

Event Propagation Conditions in Timing Analysis

by

Hakan Yalcin and John P. Hayes

CSE-TR-267-95

November, 1995



THE UNIVERSITY OF MICHIGAN

**Computer Science and Engineering Division
Department of Electrical Engineering and Computer Science
Ann Arbor, Michigan 48109-2122
USA**



Event Propagation Conditions in Timing Analysis*

Hakan Yalcin and John P. Hayes

Advanced Computer Architecture Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, Michigan 48109-2122
Email: {hakan, jhayes}@eecs.umich.edu

Abstract

We present a systematic study of signal propagation conditions (PCs) starting from a general waveform model. We develop a number of specific waveform models based on how closely they match actual signal behavior, and show that they form a well-defined hierarchy. For each model, we derive PCs based on fundamental cause-and-effect behavior. We then construct a lattice of all known PCs that reveals the relationships among them. This lattice also enables us to derive some new and potentially useful PCs. Experimental results are presented to evaluate the accuracy of the proposed PCs.

1 Introduction

Various propagation conditions (PCs) or sensitization criteria have been proposed in the literature to determine whether a signal propagation path is sensitized. A key aspect of PCs is the assumed waveform model, which specifies when and how signals actually change. While more detailed waveform models capture the actual signal behavior more accurately, their algorithmic implementations are slower. Most existing methods assume a simple waveform model because the delay computation problem is hard even with simple models. Floating mode [4], for example, is a waveform model where only the latest event on every circuit node, namely the node becoming stable, is considered. The value of the node until it stabilizes is assumed to be unknown. Transition mode [6], on the other hand, considers all possible events on internal nodes, while the primary inputs are restricted to a single event. Another important aspect of PCs is the role of causality. Event propagation is causal in that an event at the output of a circuit module occurs as a result of an input event.

Static sensitization [1] and the Brand-Iyengar condition [2] were the earliest PCs proposed in the literature. They were followed by many others including viability [10], floating mode sensitization [4], and the loose criterion [4,5]. However, most studies of these PCs do not consider the underlying waveform model or the causality principle explicitly. The analysis tends to be ad hoc or algorithm driven, and does not reflect how events actually propagate. Not surprisingly, for example, it was found that static sensitiza-

* This work was supported by the Semiconductor Research Corporation under Contract #93-DJ-338.

tion can underestimate the circuit delay [4,10]. Although the relationships between some of the PCs have been established [4,5,11,15], they have not been studied in a uniform way, and their overall relationships are far from clear.

In this paper, we attempt to derive propagation conditions in a systematic way starting from a general waveform model. We develop a number of specific models based on how closely they match actual signal behavior, and show that they form a well-defined hierarchy. For each waveform model, we derive propagation conditions based on causality. We then construct a lattice of PCs that reveals the relationships among them. This lattice also enables us to derive new and potentially useful PCs. Finally, we present experimental results to evaluate the accuracy of the proposed PCs.

2 Calculation of Propagation Conditions

A combinational logic circuit is composed of modules (gates, multiplexers, decoders, etc.) which are assumed to have known internal delays. The modules are linked by delay-free interconnections which, along with the circuit's primary input-output terminals, define the circuit's signal nodes. Given an input stimulus to the circuit, the nodes of the circuit undergo some changes (events). The entire set of events occurring at a circuit node constitutes its waveform. Figure 1 shows some waveform examples. Events at the primary inputs propagate through the circuit, are delayed by the modules, and eventually reach the primary outputs. Depending on circuit structure, events may propagate through different paths, and hence may experience different delays. Some events may be filtered out because of other events. The delays of the circuit are determined by the events propagating through it. Therefore, in order to find the circuit delays accurately, it is important to know how the events propagate.

The condition (predicate) under which an event propagates from an input x of a module to an output z is called the propagation condition (PC) and denoted by ψ_{xz} in this paper. Consider the AND gate shown in Fig. 1. The PC ψ_{xz} for the event on input x occurring at time t_x^3 is $(t_x^3 \geq t_y^2) \vee (t_x^3 \leq t_y^1)$, since input y is required to have a non-controlling value 1. Clearly, the entire waveform of y must be known in order to find whether the event on x can propagate. We call the logic-level behavior that considers each signal's entire waveform the *W0 waveform model* or the *exact model*. Timing analysis using the W0 waveform model is complicated for various reasons. First, since arbitrary number of events can occur on circuit nodes, the storage of these events can be a problem. Devadas et al. [8] give an example where the number events in

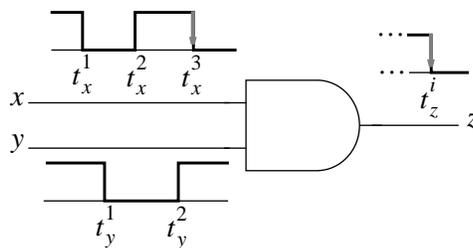


Figure 1: Event propagation through a 2-input AND gate.

the circuit is exponential in the circuit size. Second, PC calculation becomes difficult due to the potentially high number of conditions relating event times. Thus, simulation-like methods such as that of [6,8] must often be employed in practice. Complex delay models such as the min-max delay model further complicate the analysis.

