

## Timing Verification of Sequential Domino Circuits

David Van Campenhout, Trevor Mudge, and Karem A. Sakallah

EECS Department, University of Michigan

Ann Arbor, Michigan 48109-2122

{davidvc, trm, karem}@eecs.umich.edu

*Abstract — Two methods are presented for static timing verification of sequential circuits implemented as a mix of static and domino logic. Constraints for proper operation of domino gates are derived. An important observation is that input signals to domino gates may start changing near the end of the evaluate phase. The first method models domino gates explicitly, similar to latches. The second method treats domino gates only during pre- and post-processing steps. This method is shown to be more conservative, but easier to compute.*

### 1 Introduction

High-performance microprocessors use various circuit techniques to achieve high clock frequencies. Particularly popular is the use of domino logic. This style of dynamic logic, first proposed by Krambeck et al. [1], has the advantage of small area, fast operation, and low power. However the use of domino logic has been restricted mainly to full custom designs, in part because of the difficulty of verification. Not only do electrical effects such as charge sharing need to be verified, the timing of the circuits is also critical. In the absence of good timing models, designers often have to depend solely on electrical simulators such as Spice to verify their designs. This paper addresses timing verification of sequential circuits consisting of both static logic and domino logic.

The characteristic timing constraint for domino gates was stated in Krambeck's paper: "*All nodes can make at most only a single (rising) transition and then must stay there until the next precharge.*" Most work in static timing analysis of sequential circuits has not considered domino logic. Venkat et al. [2], described timing verification methodology for domino circuits. The focal points of their work is the identification of dynamic nodes, constraint generation for verifying the operation of the dynamic logic gates, and handling gated clocks. However, they do not describe how domino gates are handled during the actual static timing analysis. We derive conditions for a domino gate to operate properly. We provide extensions for domino logic to a popular static timing analysis framework. We propose two methods for static timing verification of sequential circuits implemented as a mix of static and domino logic, and analyze their relationship.

### 2 Domino Logic

Figure 1 shows a domino ANDOR21 gate and some sample waveforms. The operation of the circuit is as follows. When the clock *clk* is low, the internal node *z* is precharged, and the output node *y* is set to zero. The period in which *clk* is low is called the *precharge phase*. A rising transition on the clock conditionally discharges the internal node *z* through the pulldown network. The values of the inputs determine whether the discharge actually takes place. This phase is called the *evaluate phase*. Once *z* is discharged, it will stay low for the rest of the evaluate phase no matter what values the inputs assume. Therefore, either the inputs have to settle to their stable value before the start of the evaluate phase, or they can settle to their stable value (a high value) by making a single rising transition during the evaluate phase.

The inverter at the output of the gate is included for several reasons. First, it is required for proper operation of a chain of domino gates. Second, the internal node *z* is a weak node. When the clock is high, the high value on that node is not driven. The inverting buffer separates that dynamic node from the rest of the circuit, alleviating charge-sharing problems and minimizing capacitive coupling. However, the presence of the inverting buffer means that a domino gate can only implement a non-inverting function of its inputs.

