DELAY MODELS AND SENSITIZATION CRITERIA IN THE FALSE PATH PROBLEM

by

Patrick C. McGeer, Alexander Saldanha, Paul R. Stephan,
Robert K. Brayton, and Alberto L. Sangiovanni-Vincentelli

Memorandum No. UCB/ERL M92/63

16 June 1992

COVER PAGE

DELAY MODELS AND SENSITIZATION
CRITERIA IN THE FALSE PATH PROBLEM

by

Patrick C. McGeer, Alexander Saldanha, Paul R. Stephan,
Robert K. Brayton, and Alberto L. Sangiovanni-Vincentelli

Memorandum No. UCB/ERL M92/63

16 June 1992

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

DELAY MODELS AND SENSITIZATION CRITERIA IN THE FALSE PATH PROBLEM

by

Patrick C. McGeer, Alexander Saldanha, Paul R. Stephan,
Robert K. Brayton, and Alberto L. Sangiovanni-Vincentelli

Memorandum No. UCB/ERL M92/63

16 June 1992

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Delay Models and Sensitization Criteria in the False Path Problem [*]

Patrick C. McGeer Alexander Saldanha Paul R. Stephan Robert K. Brayton
Alberto L. Sangiovanni-Vincentelli

University of California - Berkeley

Abstract

We consider anew the false path problem in timing verification. We demonstrate that any solution to the false path problem inherently incorporates a *delay model*, and the answer is given in the context of the delay model. We make explicit the delay model underlying both the "floating" and "transition" sensitization computations, and give the basic assumption underlying gate sensitization. We extend sensitization theory on the delay model underlying the "floating mode" sensitization computation to general, asymmetric gates, and give a new delay model for transition-mode computations under bounded delay. We show that for every bounded delay model there is a natural time quantum such that every signal is a constant over every integer-multiple bounded interval of the quantum.

1 Introduction

Sensitization theory has classically been bound up with the *combinational false path problem* in timing verification. Early attempts[2, 3] were path-based, and derived to some extent from testing criteria. Broadly, such algorithms assumed networks were composed of simple gates, and asserted "non-controlling" (or identity) values on each off-path, or "side" input to each gate on the path. This approach had the difficulty that it was easy to exhibit circuits on which it reported true paths to be false. Brand and Iyengar[3] recognized this, and showed that a blind approach of only asserting identity values on some side inputs avoided this difficulty; however, the approach remained an approximation.

1989 marked a shift in approaches to sensitization in combinational timing verification. Working independently, two groups[12, 13, 8] demonstrated that the classic approaches were incomplete descriptions of the problem, in that values were asserted on circuit wires independent of time, and that by including temporal information an algorithm could obtain a tighter bound on the delay of a combinational logic circuit without underestimating the delay of the circuit.

In [12, 13], it was observed that sensitization algorithms asserted identity values on the off-path inputs in order to propagate changes in value, or *events*, down paths in the circuit. However, the identity values on the side inputs to a gate only needed to be asserted at the instant at which the event was to propagate through the gate. Merely asserting values at the proper time, however, was not enough. It was observed [12, 13, 10, 14], that each delay in a circuit was merely the representative of a *range* of delays – in fact, the maximum of the range – and the delay estimate returned by an algorithm operating on a circuit had to be valid over any selection of actual delays within the range. This property, known as the *monotone speedup* or *robustness* property, came to be regarded as an essential feature of any proposed algorithm. The solution proposed in [12, 13, 10, 14] became known as the *viability theory*. It was demonstrated that viability was robust on networks composed of symmetric gates; however, it was not demonstrated that viability was an exact estimate of the delay. In fact, little attention was paid to the semantics of the delay model underlying viability.

In [5], H-C Chen and D. H-C Du considered the false path problem on networks of simple gates, and modified the viability theory by incorporating the value of the on-path, as well as the off-path signals, into the sensitization conditions. This reduced the number of paths reported as true by their algorithm as compared with viability; however, it could be shown that it returned the same delay estimate as viability. As a theory it was weakened somewhat by its reliance on a simple gate structure.

An important idea in the Chen-Du paper was the introduction of the concept of the *stable time* of a signal. Though the Chen-Du paper (again) did not make explicit their delay model, the concept of computing a signal's stable time explicitly marked a turn away from the path-based concepts that dominated earlier research. In [6], this idea was taken a step further: rather than computing the sensitization conditions of a path at a time, these authors instead computed the sensitization conditions of *sets* of paths, of length $\geq$ some delay of interest d. This key idea was rapidly adopted by [11], in which the characteristic function of the stable time of a signal was computed explicitly.

In all this research, however, the delay model used by every author was the *monotone speedup* model defined in [12, 13, 10, 14], and the premises underlying the model were neither considered nor made explicit.

In the world of asynchronous circuits, race analysis is an active topic of research. This problem has some similarities to path sensitization; in particular, designers wish to know whether an asynchronous circuit with delays and feedback, in response to an excitation from the external environment,

[*] This research supported by Fujitsu Laboratories Ltd.

settled in a stable or unstable condition. It is well known that such analysis requires detailed and sophisticated knowledge of internal circuit delays; hence these researchers paid great attention to the delay models in their circuits, attempting to ensure that the models were realistic and conservative. An excellent compendium of such models is described by C-J. Seger and J. Brzozowski [17].

Despite the sophisticated modeling analysis of Seger and Brzozowski, little attention was paid to the problem of sensitization analysis; rather, exhaustive simulation was the vehicle of choice. Given the generally small circuits of concern to asynchronous designers and theorists, this was not an issue for these authors. However, timing verification of combinational circuits is typically run on very large circuits, and sophisticated sensitization analysis is required.