Several types of approximations can be used to simplify the timing analysis problem:

- Restricting the waveform model. A subset of all possible events is considered. For example, floating mode [4] considers only the last event on every circuit node.
- Restricting the delay model. For example, if a module has many input-output path delays, one can consider only the maximum delay.
- Simplifying the calculation of PCs. An example is the “conditionless” case where all events are assumed to propagate; this is classical topological analysis.

The above approximation methods are not independent, however. For instance, if the waveform model is restricted, PC calculation will be restricted as well.

We note that the circuit delay obtained by approximate methods is an *estimate*, as opposed to the exact value, because of the information loss during approximation. In timing analysis, an estimate of the circuit delay greater than the exact one (an overestimate) is generally acceptable. However, an estimate less than the exact one (an underestimate) is unacceptable, because a clock period based on an underestimated delay can lead to incorrect circuit operation. An underestimated value is often referred to as “incorrect” in the literature. Overestimation, on the other hand, is “safe” and only results in a circuit operating more slowly than necessary. Therefore, while a good estimate of the delay must not be incorrect, it should be as close to the exact value as possible.

In the process of deriving approximate waveform models and their associated PCs, we will make use of a special “smoothing” operator [10]. Let f be a function of variables x_1, x_2, \dots, x_n . The *smoothing operator* $S_{x_i}f$ is defined as

$$S_{x_i}f = f_{x_i} + f_{\bar{x}_i}$$

where $f_{x_i} = f(x_i = 1)$ and $f_{\bar{x}_i} = f(x_i = 0)$. The smoothing operator can be extended to multiple variables. Let $U = \{x_{i_1}, x_{i_2}, \dots, x_{i_k}\}$ be a subset of x_1, x_2, \dots, x_n . Then, $S_{x_{i_1}}S_{x_{i_2}}\dots S_{x_{i_k}}f = S_{x_{i_1}\dots x_{i_k}}f = S_Uf$. The order in which the operator is applied to the variables of U is not important since $S_{x_i}S_{x_j}f = S_{x_j}S_{x_i}f$. The smoothing operator captures pessimism in the following way. The function S_Uf is true if the original function f is true for any combination of the variables in U . We make use of the following property from [11] to relate different PCs:

$$S_Uf \supseteq f \tag{1}$$

3 Approximate Waveform Models

Our first approximation to the W0 model of timing is to restrict signal waveforms to their first and last events. The remaining events occurring between these two are ignored, and it is assumed that any value (0,

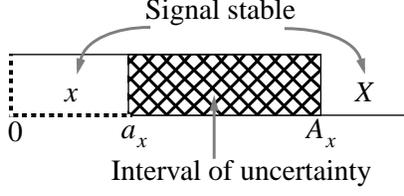


Figure 2: Waveform for a node x under the W1 model.

1, X , etc.) can appear between the first and last event times. We call this waveform model *W1* or the *first-and-last-event* (FALE) model. Figure 2 depicts a typical waveform under the W1 model. The initial and the final stable values of a signal for node x are represented by x and X , respectively. The times of the earliest and the latest events reaching node x are, respectively, a_x and A_x . This waveform model is also adopted by [14] and the compiled-code simulator Ravel [13].

To handle constant signals (no events), we associate with every signal x a special predicate C_x that indicates whether it is changing or constant: $C_x = 1$ (true) if x is changing, and $C_x = 0$ (false) otherwise. Clearly, if the initial and final stable values are different, C_x is necessarily 1. We make the assumption that if $C_x = 0$, then $a_x = \infty$ and $A_x = -\infty$. Thus, we have $C_x = a_x \leq A_x$.

Calculating PCs for the W1 model is far simpler than for the W0 model described above. However, in order not to underestimate the circuit delay, pessimistic assumptions must be made regarding the interval of uncertainty, which is shaded in Fig. 2. In particular, we have to assume that any event whose propagation depends on the values of the signal during this interval does propagate. This assumption can result in an overestimate of the circuit delay since some events may actually be blocked by certain values of the signal.

In the case of restricted waveform models, one can enumerate all possible input waveforms to calculate PCs. We illustrate this for a 2-input AND gate with inputs x and y , output z and zero delay. (Other gate types are analyzed similarly. Also, a non-zero delay just shifts the output waveform.) Let ψ_{xz}^{W1} represent the PC for the last event on input x to reach the output z , i.e., the sensitization condition for the path from x to z . We introduce three useful predicates to relate the event times of x and y :

$$E_x = A_x \leq A_y \text{ (} x \text{ stabilizes earlier than } y\text{),}$$

$$L_x = A_y \leq A_x \text{ (} x \text{ stabilizes later than } y\text{),}$$

$$V_x = a_y \leq A_x \text{ (} x \text{ stabilizes after } y \text{ starts to destabilize),}$$

Figure 3 shows ψ_{xz}^{W1} in the form of a truth table for all combinations of x , X , y , and Y , where juxtaposition represents the logical AND operation. The input waveforms for two entries of the table are shown in Fig. 3. For the input combination $x = 0, X = 0, y = 0, Y = 0$, ψ_{xz}^{W1} is $E_x V_x C_x$, that is, the two intervals must overlap, input x must not be constant 0 and must stabilize earlier than y . For the input combination $x = 0, X = 1, y = 1, Y = 1$, ψ_{xz}^{W1} is L_x , that is, input x must stabilize after y .

