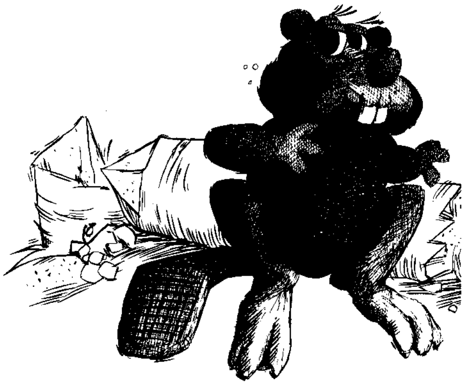
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



On Deducing Tight Bounds from Partial Timing Specifications

F. Mavaddat
T. Gahlinger

Research Report
CS-89-66

December, 1989

On Deducing Tight Bounds from Partial Timing Specifications

F. Mavaddat
T. Gahlinger

Department of Computer Science
University of Waterloo
Waterloo, Ontario

ABSTRACT

In this paper we propose a circuit delay model for the propagation of events in a digital device. We use this model to formulate several device timing characteristics, and introduce the concept of *separation bounds* to model the timing constraints specified in a device's waveform timing specifications. We show that even if we know the separation bounds on two pairs of events, (u, v) and (v, w), we cannot always deduce the correct separation bounds on (u, w). However, we show that the method proposed in [2]—to deduce tight constraints in a partial timing specification—is safe, in the sense that an affirmative answer to satisfiability is trustworthy while a negative answer may be pessimistic.

1. Background

The proper timing of digital signals is a major concern in the design of reliable systems, especially in the light of dramatic increases in the speed and complexity of digital components. In essence, 'improper' timing occurs when signals arrive at device inputs too early or too late to be recognized. Surveys of work in this area are described by Hitchcock[4], who also presents a method for verifying that circuit delays meet specified clock requirements, and Ruehli and Ostapko[6], who give an extensive list of references to digital timing and to physical and electrical aspects which must be taken into consideration.

However, these works typically assume investigators have at their disposal a comprehensive body of information related to the circuits under study, with the capability to modify the circuitry to resolve timing problems. In a previous paper we examined the types of timing problems a system designer encounters when connecting prefabricated components[2]. In this case, the designer must rely on the information contained in the manufacturer's waveform timing specifications. Other, mathematically less-rigorous, investigations of the waveform convention have appeared in[1, 3, 5].

Timing specifications restrict knowledge of device behaviour to the timing of the signal behaviour at device ports. The specifications consist of a set of digital waveform illustrations, annotated by timing parameters which specify bounds on the timing of signal transitions. Typically, the specifications are only *partial* descriptions of the timing of activities at the ports.

We formulated the timing information conveyed by the waveform convention as a set of linear constraints on the timing of pairs of signal transition events. A key point of the formulation is the distinction between timing constraints as *produced constraints* if they reflect the timing behaviour of the device in response to proper inputs, or *required constraints* if they reflect the conditions that must be met by the device's environment in order for the device to function properly.

We used linear optimization techniques to verify that the timing information expressed by the waveform convention is 'consistent', and to test whether the produced constraints of one device 'satisfy' the required constraints of a device connected to it, in the following sense. Let V be a set of integer-valued variables representing waveform transition times, and let Γ be a set of linear constraints on V. A constraint $\gamma_{uv} \in \Gamma$ is an expression of the form

$$\gamma_{uv}: \; v - u \leq b_{uv} \, ,$$

where u and v are transition events, and b_{uv} is an upper bound on the timing of the transitions (a lower bound $l_{uv} \leq v - u$ can be represented as an upper bound by reversing signs and the direction of the inequality, namely, $u - v \leq -l_{uv}$). We define the set Γ to be *consistent* if the solution space of Γ is non-empty. Given two constraints γ_{uv} and $\hat{\gamma}_{uv}$:

$$\gamma_{uv} \; satisfies \; \hat{\gamma}_{uv} \;\; \Leftrightarrow \;\; b_{uv} \leq \hat{b}_{uv} \, .$$

The test for satisfiability arises when the constraint $\hat{\gamma}_{uv}$ represents a required constraint on (u, v) for a given device—namely, one which must be met by the timing of devices connected to that device—and γ_{uv} represents a constraint on (u, v) due to the other devices; γ_{uv} will either be a produced constraint of these devices, or it is inferred from other produced constraints in the manner discussed below.