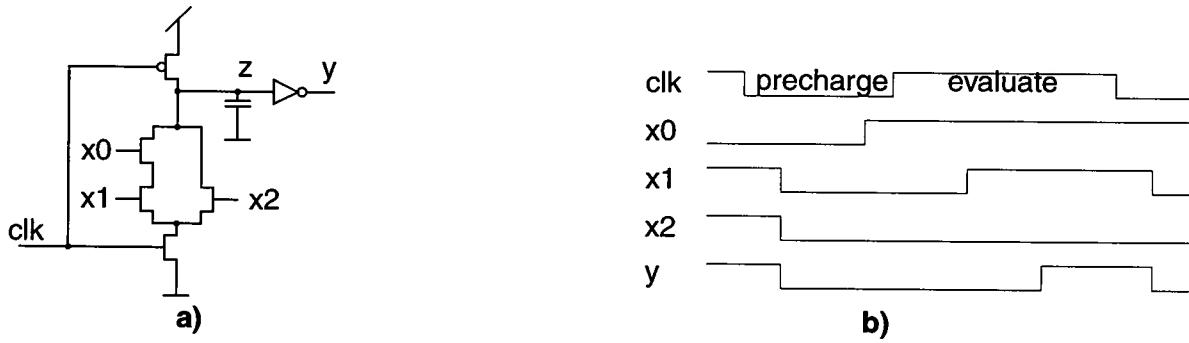


Figure 1: Domino ANDOR21 gate: a) transistor-level schematic b) sample waveforms

### 3 Static Timing Analysis Context and Domino Constraints

A comprehensive model for analyzing the temporal behavior of synchronous sequential circuits, the SMO model, is described in [3]. These sequential circuits are composed of an interconnection of static combinational logic and synchronizers. The synchronizers can either be level-sensitive (latches), or edge-triggered (flip-flops). The SMO model assumes a multi-phase clocking system with common clock period  $T_c$ . The combinational logic between each pair of synchronizers is characterized by the minimum and the maximum propagation delays. Each synchronizer is characterized by its setup time and hold time, the minimum and maximum skew of its clock signal, the phase of this signal, and the minimum and maximum delay between the data-input and the output. The time complexity for verifying the timing of a circuit with  $|L|$  latches and  $|e|$  combinational edges connecting the latches is  $O(|L||e|)$ .

A key feature of this model is that the analysis is performed modulo  $T_c$ . A reference clock cycle is associated with each synchronizer  $i$ . During the reference cycle, the data input to the latch undergoes the following sequence. First, it holds the stable value that was latched at the end of the previous reference cycle. The hold time of the synchronizer is specified as a minimum duration. The earliest time at which the input signal may start changing is called  $a_i$  (earliest arrival time). The last change of the signal with respect to this reference cycle takes place no later than  $A_i$  (latest arrival time). From then on, the input signal assumes its stable value which will be latched at the end of this reference cycle. For the latching to happen reliably, the input signal must settle no later than a setup time before the latching edge of the clock. Hence, in the SMO model signal waveforms are modeled with two events. A sample waveform is shown in Figure 2a. The stable values in each cycle are labeled  $w_i$ .

From the discussion on the operation of domino gates, it is clear that the two-event signal model is insufficient. Four event times can be used to describe input and output signals of synchronizers and domino gates. These events are the earliest rising transition, the earliest falling transition, the latest rising transition and the latest falling transition. The corresponding event times are denoted by  $a^R, a^F, A^R, A^F$  respectively. The complete waveform is periodic

