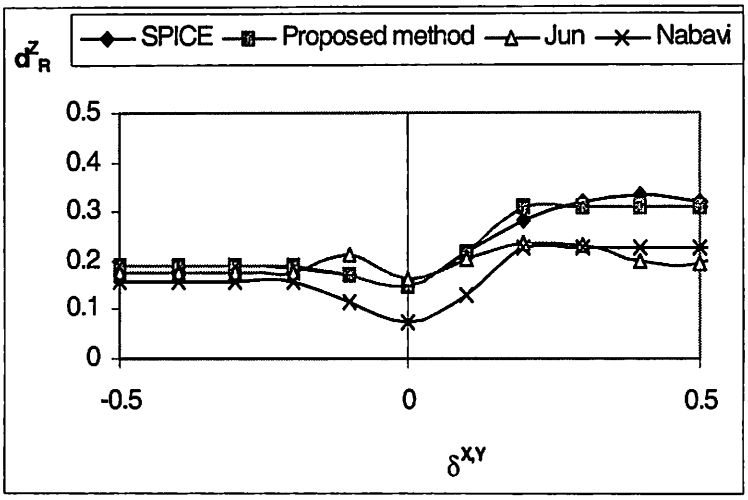


FIGURE 16. Vary $\delta^{X,Y}$ on simultaneous switch at NAND2.

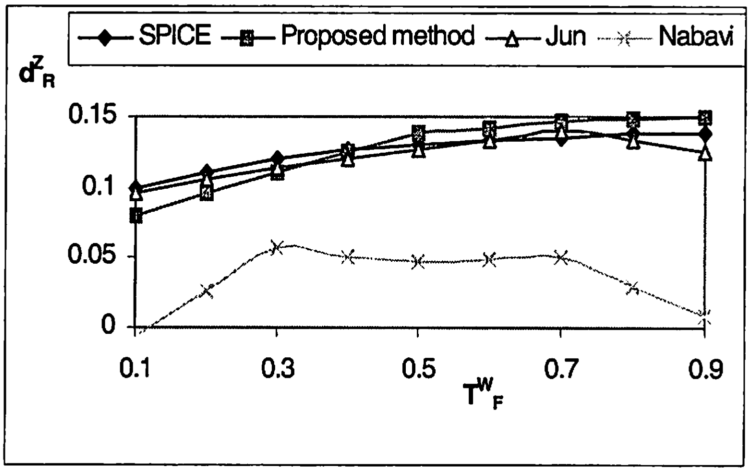


FIGURE 17. Vary $T^W{}_F$ in three simultaneous switches.

6.2 Incremental Timing Refinement

As some timing variables are not captured in the models of Jun and Nabavi, these methods work well only when some timing conditions are satisfied. In contrast, our approach works for more general cases. In addition to the improved accuracy, our model is also able to serve as the timing model for STA and ITR where the worst case corners need to be identified. These corners are hard to identify not only for equation solving models and table lookup models but also for some empirical models. For a model to adopt our method to find the worst case corners, a sufficient condition is that all timing functions of

