

Modeling timing correlation and the accurate timing verification of digital interface circuits

Marco A. Escalante and Nikitas J. Dimopoulos

Department of Electrical and Computer Engineering
University of Victoria
PO Box 3055, Victoria BC, CANADA V8W 3P6

Abstract

Recent research on timing verification of digital interface circuits has implicitly assumed that the delays of different signal transitions in the circuit are independent of one another. However timing diagrams in data sheets clearly specify correlation information. If timing correlation is ignored the results of verification can be overly pessimistic. In this paper we propose a probabilistic timed Petri net capable of representing timing correlation. For the subclass of periodic nets with AND and OR causality we develop a procedure that not only accurately checks if each one of the given timing constraints is satisfied but if the check fails also determines the probability that a particular timing constraint would be violated.

1. Introduction

Complex systems can be constructed hierarchically using simpler modules. The integration of a system involves the design of inter-module hardware interfaces which provide connectivity and possibly protocol conversion. The automation of interface timing verification is important for the design of reliable systems particularly due to the trend towards increasingly high-reliable and high-performance systems.

In order to design or verify an interface circuit, only the external behavior of the components needs be specified. Usually a component specification is given in the form of a functional description of its pins accompanied with timing diagrams which detail the temporal behavior of the interfacing signals. Signal transitions encode the events of a protocol. In the sequel we use the terms event and signal transition interchangeably.

Signal transition graphs or STG's [6] have been used to represent component protocols. Nodes in an STG correspond to the protocol events while edges represent timing relationships between events. There are two basic types of timing information: circuit delays and timing constraints. Circuit delays describe the intrinsic delays through the gates that comprise the two components and the interface. Timing constraints indicate the timing relationship that

must be observed between a pair of events for proper operation of the circuit. A delay or a timing constraint are usually represented as intervals $[t_{min}, t_{max}]$ denoting the lower and upper bounds. For a delay, the interval accounts for differences in process fabrication, variations of temperature, etc. For a constraint, the interval the tolerance of the circuits (e.g. a set up timing constraint). The objective of timing verification is to determine whether all constraints are satisfied for all possible values of the circuit delays.

In this paper we discuss the representation of time correlation in component interface specifications. The next section surveys previous work on interface timing verification. Section 3 introduces timing correlation information. Section 4 presents our probabilistic model which is based on a probabilistic timed Petri net. Section 5 illustrates our technique using an example. Finally section 6 discusses future directions.

2. Related work

The timing verification problem has been formulated as finding the solution of a set of linear/min/max delay equations [1,2]. Because the general problem is NP-complete, recent work has concentrated on finding efficient solutions of the linear/max problem [cf. 3] which is more tractable. In section 4.2 we show that such min/max timing relationships can be derived from the AND/OR causalities studied in the concurrent-systems literature [4]. We also show that neglecting correlation information may result in pessimistic estimates of event time separation between.

A major contribution of our approach is that we maintain a causal framework while being able to model as timing correlation seemingly "non-causal" phenomena provided in standard data sheets by chip manufacturers. An additional benefit of our approach is that if a particular constraint is not satisfied, our procedure then determines the probability that such timing constraint would be violated.

Currently the procedure we have developed is applicable to periodic systems. Note however that this is not a serious limitation since standard interface protocols are designed to function periodically (e.g. microprocessor transfer cycles).

3. Timing correlation

To understand time correlation, consider the following example taken from a microprocessor transfer cycle (see Figure 1). The address lines *add* are used to select a particular device. In order to avoid incorrect selection while the address lines are changing, a strobe signal *as* is used to indicate when the *add* lines contain a valid address. A timing diagram describing this protocol is shown in Figure 1. Typical values for the timing parameters are listed in Table 1. The manufacturer guarantees that a chip will perform within the minimum and the maximum given times.

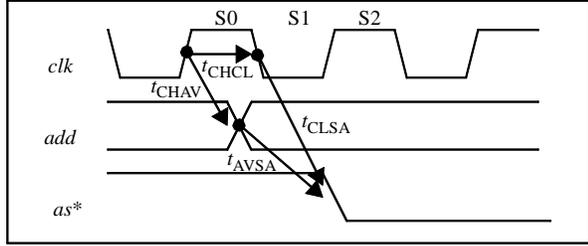


Figure 1. Timing relationship between *add* and *as*.