The PC ψ_{yz}^{W1} for path $y \rightarrow z$ is derived similarly. The latest event time A_z^{W1} for the output z , which is

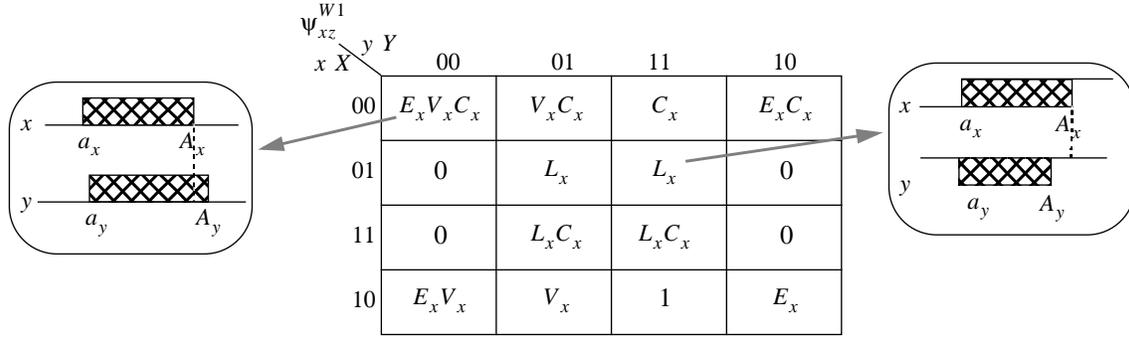


Figure 3: Truth table for the PC ψ_{xz}^{W1} for path $x \rightarrow z$ in a 2-input AND gate.

the delay up to z , can be derived from the PCs ψ_{xz}^{W1} and ψ_{yz}^{W1} as follows.

$$A_z^{W1} = \max(\psi_{xz}^{W1} \bullet A_x, \psi_{yz}^{W1} \bullet A_y) \quad (2)$$

where the AND-like operator “ \bullet ” is defined as

$$\psi_{xz}^{W1} \bullet A_x = \begin{cases} -\infty & \text{if } \psi_{xz}^{W1} = 0 \\ A_x & \text{if } \psi_{xz}^{W1} = 1 \end{cases}$$

Equation (2) follows from the fact that the event on the sensitized input(s) propagates to the output and determines the delay. It is possible that more than one input is sensitized, in which case ψ_{xz}^{W1} and ψ_{yz}^{W1} are both 1. The max operator takes care of this situation by choosing the latest event.

Although the W1 model is significantly simpler than the W0 model, it still may not be simple enough for delay calculations in large circuits. The PC ψ_{xz}^{W1} shown in Fig. 3, for example, is computationally complex. In order to process circuits with thousands of gates, we need to further simplify this model. We next consider two types of approximation, both of which arrive at the same waveform model:

- Ignoring the initial values,
- Ignoring all the events but the last one.

The waveform model resulting from these simplifications is called the *W2 model* and is depicted in Fig. 4. It is known in the literature as *floating mode*, and was introduced by Chen and Du [4]. In this paper, we consider ignoring the initial values, and derive PCs accordingly. It can be easily shown that the other approximation method yields the same PCs and latest event time (delay).

We obtain the PC ψ_{xz}^{W2} for the W2 model from the PC ψ_{xz}^{W1} for the W1 model by smoothing out the

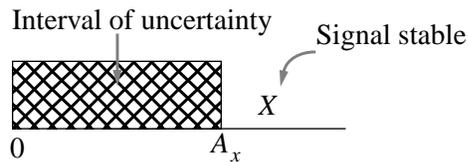


Figure 4: Waveform for a node x under the W2 (floating-mode) model.

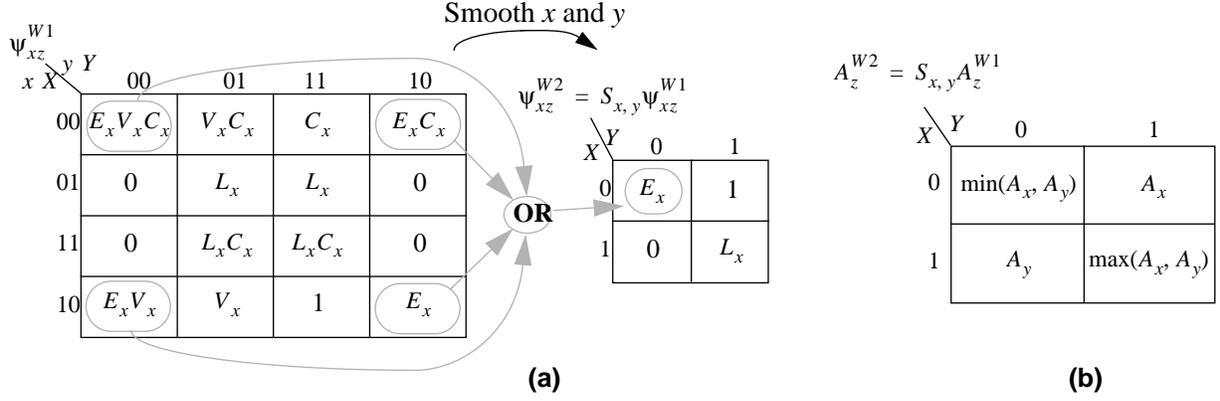


Figure 5: (a) The PC ψ_{xz}^{W2} ; (b) the latest event time A_z^{W2} under the W2 model.