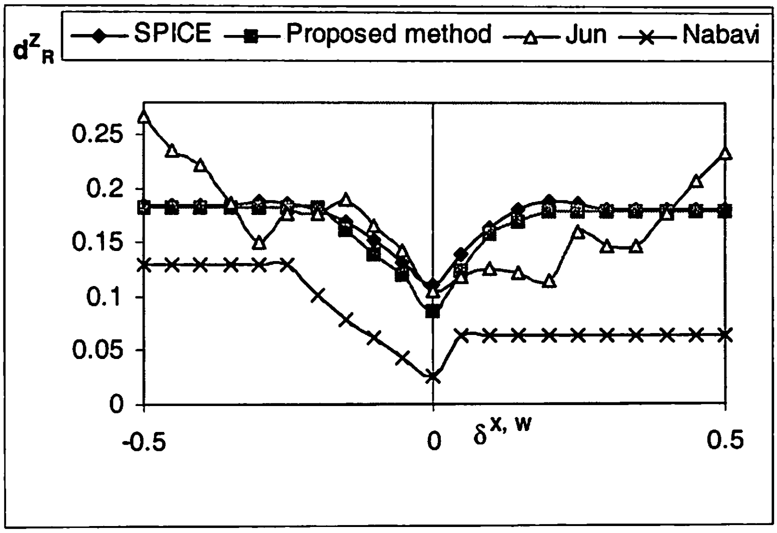


FIGURE 18. Three input switch, $\delta^{X,W}$ varied, and $\delta^{X,Y}$ constant.

this model are monotonic or bi-tonic respect to each input variable. Inheriting STA's ability to deal with unspecified value and ITA's ability to refine the timing, our model is able to deal with unspecified or partially specified vectors not handled by current timing simulators or delay calculators.

ITR is performed on benchmark C17 (Figure 19) and shown in Table 2. Here **V** means

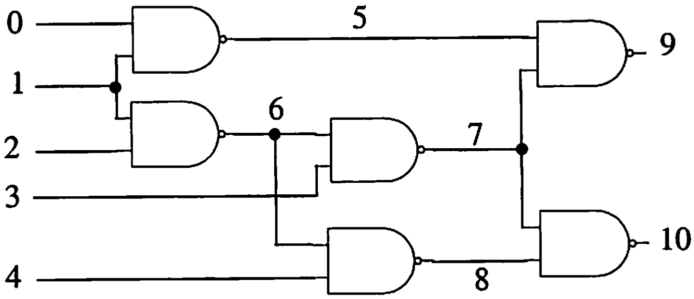


FIGURE 19. Benchmark circuit C17.

logic value. **FS** is the arrival time for earliest falling transition. **FL** is the arrival time for latest falling transition. **RS** is the arrival time for earliest rising transition. **RL** is the arrival time for latest rising transition. The time unit is 0.01ns. All inputs are initialized as unspecified. The timing information is the same as that in static timing analysis at this time. In each step, the value at one input becomes more specific and the timing parameters are updated. Timing windows keep shrinking when inputs become more specified. Logic value on a line may imply that no rising (falling) transition exists here. Then the timing

0	1	2	3	4	5	6					7					8					9					10				
V	V	V	V	V	V	V	FS	FL	RS	RL	V	FS	FL	RS	RL	V	FS	FL	RS	RL	V	FS	FL	RS	RL	V	FS	FL	RS	RL
xx	xx	xx	xx	xx	xx	xx	7	8	8	15	xx	7	23	9	26	xx	7	23	9	26	xx	16	33	17	38	xx	16	33	17	41
00	xx	xx	xx	xx	11	xx	7	8	8	15	xx	7	23	9	26	xx	7	23	9	26	xx	16	33	23	38	xx	16	33	17	41
00	11	xx	xx	xx	11	xx	7	7	13	13	xx	7	20	9	25	xx	7	20	9	25	xx	16	32	23	36	xx	16	33	17	38
00	11	10	xx	xx	11	01	-	-	13	13	1x	20	20	-	-	1x	20	20	-	-	0x	-	-	36	36	0x	-	-	30	38
00	11	10	01	xx	11	01	-	-	13	13	10	20	20	-	-	1x	20	20	-	-	01	-	-	36	36	01	-	-	30	38
00	11	10	01	x0	11	01	-	-	13	13	10	20	20	-	-	11	-	-	-	-	01	-	-	36	36	01	-	-	38	38

TABLE 2. A ITR example on C17.

information for this transition becomes meaningless and represented as '-' in this table. After the values of the first three inputs are specified in the table, the arrival time of rising transition on output 9 has shrunk from (17, 38) to (36, 36), a fixed value. It shows that partially specified vector may specify the exact timing at certain lines if the simulators can utilize the provided timing information. In fact, during each excitation, a test generator usually focuses on exciting the fault at one site, so many input values in the generated vector are left unspecified. To validate the generated tests, this simulator needs to confirm if the vevtor really satisfies the given timing requirements, without making any assumptions on the unspecified values.