It is the goal of this paper to unify these two research approaches. Specifically, our goal is to develop a unified theory, encompassing both delay and logical information in a single mathematical framework, and use this information to devise sensitization algorithms under a number of delay models. The idea of incorporating delay and logical information in the same framework is not new: in [9], delay operators were added to the standard Boolean Algebra; their effect was to shift logical signals forward in time. In [18], a four-tuple (v, a, A, V); was stored at each node; a represented the minimum time that the node would switch, v the value before that time, A the maximum time at which the node would switch and V the value thereafter. However, these attempts modeled only some aspect of timing behavior or were delay-model specific. The model of [18], for example, failed to capture islands of stability within the interval (a, A). The model of [9] was valid only for a pure binary delay model. A more suitable formalism is the waveform algebra introduced by Augustin [1] to analyze timing models in hardware description languages. Augustin's model is a mapping from the integers onto $\{0, 1\}$. We extend this to a mapping from the real line to $\{0, 1, X\}$.

In this paper, we introduce an algebraic model of a signal waveform and delay models, analogous in the static domain to the switching algebra over Booleans and Boolean operators. In this framework, the primitive objects are waveforms, which are total functions from the real line onto the ternary set $\{0, 1, X\}$. In this framework, a gate delay model and its functional model combine as does a gate in a logic network: it takes as input some input signal waveforms, and produces an output signal waveform. Since the waveforms are static (though infinite) objects, algebraic operations over waveforms are well-defined; since the gate delay model plays the role in this algebra of an operator, it is an algebra effectively parameterized by the delay model; hence it can be used with a variety of delay models.

The remainder of this paper is organized as follows. In section 2, we introduce the waveform calculus as an extension to ternary simulation, and discuss efficient methods of representing input waveforms. In section 3 we give examples of four common delay models. In section 4 we give exact sensitization algorithms over two of these models, using the waveform calculus.

2 Ternary Delay Simulation and the Waveform Calculus

Ternary Boolean Algebras have been a feature of the analysis and synthesis of asynchronous circuits for many years; they have recently received increased attention due to Seger[16], and the derivative review work of Seger and Brzozowski[17]. Informally, the ternary model can be stated as follows. A third value, generally denoted X, is added to the Boolean Algebra. X is regarded as representing an unstable value.

The X value is generally regarded as modeling two phenomena:

1. Purely binary devices switch from the '0' to the '1' state instantly; however, if we build devices out of physical components there are intermediate values, and it takes time for the physical device to transit those intermediate values. For example, in most modern MOS processes, a voltage level between 4 and 5 volts represents the '1' value and a voltage level between 0 and 1 volt represents the '0' value; voltages between 1 and 4 volts do not represent any logical values, and as a result are modeled as an X.

2. The "delay" of a gate is not an independent variable; rather, it is a measure of the amount of time required for a switching event to occur. As a result, it is a complex function of a variety of factors, some static (operating temperature, process variation), and some dynamic (crosstalk, activity on neighboring transistors, slope of the input waveform, etc). There is some uncertainty in each of these variables, and as a result some uncertainty in the delay of a gate. When the value of a gate is unknown, it is modeled as an X.

X therefore represents every case where the value of a gate cannot be precisely measured as a pure Boolean value. It is straightforward to extend the standard binary algebra to ternary computation, and is given in the following table:

a	b	a AND b	a OR b	NOT a
0	0	0	0	1
0	X	0	X	1
0	1	0	1	1
X	0	0	X	X
X	X	X	X	X
X	1	X	1	X
1	0	0	1	0
1	X	X	1	0
1	1	1	1	0

The algebra is easily extended to general gates by functional composition. We adapt the terminology of Seger[15]. A **Ternary Variable** ranges over the set $\mathcal{T} = \{0, 1, X\}$; a **Ternary function** g is a mapping:

$$g : \mathcal{T}^n \mapsto \mathcal{T}$$

A containment relation $\sqsubseteq$ is defined over $\mathcal{T}$: $t \sqsubseteq t$ for each $t \in \mathcal{T}$, and, further, $0 \sqsubseteq X$, $1 \sqsubseteq X$. $\sqsubseteq$ extends naturally to vectors. Let $x_1, ..., x_n, y_1, ..., y_n$ be vectors over $\mathcal{T}^n$; $x_1, ..., x_n \sqsubseteq y_1, ..., y_n$ iff $x_i \sqsubseteq y_i$ for each i.

The ternary space $\mathcal{T}^n$ is easily related to the underlying binary space $\mathbf{B}^n$ through the following. A vector $x_1, ..., x_n$ over $\mathcal{T}^n$ is said to be a **vertex** if each $x_i \in \{0, 1\}$. It is easy to see, through simple functional composition, that if x is a vertex then $g(x_1, ..., x_n) \in \{0, 1\}$. We define the following evaluation rule for g over an arbitrary vector $x_1, ..., x_n \in \mathcal{T}$:

$$g(x_1, ..., x_n) = \begin{cases} 1 & g(y_1, ..., y_n) = 1 \ \forall y_1, ..., y_n \sqsubseteq x_1, ..., x_n \\ 0 & g(y_1, ..., y_n) = 0 \ \forall y_1, ..., y_n \sqsubseteq x_1, ..., x_n \\ X & \textit{otherwise} \end{cases}$$

The correspondence between vectors of the ternary space and cubes of the binary space is evident. For this reason, if $g(x_1, ..., x_n) = 1$, $x_1, ..., x_n$ is said to be an implicant of g; a maximal such implicant is said to be a prime of g.

In the sequel, one lemma will be of principal importance, and we state it here.