initial values a_x and a_y . This is shown in Fig. 5(a). The new PC ψ_{xz}^{W2} exactly matches the conditions for floating mode given in [4], thus confirming our analysis. The same operation can also be applied to the latest event time (delay) A_z^{W1} to derive $A_z^{W2} = S_{x,y} A_z^{W1}$, the latest event time for the W2 model. Figure 5(b) shows the truth table for A_z^{W2} .

Next, we go one step further and smooth out the latest event times A_x and A_y . Hence, we are left only with the final (stable) values X and Y . We arrive at a model which is “static” in the sense that all dynamic signal behavior is lost. We call this waveform model W3. Although the notion of an event is absent from the W3 model, we treat sensitized paths (with respect to this model) as having abstract events whose occurrence times are the length of the paths.

The PC ψ_{xz}^{W3} for the W3 model is derived by smoothing A_x and A_y from ψ_{xz}^{W2} . The terms involving A_x and A_y are E_x and L_x . So,

$$S_{A_x, A_y} E_x = S_{A_x, A_y} (A_x \leq A_y) = \mathbf{1}, \text{ and } S_{A_x, A_y} L_x = S_{A_x, A_y} (A_y \leq A_x) = \mathbf{1}.$$

Substituting these values into ψ_{xz}^{W2} , we obtain the PC $\psi_{xz}^{W3} = S_{A_x, A_y} \psi_{xz}^{W2}$ shown in Fig. 6. It is interesting to note that the PC ψ_{xz}^{W3} is *not* the usual static sensitization condition, which is based on the D-algorithm used in test generation. It is known in the literature as *static co-sensitization* and was introduced by Devadas et al. [7]. Also shown in the figure are ψ_{yz}^{W3} and the latest (abstract) event time A_z^{W3} derived using Equation (2). Here, A_x and A_y are the maximum length of the sensitized paths ending at inputs x and y ,

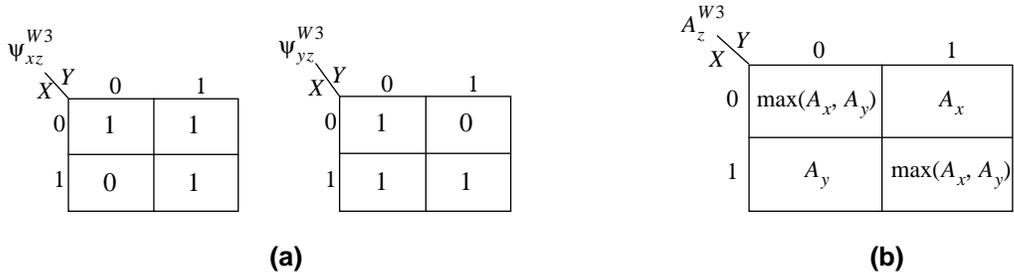


Figure 6: (a) The PCs ψ_{xz}^{W3} and ψ_{yz}^{W3} ; (b) the latest event time A_z^{W3} under the W3 model.

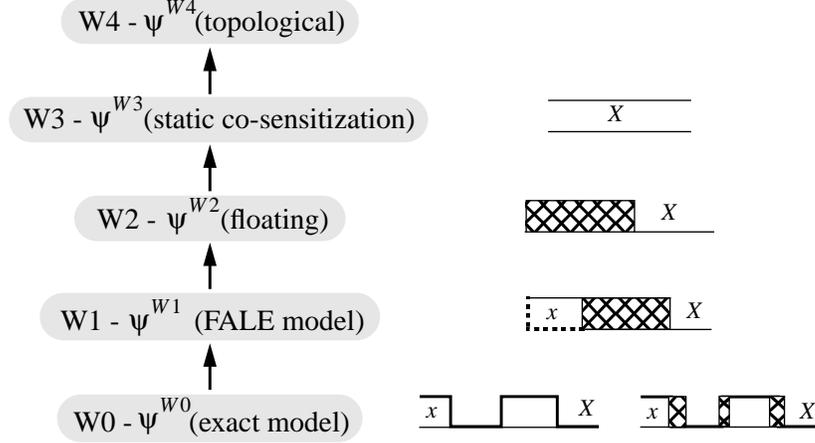


Figure 7: Summary of waveform models and their PCs.

respectively. Thus, A_z^{W3} is the maximum sensitized path length up to the output of the AND gate.