ITR was performed on ISCAS85 circuits by randomly specifying values at inputs. In our current approach where pulses are ignores, timing ranges become points when all inputs are specified. How much the total timing ranges shrink as inputs specified is shown in Figure 20. It shows that the timing ranges of large circuits may shrink as fast as those for smaller circuits, in term of percentage of inputs specified. The total timing ranges of C7552 shrinks much faster than those for others circuits, because, in this circuit, many lines close to primary inputs have large fanouts.

Plan to show in Figure 21: Timing range approximation (ignoring logic/timing dependences) v.s. the real timing ranges (all possible fully specified vectors). **X axis**: real timing ranges/approximation timing range in % (for a gate). **Y axis**: populations of gates with the same X values. n curves - each curve is the statistical data for all test targets with p inputs unspecified, where $p \in \{m, m+1, ..., m+n-1\}$, the values of m and n are to be determined, where test target is defined below.

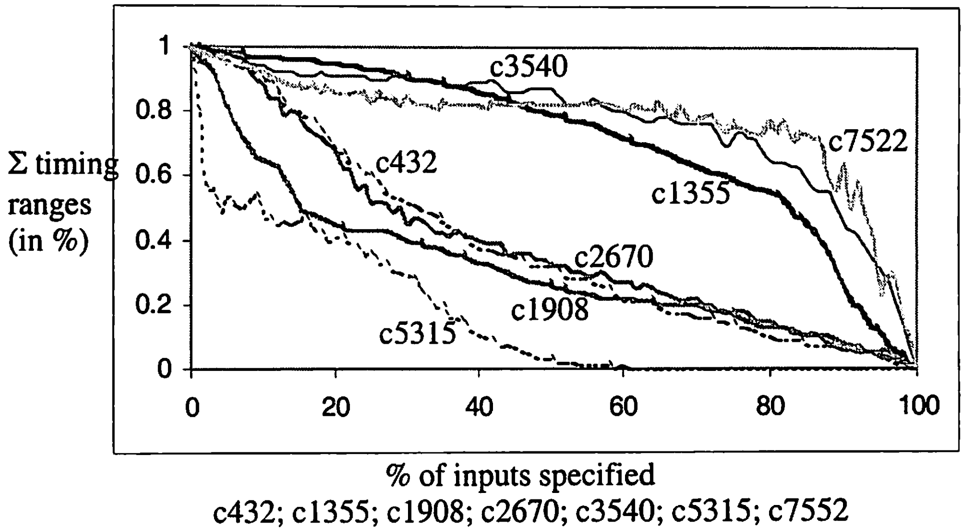


FIGURE 20. Total timing range shrink as more inputs are specified.

Possible test targets: (1) gates with cones of p inputs (reason: Do not need to assign values on any other inputs.), (2) primary outputs (reason: Timing ranges are most significant, so the difference may be more observable. Disadvantage: need to assign values to other inputs in the same cone randomly) (3) gates with fanout reconverges: more significant impacts on the difference of these two measured timing ranges.

Possible benchmark: (1) C432- smaller. (2) Circuits where timing ranges shrink fast (C432 or C1908) v.s. Circuits where timing ranges shrink slow (C1355): the reasons that timing ranges shrink fastrt is probably that the data dependence between inputs. If so, the difference between timing range approximation and the real timing ranges will be more significant.

7. Summary

We have developed a new delay model to capture the delay of simultaneous to-controlling transitions and input positions. By linearly approximating skew-delay relation on a two-input NAND gate and curve fitting on the empirical results, general forms of delay equations have been developed. The model has been extended for general cases.