The restricted nature of the consistency and satisfiability problems permits efficient solutions using shortest-path methods. We represent V and Γ as a labelled directed graph $G = (V, A)$, with nodes $V = \{1,...,n\}$ and arcs $A = \{a_{uv} \mid v - u \leq b_{uv} \in \Gamma\}$ where, associated with each arc a_{uv}, oriented from node u to node v, is an integral length b_{uv}. We also use (u, v) to refer to arc a_{uv}. A path, $p(u, v)$, between nodes u and v in G, is a sequence of arcs $p(u, v) = (u, u_1), (u_1, u_2), ..., (u_k, v)$. A cycle in G is a path $p(u, u)$; the length of a path is the sum of its arc lengths. A *negative cycle* is a cycle $p(u, u)$ with length < 0.

We solve the problems by computing the shortest paths between all pairs of nodes in G. It can be shown that Γ is inconsistent if G contains a negative cycle. Let G^* be the final graph obtained from G after computing the shortest paths. We refer to G^* as the *tightening* of G, and to the label on an arc (u,v) in G^* as the *tightened* bound on constraint γ_{uv} in Γ. The method of tightening allows us to infer new information about the timing of two events for use in a satisfiability test.

In the remainder of this paper we justify the use of the tightening method. For the sake of convenience, we represent both lower and upper bounds between two events using the following equivalent interval notation:

$$\gamma_{uv}\colon\ b_{uv}^{-} \le v - u \le b_{uv}^{+},$$

Moreover, we represent this information in our graphs with a single arc oriented from i to j labelled by the interval $[b_{ij}^{-}, b_{ij}^{+}]$. We refer to such a graph as an *interval-labelled* DAG, or *ILDAG.*

2. Elements of the Delay Model

A device manufacturer cannot guarantee that each instance of a device will exhibit the same timing behaviour. The differences are due to variations in internal delays from one instance to the next; an instance is acceptable if the delays fall into acceptable ranges for that device. In this section we use the well-formed interconnection of *firing elements* and *delay elements* to model the timing of acceptable instances of a device.

Let us define a *delay component*, *DC*, as an acyclic connection of *firing elements* and *delay elements.* These elements constitute the *static* aspects of *DC*, and we represent the static aspects as ILDAG, $G^d = (V, A)$, which we refer to as *DC*'s *delay graph*, where V is a set of n nodes, representing the firing elements of *DC*, and A is a set of m arcs, representing the delay elements. A distinguished source node, v_0, represents *DC*'s *input* firing element; sink nodes and a (possibly empty) subset of the internal nodes in G^d represent *DC*'s *output* firing elements. We refer to these nodes collectively as the *port nodes.*

Let d_k^- and d_k^+ denote integer bounds on the acceptable delay range of the k-th delay element. Then, A contains an arc a_{uv}, oriented from node u to node v with label $[d_k^-, d_k^+]$, if u is connected to v by the k-th delay element. Let D denote the m-coordinate bounded *delay space* of *DC*, where

$$D \;=\; \{\, (d_1, \ldots, d_m) \mid d_k^- \le d_k \le d_k^+,\ k = 1..m \,\}.$$

We refer to a point $d = (d_1 , \ldots , d_m)$ in this space as an *instance* of DC; thus, for a given instance d, the i-th arc in A has a fixed delay, d_i.

We now describe the *dynamic* aspects of DC. We partition the set of firing elements in $V - \{ v_0 \}$ into two sets: a set V^e of *early firing* elements and a set V^l of *late firing* elements. Thus, $V = V^e \cup V^l \cup \{ v_0 \}$. We associate with each firing element v in V a *firing time* (or *event*) *at* v, denoted by t_v.