The last simplification is to smooth out the remaining variables, which are the final values X and Y . We call the resulting waveform model $W4$. The PC ψ_{xz}^{W4} under the $W4$ model is equal to $\mathbf{1}$, since $\psi_{xz}^{W4} = S_{X,Y}\psi_{xz}^{W3}(X, Y) = \mathbf{1}$, which means that every event propagates without any condition. The latest event time A_z^{W4} is $\max(A_x, A_y)$, i.e., the longest topological path delay up to the output of the AND gate. Thus, delay calculation for this model is equivalent to topological delay analysis.

A summary of the waveform models introduced so far and their PCs is shown in Fig. 7. Model complexity decreases as one moves from the $W0$ model to $W4$. The pessimism of the PCs increases in the same direction, since approximation implies pessimism. The *smoothing relation* between these waveform models as well as between PCs is denoted by \rightarrow . For example, denoting the PCs corresponding to two waveform models P and Q by ψ^P and ψ^Q , respectively, if $P \rightarrow Q$ ($\psi^P \rightarrow \psi^Q$), then model Q (PC ψ^Q) is obtained by smoothing out from model P (PC ψ^P) some of its variables.

The following two results are the direct consequences of the foregoing analysis.

Theorem 1: Let P and Q be two waveform models such that $P \rightarrow Q$. An event that propagates under P also propagates in under Q .

Theorem 2: Let P and Q be two waveform models such that $P \rightarrow Q$. Let the circuit delay under P be d^P and that under Q be d^Q . Then $d^Q \geq d^P$.

From Theorem 2, we have $d^{W0} \leq d^{W1} \leq d^{W2} \leq d^{W3} \leq d^{W4}$. Therefore, delay computation under models $W1$ through $W4$ is safe. Delay estimates get looser as one moves from $W0$ to $W4$.

4 Lattice of Propagation Conditions

The PCs in Fig. 7 can be augmented with others proposed in the literature to illustrate the relationships among the known PCs. We must first express all the PCs in terms of our notation. For brevity, we

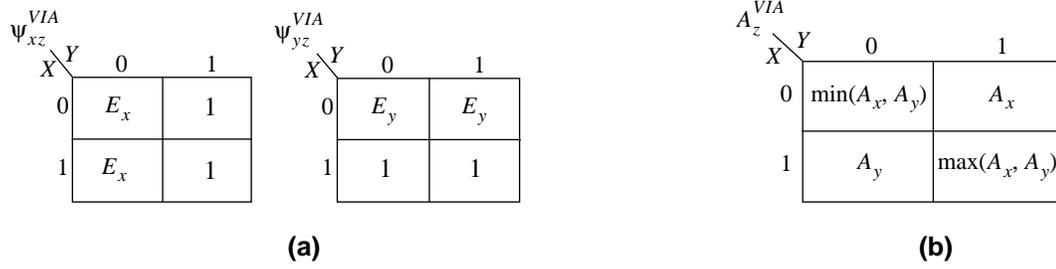


Figure 8: The viability conditions and the corresponding output delay.

only consider here viability [10], static sensitization [1] and the loose criterion [4, 5]. Other proposed PCs include the PC proposed by Perremans, Claesen and DeMan [12], the VIPER condition [3], the Brand-Iyengar condition [2], and the Du-Yen-Ghanta condition [9], which can all be treated similarly.

A widely-studied PC is *viability* [10, 11] which assumes the W2 waveform model. Consider a 2-input AND gate with inputs x and y and output z . Under viability, an event on input x propagates to the output z if

- The stable value Y (latest event time) of input y is non-controlling, or
- The stable value Y is controlling, but the event on input x is earlier than that on input y .

Denoting the PC for path $x \rightarrow z$ by ψ_{xz}^{VIA} , the above conditions translate into $\psi_{xz}^{VIA} = E_x + Y$ for a 2-input AND gate. Figure 8 shows the PCs ψ_{xz}^{VIA} and ψ_{yz}^{VIA} and the latest event time (delay) A_z^{VIA} . Note that A_z^{VIA} is the same as A_z^{W2} , the delay under the W2 model (floating mode), which confirms previous results [4,5].

Similarly, the PC ψ_{xz}^{LO} for the loose criterion [4, 5] can be shown to be $\bar{X}E_x + Y$. Further, as in the case of viability, the latest event time (delay) A_z^{LO} for the loose criterion is the same as that under the W2 model. We also note that the condition used by VIPER [3] is equivalent to the loose criterion.

Like our W3 model, static sensitization only deals with final stable values. It has been shown that static sensitization can underestimate [10] as well as overestimate [6,15]. We will also show now that this is true. The PCs for a 2-input AND gate under static sensitization are $\psi_{xz}^{STA} = Y$, $\psi_{yz}^{STA} = X$, which are shown in Fig. 9(a). As Fig. 9(b) illustrates, the delay for the input combination $XY = 00$ is undefined and is represented by $-\infty$. This is the source of the underestimation problem with static sensitization.