Lemma 2.1 *Let g be an arbitrary function of $x_1, ..., x_m$, where each x_j ranges over the set $\{0, 1, X\}$. Let $p_1, ..., p_n$ be the primes of g, $q_1, ..., q_r$ the primes of $\overline{g}$. Then $g(x_1, ..., x_m) = X$ if and only if there is no prime p_i such that $p_i(x_1, ..., x_m) = 1$ and no prime q_j such that $q_j(x_1, ..., x_m) = 1$.*

Proof: Immediate from the evaluation rule. ■

A delay model augments this algebra by associating a *time* t with each value of a gate or wire; the value of a gate at time t, $g(t)$, becomes a (generally complex) function of the values of the gate and its inputs at over some interval (t_0, t_1), where $t_1 \leq t$. Occasionally the interval is closed, and in this case $t_1 < t$.

This discussion is necessarily somewhat vague, since we are attempting to capture the semantics of a broad range of models. However, it is fairly easy to formalize.

Definition 2.1 *Given a gate g, an associated* **waveform** *for g, Ω^g, is a map:*

$$\Omega^g : \Re \mapsto \{0, 1, X\}$$

such that, for every t, every $\epsilon > 0$, if $\Omega^g(t+\epsilon) \neq \Omega^g(t)$ and $\Omega^g(t+\epsilon)$ and $\Omega^g(t)$ both contained in $\{0, 1\}$, then there is some $t < t_1 < t + \epsilon$ such that $\Omega^g(t_1) = X$.

By convention, each such map Ω^g is total: for each $t \in \Re$, $\Omega^g(t)$ is defined.

The definition of a waveform is designed to model our intuition of a logic signal varying over time; the restriction that any change in signal force a transition through X captures our intuition that the modeled physical waveform is a continuous map, and that the 0 and 1 values are physically separated.

Given a waveform Ω, and a real interval I, the partial waveform Ω_I is the waveform Ω restricted to the domain I. Ω_I is referred to as the **partial waveform** of interval I. With this in hand, we can define a delay model formally:

Definition 2.2 *A map M for a gate G, inputs $f_1, ..., f_r$:*

$$M : \Omega^{f_1}_{[0,t)} \times \Omega^{f_2}_{[0,t)} \times ... \times \Omega^{f_r}_{[0,t)} \times \Omega^{G}_{[0,t)} \mapsto \Omega^G(t)$$

is a **delay model** *if there is some subset S of the inputs of g such that $\Omega^{f_s}_{[0,t)}$ is a constant function for each $s \in S$, and, further, c is the cube $\prod_{s \in S} f_s = \Omega^{f_s}_{[0,t)}$, and G_c is a constant, then $\Omega^G(t) = G_c$.*

Broadly, the definition is designed to encapsulate our intuition about the transitions on a gate. The output waveform of a gate at t is determined by the input waveforms, and the gate waveform, occurring between 0 and some time t' preceding t. By convention, 0 is chosen as the base time; choosing a fixed base time for all model mappings enforces our intuitive notion that any delay model should be gauge symmetric upon the time axis, *i.e.* the value of the mapping specified by the delay model is independent of the base time.

The two conditions required for a mapping to be a delay model deserve some attention. The first simply enforces our common sense notion of causality; the output waveform of a gate between times t_1 and t_2 *cannot* depend upon input pin activity, or the state of the gate, *after* time t_2. The second condition is the basic assumption required for sensitization theory to have any real meaning. It states that the gate will not glitch or undergo any transient if the value of the gate is statically determined by its stable inputs.

2.1 Characteristic Functions

Consider a Boolean Algebra; this consists of a set of variables, each of which can take on a value in $\{0, 1\}$; a combinational network, evaluated statically, is the realization of a function over its input variables. A delay model, a network of gates and wires, a set of input variables, and a set of possible waveforms for each input variable, yields a Waveform Algebra. An assignment of one waveform to each input gives a **waveform vector**. The set of all waveform vectors forms a **wave space**, which (for n inputs) is denoted $\mathbf{W}^n$. A waveform vector is the analog, in wave space, to an input vertex in Boolean space. The analog to the gate in Boolean space is the pair (gate, delay model) in wave space; this product takes an input wave and produces an output wave.

The analogy between values of variables (in a Boolean Algebra) and waveforms of variables (in a waveform algebra) can be exploited to incorporate the idea of characteristic functions. In particular, we define:

Definition 2.3 *A* **characteristic function** *over a waveform space is a mapping of the form:*

$$\chi : W^n \mapsto \{0, 1\}$$

Conventionally, χ is associated with some set $S \subseteq W^n$: $\chi(w) = 1$ iff $w \in S$.

Characteristic functions will be a feature of the timing verification algorithms to be developed in the sequel. In particular, we will be calculating functions of the form:

$$\chi^{\Omega^g} \triangleq \{w | w \textit{ is a waveform vector producing } \Omega^g \textit{as the waveform on signal } g\}$$

2.2 Representation of Characteristics

Characteristic functions represent waveform vectors. A waveform, of course, is an infinite (in fact, uncountable) sequence of symbols from the set $\{0, 1, X\}$, representing the values of the wave at each (real) point in time, t. However, we do not toggle inputs infinitely often, and as a result, there are relatively few waveform vectors, and these are easily encoded. For example, in the most common combinational timing verification problems, we toggle the inputs once, at $t = 0$. In this case, the waveform space on n vectors $\mathbf{W}^n$ *may be represented* as the space $\mathbf{B}^n \times \mathbf{B}^n$, where (v_1, v_2) represents the (constant) binary input vectors applied at $t = -\infty$ and $t = 0$, respectively. Under these circumstances, the wave characteristic function may be thought of as a function:

$$\chi : \mathbf{B}^n \times \mathbf{B}^n \mapsto \{0, 1\}.$$

and is most conveniently represented as a standard Boolean function over $\mathbf{B}^n \times \mathbf{B}^n$

Delay models do not, of course, typically give an enumeration of such waveform maps. Typically, the models give rules for computing the output waveform, from which the waveform map is deduced.