Symbol	Timing parameter	Min	Max	Unit
t_{CHAV}	Clock high to Address valid	0	40	ns
t_{CLSA}	Clock low to <i>as</i> asserted	0	40	ns
t_{AVSA}	Address valid to <i>as</i> asserted	20	-	ns
t_{CHCL}	Clock high to Clock low	35	45	ns

Table 1: Timing specifications (Motorola MC68030)

Figure 2 shows the corresponding STG. One can identify two clock events indexed by the corresponding clock states, the address lines becoming valid $add\uparrow$, and the assertion of the address strobe signal $as+$. The timing parameters of Table 1 label the edges of the graph.

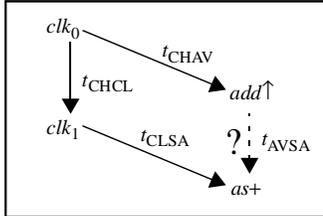


Figure 2. Signal transition graph.

With the exception of t_{AVSA} the other timing parameters describe causal relationships between the corresponding events. Timing t_{AVSA} specifies that $add\uparrow$ always precedes $as+$ by at least 20ns. One may be tempted to add a causal link from $add\uparrow$ to $as+$ (the dotted edge in Figure 2). A more appropriate modeling uses the concept of *time correlation*. Note that both *add* and *as* are generated concurrently by two logic blocks from the clock edges. Let us call d_1 and d_2 the delays from clk_0 to $add\uparrow$ and $as+$ respectively. From the parameters in Table 1, it can be inferred that $d_1 \in [0, 40]$ and $d_2 \in [35, 85]$ ignoring t_{AVSA} . There are some combinations of values of d_1 and d_2 for which t_{AVSA} is not satisfied. For example if $d_1 = 40$ and $d_2 = 35$.

However the sources of delay variation (e.g. process fabrication tolerances, temperature effects, etc.) are *likely* to affect both delays in the same direction, making the above choice of values of the delays improbable. We say that d_1 and d_2 are not independent of each other but are correlated. Figure 3 shows a joint probability density function of two variables. Note that the values that y can take are restricted by the value of x . A similar behavior is to be expected between d_1 and d_2 .

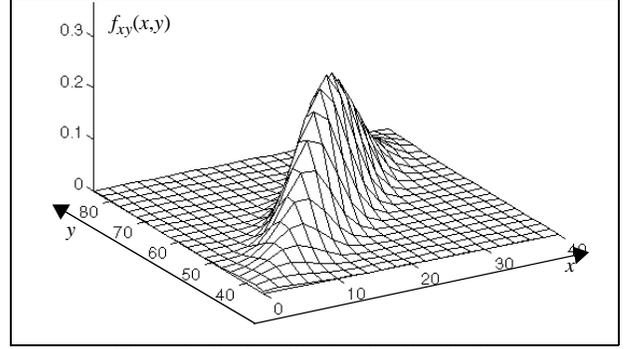


Figure 3. Probability density distribution.

In Figure 4a the shaded area describes the projection of the probability distribution function $f_{d_1,d_2}(d_1,d_2)$ which better expresses the intention of parameter t_{AVSA} , namely that $add\uparrow$ precedes $as+$ by at least 20ns. Note that if d_1 and d_2 were independent, the projection would be the rectangle bounded by a dashed line. The boundary of the projection in Figure 4a can be described by linear expressions on d_1 and d_2 . Although arbitrary boundaries can be used, linear boundaries have a clear computational advantage, and lend themselves to a concise description.

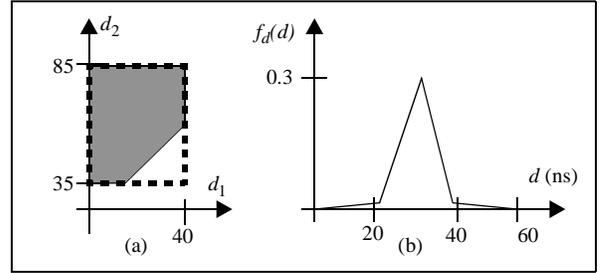


Figure 4. (a) Projection of the pdf of delays d_1 and d_2 ; (b) pdf of time separation d between two events.

The output of our verification analysis is a probability density function of the time separation between two constrained events. Thus it is possible to find out the probability that the constraint would be violated.

Consider for example the probability distribution function f_d of the time separation d between two events shown in Figure 4b. The function f_d is non-zero for $d \in [0, 60]$. If there is a constraint $\Delta = [20, 40]$ between the events, conventional verification techniques can only determine that the constraint is violated by 20 ns on each side. A *reliability* analysis can determine that Δ is violated in, say, only 1% of the cases. This information can be invaluable to designers.