Let $F(v)$ be the set of pre-neighbours of v along fan-in arcs; namely,

$$F(v) = \{ u \mid a_{uv} \in V \}.$$

For a given instance, d, we define for each node v in V the firing time, t_v^d, for that instance, where

$$t_v^d \;=\; \begin{cases} 0, & \text{if } v = v_0 \\ \min\limits_{u \in F(u)} (t_u^d + d_{uv}), & \text{if } v \in V^e \\ \max\limits_{u \in F(u)} (t_u^d + d_{uv}), & \text{if } v \in V^l\,. \end{cases} \tag{5.1}$$

We represent the dynamic aspects of DC as a complete labelled DAG, $G^s = (V, S)$, which we refer to as DC's *separation graph*. Associated with each arc s_{uv} in S is an integer weight σ_{uv}, which we refer to as the *(potential) separation* of (u,v), representing the widest time between their firings over all instances in D; namely,

$$\sigma_{uv} \;=\; \max_{d \in D} (t_v^d - t_u^d). \tag{5.2}$$

We use the concepts of delay, event, and separation to model the timing of signals at device ports. The derivation of the delay intervals, typically from circuit layout information, is beyond the scope and interest of this paper. We assume this information is available to us from circuit timing analysis or simulation.

An event occurs at an early firing element in the delay component as soon as the element receives an event from *some* pre-neighbour of the element. An event occurs at a late firing element as soon as the element has received events from *every* pre-neighbour of the element. Events at late elements model the start event of a valid period of data and events at early elements model the end event of a valid period of data. This behaviour is modelled by (5.1).

We consider only the component's behaviour due to a single event at a single source v_0. The single source is justified by the fact that the start of testing of a physical device can usually be traced to some initial trigger point. A single event at the source is justified by considering all possible instances of the device. By the definition of the delay component, a single event at the source leads to a single event at every firing element connected to the source by a delay path; thus, we refer to a node and an event at a node interchangeably.

The efficient computation of the separation bound in (5.2) is left as an open problem. For the examples in this thesis, we compute separations by enumeration over delay bounds.

The separation σ_{uv} of (u,v) is the upper bound on the time between their firings, namely, their widest separation over all possible instances. Let us use σ_{uv}^{+} interchangeably with σ_{uv} to denote this. Let σ_{uv}^{-} denote the lower bound on the separation of (u,v). To derive σ_{uv}^{-} we use the well-known result that, given any set of numbers, X,

$$\min(X) = -\max(-X). \tag{5.3}$$

By (5.2) and (5.3) it then follows that

$$\sigma^-_{uv} \;=\; -\sigma^+_{vu} \;=\; \min_{d \in D}(t^d_v - t^d_u).$$

Let $\langle \sigma^-_{uv}, \sigma^+_{uv} \rangle$ denote the range of separations, or *separation-interval*, of (u,v). For the sake of clarity, we use separation-intervals to illustrate separation graphs, and refer to these as *separation-interval graphs.* For every pair of arcs s_{uv} and s_{vu} in the separation graph, with separations σ_{uv} and σ_{vu}, the separation-interval graph has one arc, say s_{uv}, with separation-interval label $\langle -\sigma_{vu}, \sigma_{uv} \rangle$. Thus, we can represent G^s as an ILDAG.

For example, Figure 5.1 illustrates the delay graph and the separation-interval graph of a component DC with one input and three output firing elements. In this example, there are no internal nodes. We assume that the delay ranges are known from circuit information. Separations were computed by enumeration over the delay ranges.

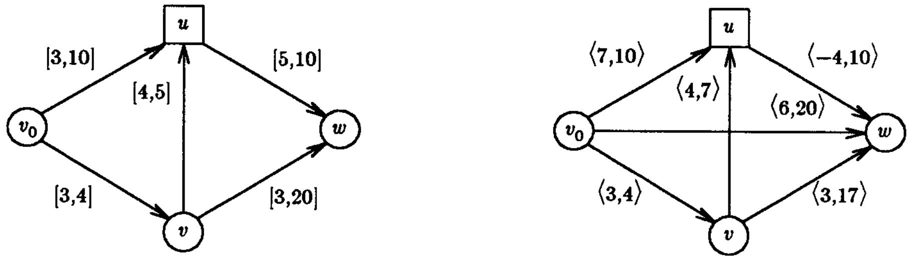


Figure 5.1. Example of a delay graph and its separation-interval graph. Early firing nodes are depicted as circles and late firing nodes as boxes; thus, u here is a late node.