3 Delay Models

Here, we briefly review some common delay models, and discuss them in the context of a timed ternary algebra. Most of these are taken from [15].

- **The Fixed Binary Pure Delay (FBPD) Model** This model is the most naive of the common delay models. Under this model, there is a fixed delay d_i from each gate input f_i to the gate output g; the instant value of $\Omega^g(t)$ is obtained by the static evaluation:

  $$\Omega^g(t) = g(\Omega^{f_1}(t - d_1), \Omega^{f_2}(t - d_2), ..., \Omega^{f_m}(t - d_m)).$$

  The difficulty with this model is twofold:

  1. As mentioned above, delays are typically uncertain; the pure delay model assumes delays are fixed, constant, and known.
  2. The model, as stated, does not explicitly introduce an X state into the computation of Ω^g; rather, it relies upon the input waveforms to provide the required transient state. As a result, the X state in this model effectively reduces to modeling the transition region, rather than modeling both the transition region and uncertainty in the actual value of a signal at a given time.

- **The Fixed Binary Pure Delay With Static Variation (FBPD-SV) Model** This model is simply the FBPD model, save that the constants d_i are treated as independent variables, whose actual values lie in the range $[d_i^{\min}, d_i^{\max}]$, with their actual values dependent upon static factors [process variation, circuit age, operating temperature (a static value on the time scales of interest), and so forth]. This model underlies the "transition delay" computation of [7]. The FBPD-SV model neglects dynamic factors such as crosstalk, degraded signals, slope factors and the like. This model is appropriate if such dynamic factors can be regarded as trivial, and if all uncertainty in delay and waveform value can be ascribed to static factors.

- **The Extended Bounded Delay (XBD) Model** Under the XBD model, the ranges $[d_i^{\min}, d_i^{\max}]$ represent uncertainties due to dynamic as well as static factors, and represent a transition region of uncertain width. As a result, pure transport of the input waveform to the output waveform at a specified time within the range is not permitted by the model. The computation of $\Omega^g(t)$ is given as a two-step process:

  $$F_i(t) = \begin{cases} \Omega^{f_i}_{(t - d_i^{\max}, t - d_i^{\min})} & \Omega^{f_i}_{(t - d_i^{\max}, t - d_i^{\min})} \textit{ is a constant} \\ X & \textit{otherwise} \end{cases}$$

  The values $F_i(t)$ form the effective values of the input waves, as presented to the output, at time t. If $\Omega^{f_i}_{(t - d_i^{\max}, t - d_i^{\min})}$ is a constant, then input f_i has not changed over the interval $(t - d_i^{\max}, t - d_i^{\min})$; since any change in state of f_i can only propagate to the output g at t if that change in state occurred between $(t - d_i^{\max}, t - d_i^{\min})$, it follows that the presented state of input f_i is simply the constant state $(t - d_i^{\max}, t - d_i^{\min})$. If, on the other hand, f_i changed state between $(t - d_i^{\max}, t - d_i^{\min})$, then the presented value of the input might be any state of f_i between the intervals, or a transient; the only reasonable value to choose in such circumstances is X. The value of $\Omega^g(t)$ is then easily found:

  $$\Omega^g(t) = g(F_1(t), ..., F_n(t))$$

  i.e., as the static (ternary) evaluation of g on the $F_i(t)$.

- **The Extended Bounded Delay-0 (XBD0) Model** This is the XBD model, with $d_i^{\min} = 0$ for all pins f_i of all gates g in the network. The XBD0 model is of particular interest, since it is the model underlying viability, and,in general, all the so-called "floating mode" sensitization calculations. Indeed, the *monotone speedup* property introduced in [12, 13, 14] may be viewed as an (incomplete) admission of the uncertainties in the XBD0 model.

For the remainder of this paper, we discuss sensitization in the context of the XBD0 and XBD delay models. We give a new sensitization procedure for these models, and demonstrate that this procedure reports the exact minimum delay in the XBD0 and XBD models. We show that this procedure gives the same delay estimate as the Chen-Du procedure on networks of simple gates, and as the viability procedure on networks of complex, symmetric gates; the Chen-Du and viability procedures can thus be regarded as exact algorithms for the XBD0 model on their respective special-case networks.

Othger delay models are possible. In recent work, Burch has introduced the binary chaos delay model [4]. This model is not treated here, although it is presently under consideration.

4 Sensitization and The Combinational Timing Verification Problem

Given a delay model M, the *combinational timing verification problem* for a circuit C under M is the following: given a family of possible waveforms on the combinational inputs of the circuit, such that each such waveform is a constant binary value on the intervals $(-\infty, 0)$ and (t_1, ∞) (i.e., each input changes state only within a fixed interval of 0), find the least positive t such that, for any possible combination of input waveforms, $\Omega^g_{(t,\infty)}$ is a binary constant for each circuit output g.

We consider this problem under the XBD0 and XBD circuit models. In order to do this, we introduce a new construct, the *characteristic function* on an interval.

4.1 Combinational Timing Verification Under the XBD0 Model

Under the XBD0 model, an input wave for input a is one of two forms:

$$\Omega^a = \begin{cases} x_{(-\infty,\infty)} & \text{or} \\ x_{(-\infty,0]}X_{(0,t_a)}\bar{x}_{[t_a,\infty)} & \end{cases} \quad \textit{for some } \; x \in \{0,1\}$$

where t_a is a positive constant associated with the variable a in the circuit. This permits us to derive an immediate result concerning the properties of circuit waveforms.