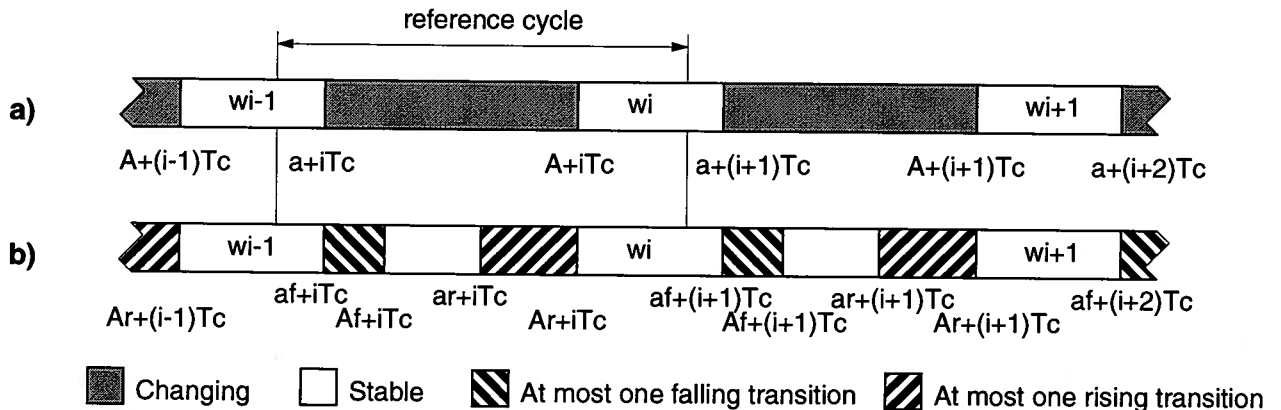


Figure 2: a) A periodic 2-event waveform; b) a periodic 4-event waveform

with period the clock period  $T_c$ . A waveform that is typical for the output signals of domino gates is shown in Figure 2b. The stable values of the signal over the consecutive cycles are denoted by  $w_i$ . As in the SMO model, no information about the relationship of stable values of signals is retained. In this sense the model makes a complete abstraction of the functional aspect of the circuit. Hence, input signals are considered completely independent signals. Given that the input signals to a domino gate are characterized by four-event periodic waveforms, the constraints for proper operation of the gate can be derived, and so can the four-event representation of the output waveform.

We say that a domino gate is operating properly if and only if the stable value of the output of the gate is determined solely by the stable values of its inputs. Given that all signals are abstracted to independent, four-event periodic waveforms, necessary and sufficient conditions for proper operation of a domino gate can be derived. Referring to Figure 3, these conditions are:

1. the latest rising transition on each input must occur one setup-time  $S^F$  before the end of the evaluate phase;
2. the latest falling transition on any input must occur one setup time  $S^R$  before the start of the evaluate phase;
3. all stable values must be held at least one hold time  $H$  after the latest rising transition on all inputs, including the clock input;
4. the positive pulse width of the clock input is at least  $T_{eval}$ ;
5. the negative pulse width of the clock input is at least  $T_{prech}$ .

The parameters  $S^F$ ,  $S^R$ ,  $H$ ,  $T_{prech}$ , and  $T_{eval}$  reflect the finite slopes of the signal transitions, and a safety margin. It is important to note that the hold constraint is with respect to either the rising edge of the clock, or to the latest rising transition among the input. There is no need to hold the inputs at their stable values throughout the evaluate phase as one might expect. Only synchronizers store the state of the sequential circuit, domino gates do not store state.

## 4 Verification Methods

### 4.1 Method 1: explicit modeling of domino gates

In this approach, domino gates are modeled explicitly, similar to latches. Data-input signals of synchronizers and domino gates are described by periodic four-event waveforms. The output signal is expressed as a function of the input signals. Input signals are expressed in terms of output signals, using the minimum and maximum combinational delays between these signals. The system of SMO equations is augmented with extra equations for each domino gate. A disadvantage of this method is that the number of variables can drastically increase for domino-rich circuits, resulting in longer computation times.