We now demonstrate that the sum of the separations along any path between two nodes of a separation graph is at least as wide as the separation of the nodes.

Lemma 5.1. *Let u_1 and u_j be two nodes of a separation graph, and let $\sigma_{u_1 u_j}$ be the separation of (u_1, u_j). Without loss of generality, let $s_{u_1 u_2}, \cdots, s_{u_{j-1} u_j}$ be any path from u_1 to u_j, with separations*

$\sigma_{u_1 u_2}, \cdots, \sigma_{u_{j-1} u_j}$. *Then,*

$$\sigma_{u_1 u_j} \leq \sigma_{u_1 u_2} + \cdots + \sigma_{u_{j-1} u_j}.$$

Proof. By (5.2), we have

$$\sigma_{u_1 u_j} = \max_{d \in D} (t_{u_j}^d - t_{u_1}^d).$$

Suppose the maximum occurs when $d = M$; thus,

$$\sigma_{u_1 u_j} = t_{u_j}^M - t_{u_1}^M .$$

We can re-write $t_{u_j}^M - t_{u_1}^M$ as

$$t_{u_j}^M - t_{u_1}^M$$

$$= (t_{u_j}^M - t_{u_{j-1}}^x) + (t_{u_{j-1}}^x - t_{u_{j-2}}^x) + \cdots + (t_{u_2}^x - t_{u_1}^M), \quad (5.4)$$

where x is any instance in D. But for every term $(t_{u_k}^a - t_{u_{k-1}}^b)$ in the right-hand side of (5.4), it is evident that

$$(t_{u_k}^a - t_{u_{k-1}}^b) \leq \max_{d \in D} (t_{u_k}^d - t_{u_{k-1}}^d) = \sigma_{u_k u_{k-1}}.$$

Thus,

$$t_{u_j}^M - t_{u_1}^M \leq \sigma_{u_1 u_2} + \cdots + \sigma_{u_{j-1} u_j},$$

which completes the proof.
■

This result depends only on the definition of separation as the maximum distance between firing times at nodes, and not on how these times were arrived at, namely, whether due to early or late firings. We use this result in the following section, where we consider the relationship between separations and derived timing constraints.

3. Device Timing Specifications

In practice, a designer doesn't know the delay intervals of a device. Rather, we argue on the basis of the delay model presented above, that a manufacturer's data sheets usually provide the designer with a partial specification of the *separation intervals* between port events. Figure 5.2 illustrates a partial specification of a hypothetical delay component. In keeping with our argument, we now use the separation-interval notation to show the timing between waveform transition events.

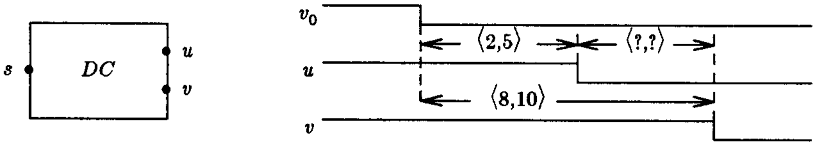


Figure 5.2. Partial specification of event separations.

Let DC be a delay component with separation graph $G^s = (V,A)$. A *(timing) specification* of DC is a subgraph of G^s, $\hat{G}^s = (\hat{V},\hat{A})$, where $\hat{V} \subseteq V$ is the set of port nodes in V, and $\hat{A} \subseteq A$ is a subset of the arcs between port nodes in G^s. Associated with each arc a_{uv} in $\hat{A}$ is the separation σ_{uv} of (u,v). We refer to (u,v) as a *specified pair* if a_{uv} is in $\hat{A}$, and as an *unspecified pair* otherwise. $\hat{G}^s$ is a *partial specification* if it contains any unspecified pairs.