Lemma 4.1 *Let g be any gate in a logic circuit. Under the XBD0 model, under any waveform vector, we have $\Omega^g(t) \in \{1,0\}$ for any $t > 0$ implies $\Omega^g(t_1) = \Omega^g(t)$ for all $t_1 \geq t$.*

Proof: The proof is by induction on the level of gate g. By definition the result holds for the primary inputs. Suppose it is true for all gates of level $< N$. Consider some g at level N, and some arbitrary input waveform w. Let Ω^g be induced by w with $\Omega^g(t) = 1$. We have $g(f_1, ..., f_n)$, and by the XBD0 evaluation model, $1 = \Omega^g(t) = g(F_1(t), ..., F_n(t))$, where

$$F_i(t) = \begin{cases} \Omega^{f_i}_{(t-d_i^{\max},t)} & \Omega^{f_i}_{(t-d_i^{\max},t)} \textit{ is a constant} \\ X & \textit{otherwise} \end{cases}$$

since $g(F_1(t), ..., F_n(t)) \in \{0,1\}$. By lemma 2.1 there is some prime p of g, of g such that $p(F_1(t), ..., F_n(t)) = 1$. Consider an arbitrary $t_1 > t$. Since each input to g is of level $< N$, if $\Omega^{f_i}(t) \in \{0,1\}$, then by induction $\Omega^{f_i}(t_1) \in \{0,1\}$, and hence $F_i(t) \in \{0,1\} \Rightarrow F_i(t_1) \in \{0,1\}$; hence $p(F_1(t_1), ..., F_n(t_1)) = 1$, and by lemma 2.1, $\Omega^g(t_1) = 1$. ■

This lemma immediately permits a characterization of the waves given by the XBD0 model, as an immediate corollary.

Theorem 4.1 *Let g be any gate in a logic circuit. Under the XBD0 model, under any waveform vector, we have:*

$$\Omega^g = \begin{cases} x_{(-\infty,\infty)} & \text{or} \\ x_{(-\infty,0]}X_{(0,t_g)}\overline{x}_{[t_g,\infty)} & \text{or} \\ x_{(-\infty,0]}X_{(0,t_g)}x_{[t_g,\infty)} & \end{cases} \quad \text{for some } \; x \in \{0,1\} \tag{1}$$

Proof: Immediate consequence of the preceding lemma. ■

Since we have a characterization of the waveforms of any gate g, we can immediately proceed to the timing analysis problem. Recall:

$$\chi^{\Omega^g} \triangleq \{w | w \textit{ induces } \Omega^g \textit{ on gate } g\}$$

Now consider:

$$\chi^{\Omega^g_{[t,\infty)} \in \{0,1\}}$$

which is the set of all waveforms such that g is a binary constant on the interval $[t, \infty)$. Under the XBD0 model, the delay of a circuit with primary outputs $o_1, ..., o_n$, under input waveform vector w, is:

$$\max_i \min_t w \in \chi^{\Omega^{o_i}_{[t,\infty)} \in \{0,1\}}$$

And hence the delay over all waveform vectors may be written:

$$\max_i \min_t \chi^{\Omega^{o_i}_{[t,\infty)} \in \{0,1\}} \neq 0$$

We are faced with the calculation of $\chi^{\Omega^{o_i}_{[t,\infty)} \in \{0,1\}}$. Now, we have:

$$\chi^{\Omega^{o_i}_{[t,\infty)} \in \{0,1\}} = \chi^{\Omega^{o_i}_{[t,\infty)} = 0} + \chi^{\Omega^{o_i}_{[t,\infty)} = 1}$$

From lemma 4.1,

$$\begin{aligned} \chi^{\Omega^g_{[t,\infty)} = 0} &= \chi^{\Omega^g(t) = 0} \\ \chi^{\Omega^g_{[t,\infty)} = 1} &= \chi^{\Omega^g(t) = 1} \end{aligned}$$

We must calculate $\chi^{\Omega^g(t)=0}$ and $\chi^{\Omega^g(t)=1}$.

Lemma 4.2 *Let g be a gate with inputs $f_1, ..., f_r$. Let $p_1, ..., p_n$ be all the primes of g, and $q_1, ..., q_m$ be all the primes of $\overline{g}$. Then:*

$$\begin{aligned} \chi^{\Omega^g(t)=1} &= \sum_{i=1}^{n} (p_i(F_1, ..., F_r) = 1) \prod_{k=1}^{r} \chi^{\Omega^{f_k}(t - d_k^{\max}) \sqsubseteq F_k} \\ \chi^{\Omega^g(t)=0} &= \sum_{i=1}^{n} (q_j(F_1, ..., F_r) = 1) \prod_{k=1}^{r} \chi^{\Omega^{f_k}(t - d_k^{\max}) \sqsubseteq F_k} \end{aligned} \tag{2}$$

Proof: $w \in \chi^{\Omega^g(t)=1}$. Hence $\Omega^g(t) = 1$ when w is applied as the input waveform vector. Hence there is some prime p_i such that $p_i(F_1, ..., F_r) = 1$, and, further, $F_k \sqsupseteq \Omega^{f_k}(t - d_k^{\max})$, i.e., $w \in \chi^{\Omega^{f_k}(t-d_k^{\max}) \sqsubseteq F_k}$. Conversely, let $w \in \chi^{\Omega^{f_k}(t-d_k^{\max}) \sqsubseteq F_k}$ for all k, and, further, let $p_i(F_1, ..., F_r) = 1$. Then $\Omega^g(t) = 1$ by the evaluation rule, and hence $w \in \chi^{\Omega^g(t)=1}$. ■

The equations (2) suggest a simple recursive scheme for the computation of the exact true path delay under the Extended Bounded Delay-0 model.