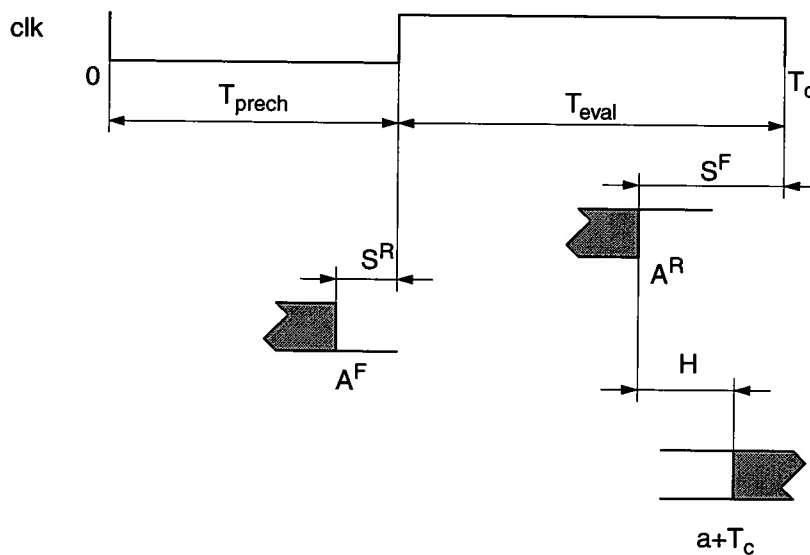


Figure 3: Constraints for a domino gate

## 4.2 Method 2: implicit handling of domino gates

This method has three steps:

1. preprocessing: computation of combinational delays
2. SMO timing analysis
3. postprocessing: verification of all timing constraints associated with domino gates

During a preprocessing phase the combinational delay between each connected pair of latches is computed. In contrast to method 1, these paths may cross domino gates (only rising transitions propagate through domino gates). The presence of domino gates necessitates the consideration of an additional type of path. These paths start at a transition (rising or falling) of a primary clock phase, entering a first domino gate through the clock input. Once the combinational delays are known, the timing verification reduces to the basic problem addressed by the SMO model. Note that no extra variables corresponding to events on domino gates are required as in method 1, resulting in a smaller computational cost. After step 2, the departure times of the output signals of all synchronizers are known. In the last step, the arrival times of the input signals of all domino gates are computed, and the constraints associated with domino gates are checked. This can be done with a single traversal of the circuit.

It can be shown that the feasible region defined by the system of late-signal constraints is the same for both methods. However, the feasible region defined by the system of early-signal constraints derived by method 2 is a subset of that derived by method 1. In other words, method 2 might produce a more conservative system of constraints, and hence may produce false negatives where method 1 would not.

## 4.3 Domino verification revisited

Design style is often more a political issue than an engineering one. Some design styles might overconstrain the design space, resulting in a more conservative design. One interesting, and not uncommon, instance in domino design imposes a more stringent hold constraint on domino gates. That constraint stipulates that input signals to domino gates must not change earlier than a hold time after the beginning of the precharge phase. This verification problem will be referred to as the *conservative verification problem*. Method 1 can be readily modified to solve this problem. For method 2, the modification requires that the paths going through domino gates via the data-inputs be ignored during short path computation. The only paths containing domino gates that are considered during short path computation start at a transition of a primary clock, triggers the output of a domino gate, and proceeds through all-static logic. It can be shown that for the conservative verification problem, both methods derive identical results.

## 5 Conclusion

We addressed static timing verification for sequential circuits implemented as a mix of static and domino logic. The constraints for proper operation of domino gates were derived. An important observation is that input signals to domino gates may start changing near the end of the evaluate phase. This gives the circuit designer extra flexibility. Two verification methods were presented. Both are based on the SMO model for static timing analysis of sequential circuits. The first method models domino gates explicitly. The signals at the terminals of the domino gates are modeled by four events: the earliest/latest rising/falling transition. The second method applies the original SMO model after a preprocessing step that computes the combinational delays. A postprocessing step checks the domino-specific constraints. The relationship between both methods was studied. We showed that the second method may result in a more conservative analysis than the first method, but it has at a smaller computational cost.

## References

- [1] R. H. Krambeck, Charles M. Lee, and Hung-Fai Stephen Law, "High-Speed Compact Circuits with CMOS," *IEEE Journal of Solid-State Circuits*; Vol. 17, No. 3, p 614-619, June 1982.
- [2] K. Venkat et al, "Timing verification of dynamic circuits," *Proceedings of the Custom Integrated Circuits Conference 1995*, p271-274.
- [3] K. Sakallah, T. Mudge, O. Olukotun, "Analysis and Design of Latch-Controlled Synchronous Digital Circuits," *IEEE Trans. on Computer-Aided Design*, Vol. 11, No. 3, pp. 322-333, March 1992.