4. Probabilistic interface timing verification

To represent circuit delays we use a probabilistic signal transition graph, which is a Petri net whose transitions are interpreted as events of the interfacing protocols. A set of constraint rules on selected transitions of the net specifies the timing constraints. Firstly we introduce a probabilistic Petri net which can accommodate random variables with arbitrary probability functions.

4.1 Stochastic Petri net mode

A probabilistic Petri net is a quintuple $N = \langle P, T, F, M_0, \Gamma \rangle$ where P is a non-empty set of places, T is a non-empty set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation, $M: P \rightarrow \mathfrak{N}$ is the marking function (where \mathfrak{N} is the set of non-negative integers), and $\Gamma: P \rightarrow \tau$ is the delay labeling function that assigns to each place $p_i \in P$ a random variable (r.v.) $\tau(p_i)$ [5].

The preset (postset) of a transition t is the set of incoming places to (outgoing places from) t and is denoted $\bullet t$ ($t \bullet$). Random variables τ_i 's are used to represent circuit delays:

Firing rule.

1. A transition $t \in T$ is enabled when every place $p \in \bullet t$ contains a visible token.

2. An enabled transition fires immediately. When it fires, the transition consumes the tokens in every $p \in \bullet t$ and sends tokens to every place $p \in t \bullet$.

3. A place p upon receiving a token at time τ makes it visible to transitions $t \in p \bullet$ at time $\tau + \tau_i$, where τ_i is a random variable with distribution $f_{\tau_i}(\tau_i)$.

For causality it is required that the probability of any random variable τ_i taking a negative value be zero. The set of random variables τ_i , $i = 1..M$, assigned to the places of the net are fully described by the probability density function (in short pdf) $f_{\tau_1 \dots \tau_M}(\tau_1, \dots, \tau_M)$. In practice, most of the r.v. are independent, so that f can have a more compact form. For example, if all τ_i are independent then:

$$f_{\tau_1 \dots \tau_M}(\tau_1, \dots, \tau_M) = f_{\tau_1}(\tau_1) \dots f_{\tau_M}(\tau_M) \quad [\text{Eq. 1}]$$

A constraint rule associated to net $N = \langle P, T, F, M_0, \Gamma \rangle$ is a triple $c_{ij} = \langle t_i, t_j, \Delta_{ij} \rangle$ where $t_i, t_j \in T$ is a pair of transitions of the net, and Δ_{ij} is a compact real interval. The set of constraint rules $C_N = \{c_{ij}\}$ specifies the timing constraints of net N .

A probabilistic STG is defined as a tuple $\langle N, C_N, Y, \lambda \rangle$ where N is a probabilistic Petri net, C_N is the net's associated set of constraint rules, Y is a set of signals, and $\lambda: T \rightarrow E$ is a labelling function which assigns transitions of N with events $e \in Y \times D \times D$ (D is the signal domain).

Definition.- A probabilistic STG is said to be time-consistent if for every constraint rule $c_{ij} \in C_N$, the time separation between events t_i and t_j is within the bounds of Δ_{ij} over all possible executions of the STG.

Determining if an STG is time-consistent for an arbitrary net topology is an open problem. In this paper we solve the time-consistency problem for a sub-class of Petri nets with AND and OR causality.

4.2 AND and OR causality

An event t is said to be AND-caused by a set S of events if t

occurs after all the events in S in all occurrence sequences. Similarly, an event t is said to be OR-caused by a set S of events if t occurs after the first event in S occurs. For example in Figure 5a event c occurs only after a and b occur, and in Figure 5b c occurs as soon as a or b occurs. Figures 5c and d show the corresponding Petri net constructs. For the sake of clarity from now on we adopt the more compact representation shown in Figures 5a and b.

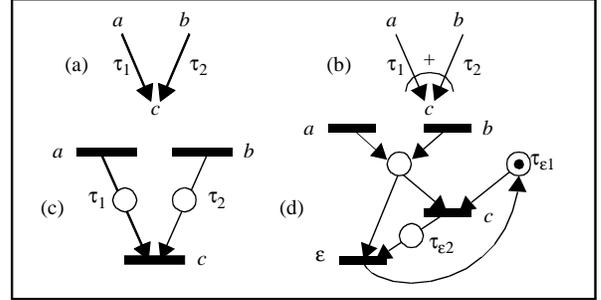


Figure 5. Causality classes: (a) AND causality; (b) OR causality; (c) and (d) Petri net constructs.