4.2 The Chen-Du Model

Before leaving the XBD0 model we prove the speculated exactness of the Chen-Du and viability criteria for this model on the subset of gates over which they were defined. The Chen-Du criteria is defined over simple gates; viability, over simple and complex symmetric gates. Here we prove only exactness of Chen-Du over simple gates. This also serves to prove exactness of viability over those gates. The Chen-Du criterion is stated as: Given a network of simple gates (arbitrary input AND and OR gates, and NOT gates), and vector v, gate g:

1. If $g(v)$ is equal to g's controlled value, then the stable time of g under v, $s(g, v)$ is defined as the minimum t, over all inputs f such that $f(v)$ is at a controlling value for g, $t = s(f, v) + d_{g,f}^{\max}$, where $d_{g,f}^{\max}$ is the delay model's maximum delay between f and g.
2. If $g(v)$ is equal to g's non-controlled value, then $s(g, v)$ is equal to the maximum, over all inputs f of g, $s(f, v) + d_{g,f}^{\max}$

We analyze the Chen-Du method as follows. Consider the function

$\omega^{g,t} \triangleq \{v | t \geq \textit{stable time of } g \textit{ under } v \textit{ by Chen-Du model}\}$

Lemma 4.3 *If $c(g)$ is the controlled value for g, $nc(g)$ the non-controlled value for g, we have:*

$$\begin{aligned} \omega^{g,t} &= c(g) \sum_{f \in FI(g)} ((f = c(g))\omega^{f,t-d_{g,f}^{\max}}) \\ &+ nc(g) \prod_{f \in FI(g)} \omega^{f,t-d_{g,f}^{\max}} \end{aligned}$$

Proof: Immediate from the Chen-Du evaluation rules ■

We can now prove equality of the Chen-Du criterion and the exact analysis procedure on networks of simple gates. We have:

Theorem 4.2 *Let $u \in \omega^{g,t}$. Then $(u^*, u) \in \chi^{\Omega^g(t) \in \{0,1\}}$ where u^* denotes the binary vector opposite u on the $n-$cube.*

Proof: The proof is by induction on the level of g in the network. Trivial for primary inputs, so suppose it is true for the level of $g < N$. If the level of g is $= N$, then without loss of generality suppose g is an OR gate. We have two cases to consider:

1. $g(u) = 0$. Then $\omega^{g,t} = \prod_{f \in FI(g)} \omega^{f,t-d_{g,f}^{\max}}$, and $\chi^{\Omega^g(t)=0} = \prod_{f \in FI(g)} \chi^{\Omega^f(t-d_{g,f}^{\max})=0}$. By induction $u \in \omega^{f,t-d_{g,f}^{\max}}$ iff $(u^*, u) \in \chi^{\Omega^f(t-d_{g,f}^{\max})=0}$.
2. $g(u) = 1$. Then $\omega^{g,t} = \sum_{f \in FI(g)} (f = 1)\omega^{f,t-d_{g,f}^{\max}})$, and $\chi^{\Omega^g(t)=1} \sum_{f \in FI(g)} \chi^{\Omega^f(t-d_{g,f}^{\max})=1}$. By induction, $u \in \omega^{f,t-d_{g,f}^{\max}}(f = 1)$ if and only if $(u^*, u) \in \chi^{\Omega^f(t-d_{g,f}^{\max})=1}$.

■

This shows equality of the criteria on networks of simple gates, and hence the exactness of the Chen-Du procedure on such networks. Similar reasoning illustrates the exactness of viability on complex, symmetric gates. However, it is trivial to demonstrate that on networks of complex gates (symmetric and asymmetric) the Chen-Du procedure can be arbitrarily far from the minimum. In contrast, viability is only inexact on complex asymmetric gates.

4.3 Delay Calculation under the Extended Bounded Delay Model

Computation on the XBD0 model was greatly aided by the fact that each waveform was either constant on the positive half-plane or changed exactly once, from X to 0 or 1, on the half-plane. Since that was the case, as mentioned earlier, we could reduce our representation of the waveform vector to a single Boolean vector, and trim our search space accordingly. Under the XBD model, no such trimming is possible. Each gate changes potentially many times, from 1 to X to 0 and back again; further, such islands of stability are of great interest in some applications. As a result, we must track each waveform over every time of interest on the positive half-plane. Fortunately, bounded delay models have a property which makes this computation far more tractable.

4.4 A Theory of Quantum Time

At first glance, the requirement of symbolic ternary simulation at all time points can seem daunting. However, it may be demonstrated that, associated with each bounded delay model there is a fundamental time quantum, such that every waveform is a constant on each open interval bounded by integer multiples of the time quantum.

A bounded delay model M may be thought of as being fully described by a collection of constants: $(d_i^{\max}, d_i^{\min})$ for each connection i. (We model variations in the rise time of inputs by such a pair of constants, as well). Since we can, and do, write these constants as finite strings of digits, each such constant is a rational number. Let $\hbar_M$ denote the greatest common divisor of the set of constants associated with a bounded delay model M. We call $\hbar_M$ the **time quantum** of the model M. We have:

Theorem 4.3 *Let g be any gate in a circuit, M an XBD bounded delay model, $\hbar_M$ the associated time quantum. Then, $\Omega^g_{(n\hbar_M,(n+1)\hbar_M)}$ is a constant for each integer $n \geq 0$.*