Let us now consider the satisfiability of a specification. Suppose that the outputs of a device X are connected to inputs of a device Y. Let $\delta_{uv} = [\delta^-_{uv}, \delta^+_{uv}]$ be a required constraint in Y's timing, namely, one which must be met by X if Y is to function properly. Let $\tau_{uv} = [\tau^-_{uv}, \tau^+_{uv}]$ be the timing on (u,v) due to X. Then,

$$\tau_{uv} \text{ satisfies } \delta_{uv} \;\Leftrightarrow\; \delta^-_{uv} \le \tau^-_{uv} \;\&\; \tau^+_{uv} \le \delta^+_{uv} \,. \tag{5.5}$$

It is evident that (5.5) is equivalent to the following expression

$$\tau_{uv} \text{ satisfies } \delta_{uv} \;\Leftrightarrow\; \tau^+_{vu} \leq \delta^+_{vu} \;\&\; \tau^+_{uv} \leq \delta^+_{uv} \;. \qquad (5.5')$$

For the sake of convenience, we use (5.5') in our subsequent arguments, and assume that satisfiability has been formulated in terms of a test on upper bounds; thus, we may drop the superscript '+'.

Now suppose that (u, v) is an unspecified pair in X; in this case, τ_{uv} is deduced from other constraints by using the method of tightening described in the first section. The following theorem tells us that we cannot determine, from a specification alone, whether a deduced timing is indeed the potential separation.

Theorem 5.1. *The information contained in a partial specification of a delay component is insufficient for deducing the potential separation of an unspecified pair.*

Proof. (By counterexample). Consider the device and partial specification of Figure 5.2; Figure 5.3 shows two delay graphs for the device. The delays shown yield the specified separations of Figure 5.2, but different separations for the unspecified pair (u,v).
■

This result tells us that, unless we know the underlying delays in a circuit, we cannot guarantee that we have deduced the correct potential separation from a partial specification alone. This result suggests that we re-examine the methods we used to derive the timing on an unspecified pair of events.

In[2] we used the method of shortest paths to derive tight bounds on a set of constraints. We now show that the use of this method to verify the reliability of device connections is safe, in the sense that an affirmative answer to reliability is trustworthy, while a negative answer may identify a connection as unreliable when in fact it is reliable.

Let $\hat{G}^s$ be a partial specification of a device X. We claim that $\hat{G}^s$ in fact represents the set of produced constraints of a device. Let P_{uv} be the set of all paths from node u to node v in $\hat{G}^s$. Let p_{uv} be a path in P_{uv}, and let $|p_{uv}|$ denote the sum of the separations on the arcs in p_{uv}. The shortest

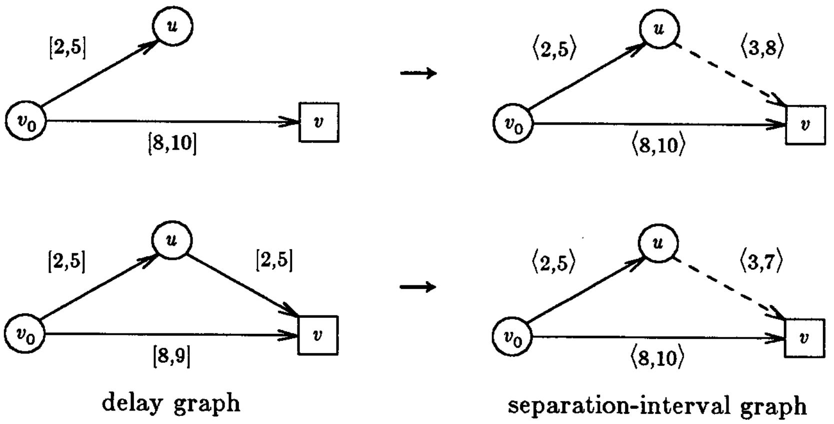


Figure 5.3. Counterexample used in the proof of Theorem 5.1.

path from u to v, sp_{uv}, is defined as

$$sp_{uv} \;=\; \begin{cases} \text{undefined}, & \text{if } \hat{G}^s \text{ is inconsistent} \\ \text{unbounded}, & \text{if } P_{uv} = \emptyset \\ \min\limits_{p_{uv} \in P_{uv}} \left(|p_{uv}| \right), & \text{otherwise} . \end{cases} \tag{5.6}$$

The following theorem states the relationship between the length of the shortest path between two nodes of the (timing specification) graph $\hat{G}^s$ and the potential, or widest, separation of the nodes.