All the PCs $\psi_{xz}^{W_i}$ for path $x \rightarrow z$ of a 2-input AND gate discussed so far are summarized in Fig. 10 in the form of a lattice. We redefine \rightarrow to represent the *covering* relation, that is, $\psi^P \rightarrow \psi^Q$ now means $\psi^P \subseteq \psi^Q$, which is looser than the previous definition based on smoothing. The extreme elements in the

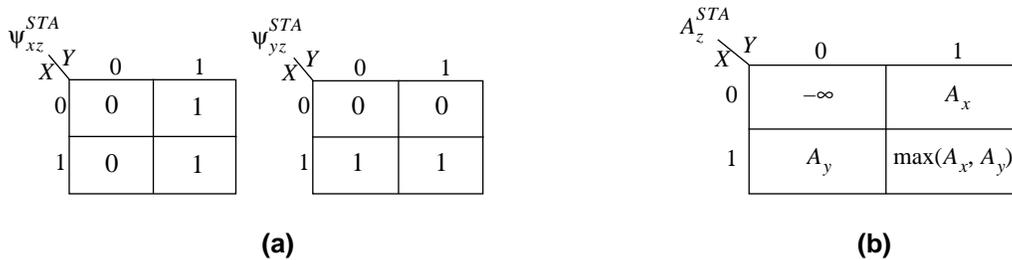


Figure 9: The static sensitization conditions and the corresponding output delay.

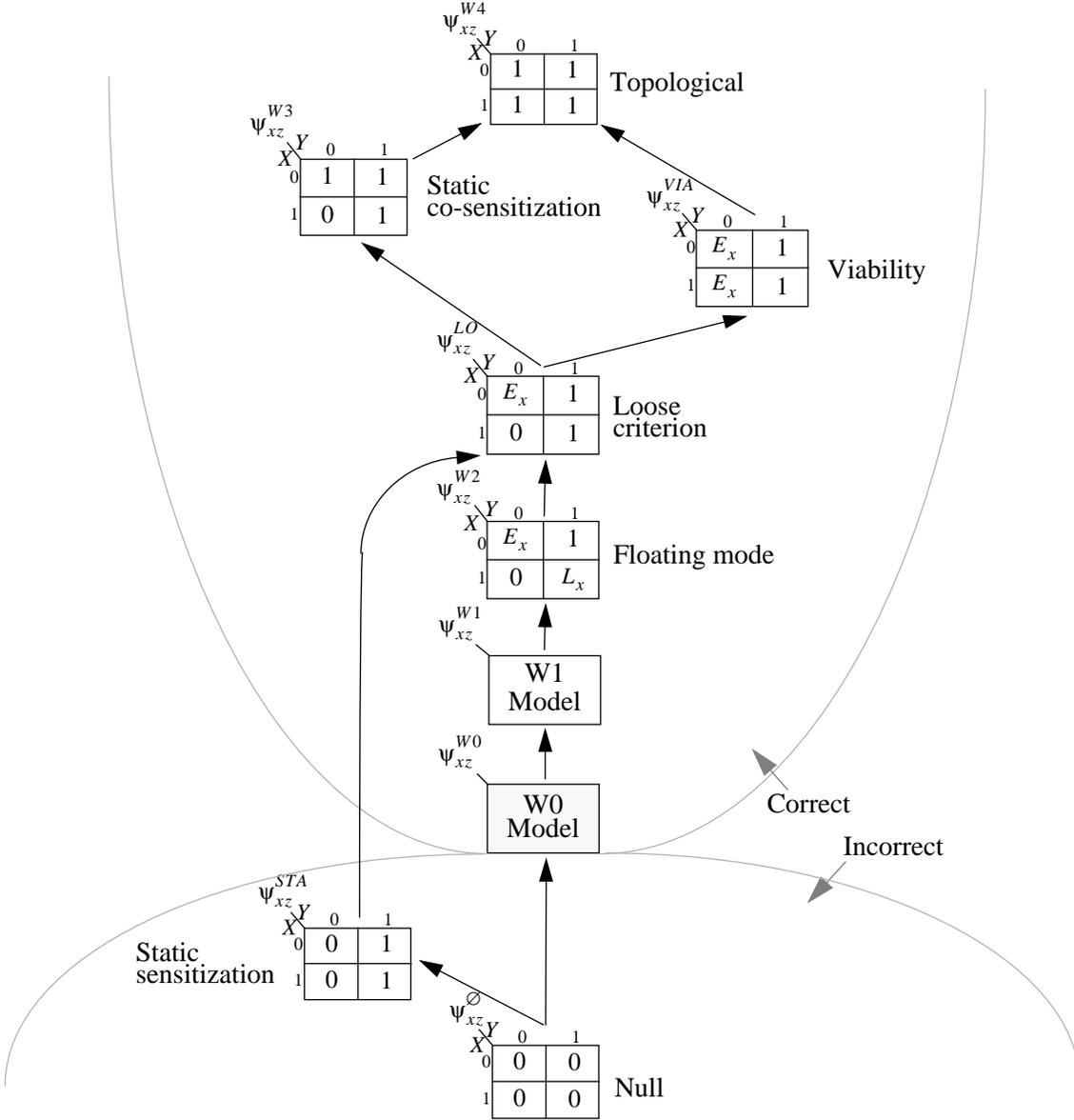


Figure 10: The lattice of PCs for a 2-input AND gate.

lattice are ψ_{xz}^{W4} (the topological PC), which propagates every event, and the *null PC* $\psi^{\emptyset} = \mathbf{0}$ which does not allow any event to propagate. The PC ψ^{W0} is the ideal one which gives the exact circuit delay d^{W0} . Both Theorems 1 and 2 hold for the covering relation \rightarrow also since their basis Equation (1) is satisfied. Thus, if $\psi^P \rightarrow \psi^Q$ for two PCs P and Q with corresponding circuit delays d^P and d^Q , respectively, then we have $d^P \leq d^Q$.