Proof: The proof is by induction on the level of gate g. Each primary input changes state to X at $d_i^{\min}$ for some delay constant $d_i^{\min}$, and changes state away from X at $d_i^{\max}$; by definition, $d_i^{\min}$ and $d_i^{\max}$ are both integer multiples of $\hbar_M$, and hence so is their difference, giving the result. Suppose true for all gates of level $< N$. Consider some g at level N, and some arbitrary input waveform w. g is a gate $g(f_1, ..., f_r)$, and each f_i is of level $< N$. Choose $\hbar_M n < t_0 < t_1 < \hbar_M(n+1)$; the result holds iff $\Omega^g(t_0) = \Omega^g(t_1)$. $\Omega^g(t_0) = g(F_1(t_0), ..., F_r(t_0))$, and $\Omega^g(t_1) = g(F_1(t_1), ..., F_r(t_1))$. Now, $F_j(t_0) = \Omega^{f_j}_{(t_0-d_j^{\max}, t_0-d_j^{\min})}$ if $\Omega^{f_j}_{(t_0-d_j^{\max}, t_0-d_j^{\min})}$ is a constant, X otherwise, and similarly for $F_j(t_1)$. By the construction of $\hbar_M$, $d_j^{\max} = m\hbar_M$ for some positive integer m, and so:

$$\hbar_M n < t_0 < t_1 < \hbar_M(n+1) \Longrightarrow$$

$$\hbar_M n - d_j^{\max} < t_0 - d_j^{\max} < t_1 - d_j^{\max} < \hbar_M(n+1) - d_j^{\max} \Longrightarrow$$

$$\hbar_M n - (m\hbar_M) < t_0 - d_j^{\max} < t_1 - d_j^{\max} < \hbar_M(n+1) - (m\hbar_M) \Longrightarrow$$

$$\hbar_M(n-m) < t_0 - d_j^{\max} < t_1 - d_j^{\max} < \hbar_M(n+1-m)$$

and hence $t_0 - d_j^{\max}$ and $t_1 - d_j^{\max}$ are contained on the open interval $(\hbar_M(n-m), \hbar_M(n+1-m))$, and by induction $\Omega^{f_j}(t_0 - d_j^{\max}) = \Omega^{f_j}(t_1 - d_j^{\max})$. By exactly the same reasoning, $\Omega^{f_j}(t_0 - d_j^{\min}) = \Omega^{f_j}(t_1 - d_j^{\min})$. Hence if $\Omega^{f_j}_{(t_0-d_j^{\max}, t_0-d_j^{\min})}$ is a constant, $\Omega^{f_j}_{(t_1-d_j^{\max}, t_1-d_j^{\min})}$ is the same constant, and so $F_j(t_0) = F_j(t_1)$. If $\Omega^{f_j}_{(t_0-d_j^{\max}, t_0-d_j^{\min})}$ is not a constant, then $F_j(t_0) = X$. But then $\Omega^{f_j}_{(t_1-d_j^{\max}, t_1-d_j^{\min})}$ is not a constant, and so $F_j(t_1) = X$. In either case $F_j(t_0) = F_j(t_1)$, and, since j was chosen arbitrarily, $F_j(t_0) = F_j(t_1)$ for all $1 \leq j \leq r$. Therefore, $\Omega^g(t_0) = g(F_1(t_0), ..., F_r(t_0)) = g(F_1(t_1), ..., F_r(t_1)) = \Omega^g(t_1)$, giving the result. ■

This theorem gives a simple algorithm to compute the state of a network, its delay and hazard properties, given an XBD Model. Once again we use a method of characteristics. In particular, for each integer multiple of $\hbar_M$, $n\hbar_M$, and for each gate g we compute three functions from $\mathbf{W}^n \mapsto \{0, 1\}$

$$\begin{aligned} \chi_n^{g,0} &= \{w | \Omega^g_{(n\hbar_M, (n+1)\hbar_M)}(w) = 0\} \\ \chi_n^{g,1} &= \{w | \Omega^g_{(n\hbar_M, (n+1)\hbar_M)}(w) = 1\} \\ \chi_n^{g,X} &= \{w | \Omega^g_{(n\hbar_M, (n+1)\hbar_M)}(w) = X\} \end{aligned}$$

The equations for each of the characteristics are easily computed, as in the case for the XBD0 model. We have immediately:

Lemma 4.4 *Let g be a gate with inputs $f_1, ..., f_r$. Let $p_1, ..., p_n$ be the enumeration of the primes of g, and $q_1, ..., q_m$ be the primes of $\overline{g}$. Then:*

$$\begin{aligned} \chi_n^{g,1} &= \sum_{i=1}^{n} (p_i(F_1, ..., F_r) = 1) \prod_{k=1}^{r} \chi^{f_k, G_k}_{n - d_k^{\max}/\hbar_M} (F_k \sqsupseteq G_k) \\ \chi_n^{g,0} &= \sum_{i=1}^{n} (q_j(F_1, ..., F_r) = 1) \prod_{k=1}^{r} \chi^{f_k, G_k}_{n - d_k^{\max}/\hbar_M} (F_k \sqsupseteq G_k) \\ \chi_n^{g,X} &= \overline{(\chi_n^{g,0} + \chi_n^{g,1})} \end{aligned} \quad (3)$$

Proof: $w \in \chi_n^{g,1}$. For all $\hbar_M n < t < \hbar_M(n+1)$, $\Omega^g(t) = 1$ when w is applied as the input waveform vector. Hence there is some prime p_i such that $p_i(F_1, ..., F_r) = 1$, and, further, $F_k \sqsupseteq \Omega^{f_k}(t - d_k^{\max})$, i.e., $w \in \chi_l^{f_k, G_k}(G_k \sqsubseteq F_k)$ for $l < t - d_k^{\max} < l + 1$; it is easy to see $l = n - d_k^{\max}/\hbar_M$. Conversely, let $w \in \chi^{f_k, F_k}_{n - d_k^{\max}/\hbar_M}$, and $p_i(F_1, ..., F_r) = 1$; thence $F_k = \Omega^{f_k}(t - d_k^{\max})$ for all $n\hbar_M < t < (n+1)\hbar_M$ Then $\Omega^g(t) = 1$, for all $n\hbar_M < t < (n+1)\hbar_M$, and hence $w \in \chi_n^{g,1}$. The same proof holds for $\chi_n^{g,0}$, and the expression for $\chi_n^{g,X}$ is a direct product of the observation that these three functions must partition the waveform space. ■

(3) yields an obvious algorithm for computing temporal properties on a bounded delay model; we simply construct a matrix, whose columns correspond to integer multiples of the associated model quantum $\hbar_M$, and whose rows correspond to circuit gates. We call this matrix the Matrix of Time. The matrix has D columns and $|V|$ rows, where $|V|$ is the number of gates in the logic circuit and D is the maximum path length of the circuit in terms of the quantum $\hbar_M$. Three functions need be stored at each node, and it is easy to see that each function is of size linear in the number of primes of the gate function and its complement.