Let us consider the multiple AND join shown in Figure 6. Places are associated with random variables τ_i with distributions $f_{\tau_i}(\tau_i)$. After the firing of an event, say a , at time τ_a , a token is made visible to event d at time $\tau_a + \tau_1$, with pdf $f_{\tau_1}(\tau_1)$. According to the firing rule, d fires when all tokens in a , b , and c are visible, which occurs at:

$$\tau_d = \max(\tau_a + \tau_1, \tau_b + \tau_2, \tau_c + \tau_3) \quad [\text{Eq. 2}]$$

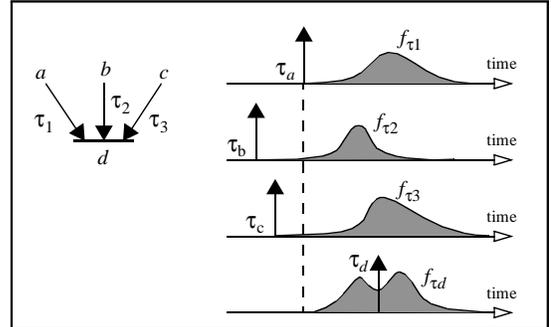


Figure 6. Distribution of the firing time of an event.

Similarly for a multiple OR join, event d will fire as soon as the first of a or b or c occurs, which occurs at:

$$\tau_d = \min(\tau_a + \tau_1, \tau_b + \tau_2, \tau_c + \tau_3) \quad [\text{Eq. 3}]$$

Thus AND/OR causality generates the max/min delay equations respectively of interface timing verification.

4.3 Functions of random variables

In this section we state some results from probability theory needed to compute time separation between events. Let x and y be two random variables with joint pdf $f_{xy}(x,y)$. The pdf of the random variable z is given by [5]:

$$f_z(z) = \int_{-\infty}^{\infty} f_{xy}(z-y, y) dy \quad \text{if } z = x + y \quad [\text{Eq. 4}]$$

$$f_z(z) = \int_{-\infty}^{\infty} f_{xy}(y+z, y) dy \quad \text{if } z = x - y \quad [\text{Eq. 5}]$$

$$f_z(z) = \frac{d}{dz} F_{xy}(z, z) \quad \text{if } z = \max(x, y) \text{ [Eq. 6]}$$

$$f_z(z) = \frac{d}{dz} (F_x(z) + F_y(z) - F_{xy}(z, z)) \quad \text{if } z = \min(x, y) \text{ [Eq. 7]}$$

where $F_{xy}(x, y)$ is the cumulative joint distribution function of x and y .

4.4 Verifying time-consistency

The timing verification procedure for a periodic STG with AND and OR causality can be stated as follows:

1. For every constraint rule c_i with associated interval Δ_{ij} from event t_i to event t_j , find the pdf $f_z(z)$ of the constraint equation $z = \tau_{ij} - \tau_{ti}$ (using Eqs. 4-7 and Eq. 9).
2. The STG is time-consistent if

$$I_i \equiv \int_{\Delta_{ij}} f_z(z) dz = 1 \quad \text{[Eq. 8]}$$

for every constraint rule c_i .

In [6] it is shown how to write constraint equations for a periodic STG with AND and OR causality. Constraint equations only contain the operators described in section 4.3. To determine the joint pdf of a constraint equation we use the following procedure based on point conditional probability [5]. Let $f_{ba}(b, a)$ be the joint pdf of two r.v. a and b , which are functions of two vectors of r.v. τ_a and τ_b respectively, where the elements of τ_a and τ_b are subsets of the set of r.v. τ_i . Denote $a = a(\tau_a)$, $b = b(\tau_b)$, and τ_\cap the vector of r.v. τ_i common to both τ_a and τ_b . The distribution of $f_{ba|\tau_\cap}(b, a, \tau_\cap)$ is just $f_{b|\tau_\cap}(b, \tau_\cap) f_{a|\tau_\cap}(a, \tau_\cap)$ for a given fixed τ_\cap . Furthermore, the joint pdf of a and b is given by:

$$f_{ba}(b, a) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f_{ba\tau_\cap}(b, a, \tau_\cap) d\tau_\cap \quad \text{[Eq. 9]}$$

where

$$f_{ba\tau_\cap}(b, a, \tau_\cap) = f_{ba|\tau_\cap}(b, a, \tau_\cap) f_{\tau_\cap}(\tau_\cap) \quad \text{[Eq. 10]}$$