The entire set of PCs appearing Fig. 10 can be divided into two groups:

- Those above ψ_{xz}^{W0} in the lattice, that is, any PC Q for which $\psi^{W0} \rightarrow \psi^Q$, and hence $d^Q \geq d^{W0}$. These PCs are correct according to the special definition of correctness given earlier.
- Those below ψ_{xz}^{W0} . Obviously, for such a PC Q , $\psi^{W0} \rightarrow \psi^Q$ does not hold and $d^Q < d^{W0}$ is possible. These PCs are thus incorrect.

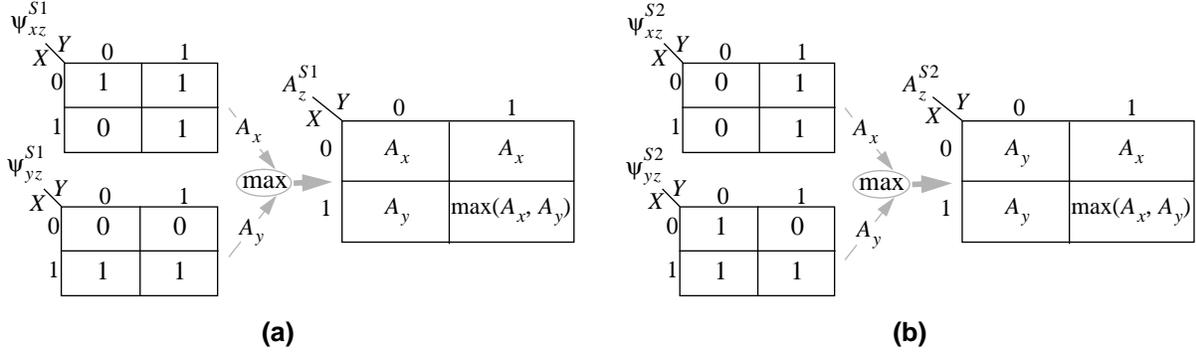


Figure 11: The conditions and the output delay for the new PCs: (a) S1, (b) S2.

The lattice is quite useful in evaluating the correctness and tightness of the PCs. For example, it is immediate that static sensitization can underestimate the circuit delay. While being correct, the closer to Ψ^{W0} (the PC for the W0 waveform model), the tighter the PC. All the relationships among the PCs that can be established from the lattice of Fig. 10 confirm previously published results [5, 11, 15]. Additionally, we have the following new result.

Theorem 3: When static sensitization does not underestimate, its estimate is equal to or better than that of the W2 model (floating mode).

With the help of the preceding analysis, one can find new and potentially useful propagation PCs. Of particular interest are those that only deal with final stable values, like static sensitization and W3 (static co-sensitization). We have come up with two such PCs that we call S1 and S2; they are shown in Fig. 11 along with their corresponding output delays computed according to Equation (2). As the figure reveals, S1 and S2 are a combination of static sensitization and W3 (static co-sensitization), and hence blend the safety of W3 and the tightness of static sensitization. While S1 imposes static sensitization on input y, S2 imposes static sensitization on input x. Both S1 and S2 are correct in that they never underestimate the circuit delay. Their tightness is between those of W3 and W2.

While S1 and S2 can be used individually, we also propose the following additional PC called *safe static* that makes use of both. It is defined as follows for any 2-input gate:

Safe static propagation condition: If $\Delta_x < \Delta_y$, then use S1, otherwise use S2

where Δ_x and Δ_y are the longest topological delays to inputs x and y, respectively. We note that safe static is different from the PC used in [16], which is equivalent to floating mode. The idea behind safe static is to impose static sensitization, which has tighter conditions, on the input whose topological delay is longer. This reduces the probability of the longer path being reported true when it is actually false. In this respect, safe static is similar to the Du-Yen-Ghanta [9] condition, which improves on the Brand-Iyengar [2] condition with the help of topological delays [11].

5 Experimental Results

To evaluate the tightness of the proposed PCs S1, S2 and safe static, we have performed experiments with the timing analysis program CAT described in [17]. CAT is a symbolic timing analyzer and can compute a circuit’s delays and associated conditions under any PC. The delay estimates of S1, S2 and safe static along with those of W4 (topological), W3 (static co-sensitization), W2 (floating), and static sensitization are shown in Table 1 for the ISCAS-85 benchmark circuits, carry-skip adders, and some examples from [7].

As mentioned before, the estimates of S1, S2 and safe static are between those of W2 and W3. For the ISCAS-85 benchmark circuits, S1 and S2 yield the same delay values as W2 except for c1908, where their estimate is equal to the longest topological delay. For carry-skip adders, S1 and S2 report the longest topological path delay. The delay estimates of static sensitization for cla.16 and tau92ex2 are less than those of W2, which indicates the possibility of underestimation for static sensitization. The estimates of S1 and S2 are safe, as shown in the table. The estimates of safe static are very good; they are the same as those of W2 for all the examples except two cases, where safe static overestimates by only 1 (tau92ex1) and 2 (tau92ex2). These results show that safe static is quite tight, especially considering the fact that it ignores dynamic signal behavior.