The delay transfer functions for static timing analysis and incremental timing refinement have been developed based on our model. ITR can deal with min-max timing ranges and partially specified vectors. All worst case corners are identified to guarantee the correct propagation of min-max timing ranges in this delay model. A sufficient condition for adopting the ITR transfer functions into a delay model is found. The simulation results show that this model provides superior accuracy over other timing models for the same purpose. Ability of our model to refine the timing information as more values specified is verified.

[1] W. Y. Chen, "Test generation for crosstalk noise in VLSI circuits", Ph.D. dissertation, University of Southern California, EE-System Dept., 2000.

[2] R. B. Hitchcock, "Timing verification and timing analysis program" , Proc. of the 19th ACM / IEEE DAC 1982, pp. 594-604.

[3] L. C. Chen, S. K. Gupta, and M. A. Breuer. "High quality robust tests for path delay faults", Proceedings VLSI Test Symposium, pp. 88-93, April 1997.

[4] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Test generation in VLSI circuits for crosstalk noise" , Proc, Int'l Test Conference, pp. 641-650, 1998.

[5] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Test generation for crosstalk-induced delay in integrated circuits" , Proc, Int'l Test Conference, 1999.

[6] IEEE DASC standard delay format (SDF) - web page http://vhdl.org/vi/sdf/.

[7] C. Visweswariah and R. A. Rohrer, "Piecewise approximate circuit simulation" , IEEE Trans. Computer-Aided Design, vol. 10, pp. 861-870, July 1991.

[8] Y. H. Shih, Y. Leblebici, and S. M. Kang, "ILLIADS : A fast timing and reliability simulator for digital MOS circuits" , IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 12, pp. 1387-1402, Sept. 1993.

[9] L. W. Nagel, "SPICE2, A computer program to simulate semiconductor circuits" , Memo UCB / ERL M520, Univ. Cal., Berkeley, May 1975.

[10] J. Rubenstein, P. Penfield, Jr., and M. A. Horowitz, "Signal delay in RC networks" , IEEE Transactions on CAD, vol. CAD-2, july 1983, pp.202-211.

[11] L. Bisdounis, S. Nikolaidis, and O. Koufopavlou, "Analytical Transient Response and Propagation Delay Evaluation of the CMOS Inverter for Short-Channel Devices", IEEE J. Solid-State Circuit, Vol.33, pp.302-306, Feb. 1998.

[12] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas", IEEE J. Solid-State Circuit, Vol.25, pp.584-594, Apr. 1990.

[13] V. B. Rao, T. N. Trick, and I. N. Hajj, "A table-driven delay-operator approach to timing simulation of MOS VLSI circuits" , Proc. 1983 IEEE ICCD. Nov. 1983, pp.445-448.

[14] F. C. Chang, C. F. Chen, and P. Subramaniam, "An accurate and efficient gate level delay calculator for MOS circuits" , in 25th ACM / IEEE Design Automation Conference, 1988.

[15] D. Overhauser and I. Hajj, "A tabular macromodeling approach to fast timing simulation including parasitics" , Proc. IEEE ICCAD, pp. 70-73 1988.

[16] V. Chandramouli and K. A. Sakallah, "Modeling the effects of temporal proximity of input transitions on gate propagation delay and transition time" , Proc. of the 32th Design Automation Conference 1996, pp. 617-622.

[17] 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. Computer-Aided Design, vol. 8, pp. 1027-1032, Sept. 1989.

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

[19] A.Chatzigeorgiou, S. Nikolaidis, and I. Tsoukalas, "A modeling technique for CMOS gates" , IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 18, pp. 557-575, May 1999.

[20] "Incremental timing refinement on a min-max delay model", Computer Engineer technical report No. 00-08, Electrical Engineer - System Dept., University of Southern California, April 2000.

[21] M. Abramovici, M. Breuer, and A. Friedman, "Digital Systems Testing and Testable Design", Computer Science Press, 1990.