5. Example

Consider for example the partial unfolded graph [6] shown in Figure 7. Event d will occur as soon as the first of c or d occurs (OR causality). Suppose the delays are as follows: τ_1 and τ_2 in $[0, 20]$; τ_3 in $[10, 50]$; τ_4 in $[0, 60]$; and τ_5 in $[10, 30]$. Moreover τ_1 and τ_2 are correlated by ρ_1 which is $[-5, 5]$. Correlation ρ_1 states that for all possible values of τ_1 and τ_2 , $\tau_1 - \tau_2 \in \rho_1$, or $-5 \leq \tau_1 - \tau_2 \leq 5$. Note that non-causality is not implied by ρ_1 , as events b and c always occur after event a . In the absence of additional probabilistic data, we assume that the pdf's are uniformly distributed.

The joint probability distribution $f_{\tau_1\tau_2}(\tau_1, \tau_2)$ is shown in Figure 4. To check constraint Δ one must find the time separation $z = \tau_d - \tau_e$ from e to d , where $\tau_d = \min(\tau_1 + \tau_3, \tau_2 + \tau_4)$, and $\tau_e = \tau_2 + \tau_5$, relative to the fork event a . Figure 8 shows the pdf of z which was obtained using the verification procedure stated in the previous section.

The bounds on z are given by $[-45, 30]$. Therefore any constraint Δ such that $[-45, 30] \subseteq \Delta$ would be satisfied.

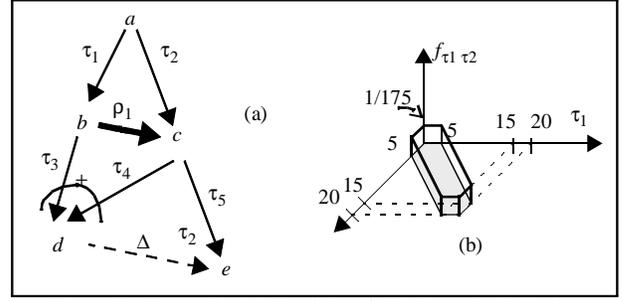


Figure 7. Example: (a) Partial unfolded graph; (b) joint probability density function of delays τ_1 and τ_2 .

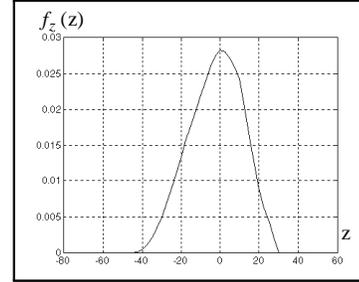


Figure 8. Pdf of $z = \tau_d - \tau_e$ with correlation.

Suppose that $\Delta = [-30, 30]$; then using Eq. 8, $I = 0.9703$, that is Δ would be violated in about 3% of the executions. If correlation ρ_1 is not taken into consideration, by applying conventional timing verification techniques which can handle *min* delays (e.g. [2, 1]) it can be found that the bounds on z are $[-50, 40]$ thus yielding pessimistic results.

6. Conclusions

In this paper we considered the timing verification of digital interface circuits. Previous research has implicitly assumed that circuit delays are independent of one another. However in the timing diagrams provided by manufacturers it is common to find timing correlation information. If correlation information is not taken into account the verification procedure may produce pessimistic results. In order to cope with correlation data we have proposed a probabilistic timing verification which can handle AND/OR causality. In unbounded delay situations (e.g. due to metastable condition), it is unrealistic to demand 100% constraint satisfaction. As future work we are planning to extend the constraint rules to include a reliability factor that expresses less strict conditions on the constraint satisfaction.

References

- [1] K. L. McMillan and D. L. Dill, "Algorithms for interface timing verification," *Proc. ICCD*, pp. 48–51, 1992.
- [2] T. M. Burks and K. Sakallah, "Min-max linear programming and the timing analysis of digital circuits", *Proc. ICCD*, pp. 152–155, 1993.
- [3] T.-Y. Yen, A. Ishii, A. Casavant, and W. Wolf, "Efficient algorithms for interface timing verification", *Proc. EuroDac '94*, pp. 34–39, 1994.
- [4] A. V. Yakovlev, "On limitations and extensions of STG model for asynchronous control circuits," *Proc. ICCD*, pp. 396–400, 1992.
- [5] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd. edition. McGraw-Hill, 1984.
- [6] M. A. Escalante *et. al.*, "Timing analysis for synthesis of hardware interface controllers using timed signal transition graphs", *Proc. of the 8th Intl Workshop on Petri nets and Performance Models*, pp. 232–240, 1995.