Finally, we make the following observations regarding the computation times. As one moves from W4 to W2, the computation times increase, as expected. Under a specific waveform model, different PCs can significantly change computation times, depending on the method and implementation. In our case,

Table 1: Comparison of delays for PCs S1,S2, and safe static with those for W4 (topological), W3 (static co-sensitization), W2 (floating), and static sensitization.

Circuit	W4 (longest top. delay)	W3 (static co-sens.)	S1	S2	Safe static	W2 (floating)	Static sens.
c432	17	17	17	17	17	17	17
c499	11	11	11	11	11	11	11
c880	24	24	24	24	24	24	24
c1355	24	24	24	24	24	24	24
c1908	40	40	40	40	37	37	37
c2670	32	30	30	30	30	30	30
c3540	47	46	46	46	46	46	46
c5315	49	47	47	47	47	47	47
c7552	43	42	42	42	42	42	42
csa.32.2	97	97	97	97	38	38	38
csa.64.4	161	161	161	161	46	46	46
csa.128.8	289	289	289	289	62	62	62
cla.16	34	34	34	34	34	34	33
tau92ex1	27	27	27	26	25	24	24
tau92ex2	93	62	55	46	44	42	41

however, we have observed that the CPU times for W3 (static co-sensitization), S1, S2 and safe static are very close to each other. From this, we conclude that safe static has the best accuracy/computation time trade-off under the W3 model.

Acknowledgment: We would like to thank Karem A. Sakallah for his comments regarding this work.

References

- [1] J. Benkoski, et al., "Efficient Algorithms for Solving the False Path Problem in Timing Verification," *Proc. Int. Conf. Computer-Aided Design*, 1987, pp. 44-47.
- [2] D. Brand and V.S. Iyengar, "Timing Analysis Using Functional Relationships," *Proc. Int. Conf. Computer-Aided Design*, 1986, pp. 126-129.
- [3] H. Chang and J.A. Abraham, "VIPER: An Efficient Vigorously Sensitizable Path Extractor," *Proc. 30th Design Automation Conf.*, 1993, pp. 112-117.
- [4] H.-C. Chen and D.H.-C. Du, "Path Sensitization in Critical Path Problem," *Proc. ACM Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU)*, 1990.
- [5] H.-C. Chen and D.H.-C. Du, "Path Sensitization in Critical Path Problem," *IEEE Trans. on CAD*, vol. 12, no. 2, February 1994, pp. 196-207.
- [6] S. Devadas, et al., "Certified Timing Verification and the Transition Delay of a Logic Circuit," *Proc. 29th Design Automation Conf.*, 1992, pp. 549-555.
- [7] S. Devadas, et al., "Computation of Floating Mode Delay in Combinational Circuits: Theory and Algorithms," *IEEE Trans. on CAD*, vol. 12, no. 12, December 1993, pp. 1913-1923.
- [8] S. Devadas, et al., "Event Suppression: Improving the Efficiency of Timing Simulation for Synchronous Digital Circuits," *IEEE Trans. on CAD*, vol. 13, no. 6, June 1994, pp. 814-822.
- [9] D.H.-C. Du, et al., "On the General False Path Problem in Timing Analysis," *Proc. 26th Design Automation Conf.*, 1989, pp. 555-560.
- [10] P.C. McGeer and R.K. Brayton, "Efficient Algorithms for Computing the Longest Viable Path in a Combinational Network," *Proc. 26th Design Automation Conf.*, 1989, pp. 561-567.
- [11] P.C. McGeer and R.K. Brayton, *Integrating Functional and Temporal Domains in Logic Design: The False Path Problem and its Implications*, Kluwer Academic Publishers, Boston, 1991.
- [12] S. Perremans, et al., "Static Timing Analysis of Dynamically Sensitizable Paths," *Proc. 26th Design Automation Conf.*, 1989, pp. 568-573.
- [13] E.J. Shriver and K.A. Sakallah, "RAVEL: Assigned-Delay Compiled-Code Logic Simulation," *Proc. Int. Conf. Computer-Aided Design*, 1992, pp.364-368.
- [14] J.P.M. Silva et al., "FPD - An Environment for Exact Timing Analysis," *Proc. Int. Conf. Computer Design*, 1991, pp. 212-215.
- [15] J.P.M. Silva and K.A. Sakallah, "An Analysis of Path Sensitization Criteria," *Proc. Int. Conf. Computer Design*, 1993, pp. 68-72.
- [16] J.P.M. Silva and K.A. Sakallah, "Efficient and Robust Test Generation-Based Timing Analysis," *Proc. Int. Symp. Circuits and Systems*, 1994, pp. 303-306.
- [17] H. Yalcin and J.P. Hayes, "Hierarchical Timing Analysis Using Conditional Delays," *Proc. Int. Conf. Computer-Aided Design*, 1995, to appear.