Theorem 5.2. *For any pair of nodes (u,v) in a partial specification $\hat{G}^s$, the length of the shortest path from u to v is at least as large as the potential separation; namely,*

$$\sigma_{uv} \;\leq\; sp_{uv} \,.$$

Proof. We may assume that $\hat{G}^s$ is consistent. By (5.6), it is evident that the claim holds if there is no path from u to v. Suppose there is a path; by Lemma 5.1, σ_{uv} is less than or equal to the length of any path from u to v.
■

The next theorem establishes the safety of bounds derived by the method of shortest paths.

Theorem 5.3. *Let $\hat{\sigma}_{uv}$ be the separation on (u,v) in a device X, and let τ_{uv} be the shortest path bound on (u,v). Let σ_{uv} be the separation of (u,v) in a device Y. Then*

$$\tau_{uv} \textit{ satisfies } \sigma_{uv} \Rightarrow \hat{\sigma}_{uv} \textit{ satisfies } \sigma_{uv} .$$

Proof. By (5.5'), (τ_{uv} satisfies σ_{uv}) $\Leftrightarrow$ ($\tau_{uv} \leq \sigma_{uv}$). By Theorem 5.2, $\hat{\sigma}_{uv} \leq \tau_{uv}$; thus, $\hat{\sigma}_{uv} \leq \sigma_{uv}$, namely, $\hat{\sigma}_{uv}$ satisfies σ_{uv}.
■

This result tells us that if a computed separation passes a satisfiability test then we can be assured that the correct separation, based on the underlying delays, will pass the test as well; consequently, our tightening techniques are safe. However, since they could be wider than the potential separation, they could also fail a satisfiability test which the potential separation might pass. Consequently, they are pessimistic.

4. Summary

In this paper we viewed a circuit as a network of early and late firing elements, separated by delay elements. A specific instance of the circuit, namely, a manufacturer's product, will exhibit its own characteristic delays. We suggested that a manufacturer's timing specification is a partial description of the set of bounds on the separation between the firing times calculated by considering all instances of the circuit; typically, these bounds are derived from the electrical properties of the circuit, or approximated by simulation. Theorem 5.1 states that unless we know the circuit delays, we cannot derive these bounds with certainty. However, Theorem 5.2 indicates that the method of shortest paths used to derive the separation between firing times will compute a separation at least as wide as the separation due to the underlying circuit delays. Finally, Theorem 5.3 states that if the computed separation passes a satisfiability test then we can be assured that the correct separation—the one based on the underlying delays—will pass the test as well. Consequently, our tightening techniques are safe; at the same time they are pessimistic, in the sense that if the satisfiability test fails then we might label connections as unreliable which might nevertheless be reliable based on their 'correct' separations.

5. References

1. G. Borriello, "A New Interface Specification Methodology and its Application to Transducer Synthesis," PhD Thesis, Computer Science Division, EECS, University of California, Berkeley (May, 1988).

2. J.A. Brzozowski, T. Gahlinger, and F. Mavaddat, "Consistency and Satisfiability of Waveform Timing Specifications," *Networks,* (to appear).

3. B. Dasarathy, "Timing Constraints of Real-Time Systems: Constructs for Expressing Them, Methods of Validating Them," *IEEE Proceedings of Real-Time Systemns Symposium,* pp. 197-204 IEEE, (Dec. 1982).

4. R.B. Hitchcock, "Timing Verification and the Timing Analysis Program," *19th Design Automation Conference,* pp. 594-604 IEEE, (1982).

5. L. Kara, R. Rastogi, and K. Kawamura, "TDS: An Expert System to Automate Timing Design for Interfacing VLSI Chips in Micrcomputer Systems," *International Conference on Computer-Aided Design*, pp. 362-365 (1986).

6. A. Ruehli and D.L. Ostapko, "VLSI Circuit Analysis, Timing Verification and Optimization," pp. 129-146 in *VLSI CAD Tools and Applications*, ed. Fichter and Morf, Klewer Publ., Boston (1987).