Direct computation of the critical delay is relatively straightforward on the Matrix of Time. It is fairly easy to see that:

$$\chi^{\Omega^g_{t,\infty} = 1} = \prod_{i=\lfloor t/\hbar_M \rfloor}^{D} \chi_i^{g,1}$$

and, similarly:

$$\chi^{\Omega^g_{t,\infty} = 0} = \prod_{i=\lfloor t/\hbar_M \rfloor}^{D} \chi_i^{g,0}$$

As before,

$$\chi^{\Omega^g_{t,\infty} \in \{0,1\}} = \chi^{\Omega^g_{t,\infty} = 0} + \chi^{\Omega^g_{t,\infty} = 1}$$

Efficiency of computation on the Matrix of Time is clearly dependent upon the number of columns of the matrix; which is to say, in terms of the static depth (relative to the time quantum) of the network. Recall that the constants of the model form the boundaries of the *uncertainties* in delay; arbitrary precision in these constants seems unlikely. Our experience with current industrial delay models suggest that these constants are on the rough order of a few percent of the standard gate delay; in this case, a quantum size of roughly five percent of a gate delay is probably a conservative estimate. Network depths in our experience are rarely deeper than 30-40 gates, and are often less than that; as a result, we expect that the Matrix of Time will have at most a few hundred columns; in this case, neither storage size nor current constraint-satisfaction techniques will be unduly strained by matrix computations. Still, this is very much a matter for experimental observation.

As with the XBD0 delay model, the waveform vectors can be represented by a short sequence of input vectors; in this case, the restriction to two vectors is unnecessary. In some applications (verification of wave-pipelined circuits, inputs which strobe at some multiple of the rate of other inputs, etc.), it is desirable to compose an input waveform vector from a sequence of Boolean input vectors; all that is required in this case is that one specify a separate max/min arrival time for each vector. Further, if some input is not changing between vectors, one simply specifies that its value is unchanged. However, asynchronous applications where an input can be generated by an output will pose more difficulties.

5 Conclusions

We have presented a unified approach to solving the false path problem under a variety of delay models. We have presented an exact sensitization criterion for networks of complex asymmetric gates under the XBD0 delay model (or "floating mode"), and XBD delay model. The practical efficiency of the technique remains to be determined, although an implementation of delay estimation under the XBD0 model has yielded impressive results [11]. The results described apply to combinational logic circuits as well as synchronous (edge-triggered) sequential circuits. The application of this formulation of delay models and sensitization criteria to hazard analysis and asynchronous timing verification is presently being explored.

References

[1] L. Augustin. An algebra of waveforms. *Technical Report, Computer Systems Laboratory, Stanford University*, 1989.

[2] J. Benkoski, E. Vanden Meesch, L. Claesen, and H. DeMan. Efficient algorithms for solving the false path problem in timing verification. In *IEEE International Conference on Computer-Aided Design*, 1987.

[3] Daniel Brand and Vijay S. Iyengar. Timing analysis using functional analysis. Technical Report RC 11768, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 10598, 1986.

[4] J. Burch. Delay models for verifying speed-independent asynchronous circuits. In *The Proceedings of the ACM international Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (τ '92)*, 1992.

[5] H. C. Chen and D. H. C. Du. On the critical path problem. In *ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (τ '90)*, 1990.

[6] S. Devadas, K. Keutzer, and S. Malik. Delay computation in combinational logic circuits: Theory and algorithms. In *IEEE International Conference on Computer-Aided Design*, 1991.

[7] S. Devadas, K. Keutzer, S. Malik, and A. Wang. Certified timing verification and the transition delay of a logic circuit. In *Proceedings of Tau '92*, 1992.

[8] David H. C. Du, Steve H. C. Yen, and S. Ghanta. On the general false path problem in timing analysis. In *Design Automation Conference*, 1989.

[9] S-T Huang, T-M Pang, and J-M Shyu. A new approach to solving false path problem in timing analysis. In *IEEE International Conference on Computer-Aided Design*, 1991.

[10] P. C. McGeer. *On the Interaction of Functional and Timing Behaviour of Combinational Logic Circuits.* PhD thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1989.

[11] P. C. McGeer, A. Saldanha, P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. Timing analysis and delay-fault test generation using path recursive functions. In *ACM International Workshop on Logic Synthesis*, 1991.

[12] Patrick C. McGeer and Robert K. Brayton. Efficient algorithms for computing the longest viable path in a combinational network. In *Design Automation Conference*, 1989.

[13] Patrick C. McGeer and Robert K. Brayton. Provably correct critical paths. In *Decennial CalTech VLSI Conference*, 1989.

[14] Patrick C. McGeer and Robert K. Brayton. *Integrating Functional and Temporal Domains in Logic Design: The False Path Problem and its Implications.* Kluwer Academic Publishers, 1991.

[15] C.-J. Seger. A bounded delay race model. In *IEEE International Conference on Computer-Aided Design*, November 1989.

[16] C.-J. Seger. *Models and Algorithms for Race Analysis in Asynchronous Circuits.* PhD thesis, Department of Computer Science, University of Waterloo, 1992.

[17] C.-J. Seger and J. Brzozowski. *Advances in Asynchronous Circuit Theory.* In Preparation, 1992.

[18] J. P. Silva, K. A. Sakallah, and L. M. Vidigal. FPD: An environment for exact timing analysis. In *IEEE International Conference on Computer-Aided Design*, 1991.