IEEE/ACM INTERNATIONAL CONFERENCE
ON COMPUTER AIDED DESIGN

ICCAD 96

A CONFERENCE FOR THE EE CAD PROFESSIONAL

IEEE/ACM DIGEST OF TECHNICAL PAPERS

NOVEMBER 10-14, 1996

RED LION HOTEL

SAN JOSE, CALIFORNIA

Sponsored by: IEEE COMPUTER SOCIETY
50 YEARS OF SERVICE •1946-1996

Technical Committee on Design Automation®

sigda
special interest group on design automation

ASSOCIATION FOR COMPUTING MACHINERY
Special Interest Group on Design Automation

In cooperation with: ELECTRON DEVICES SOCIETY
THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS, INC.
ELECTRON DEVICES SOCIETY

IEEE Computer Society Press

The Institute of Electrical and Electronics Engineers, Inc.

Timing Verification of Sequential Domino Circuits

David Van Campenhout, Trevor Mudge, and Karem A. Sakallah
Advanced Computer Architecture Laboratory
EECS Department, University of Michigan
Ann Arbor, Michigan 48109-2122
{davidvc, tnm, 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

Domino logic is popular for high-performance microprocessors where high clock frequencies are required. Domino logic, a form of dynamic logic, has the advantage of small area, fast operation, and low power [1]. 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 the 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 [1]: *"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 are 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 provide extensions for domino logic to a popular static timing analysis framework [3,4,5]. We propose two methods for static timing verification of sequential circuits implemented as a mix of static and domino logic, and analyze their relationship.

The next section summarizes the SMO model for static timing analysis of sequential circuits. The operation of domino logic is described in Section 3. In Section 4, the timing behavior of domino gates is analyzed in detail and constraints for proper operation are derived. The actual verification methods are described in Section 5. A detailed example illustrating the model's important features is provided in the Section 6. Concluding remarks follow.

2 Static timing analysis

A comprehensive model for analyzing the temporal behavior of synchronous sequential circuits, the SMO model, is described in [4]. 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

1063-6757/96 $5.00 © 1996 IEEE

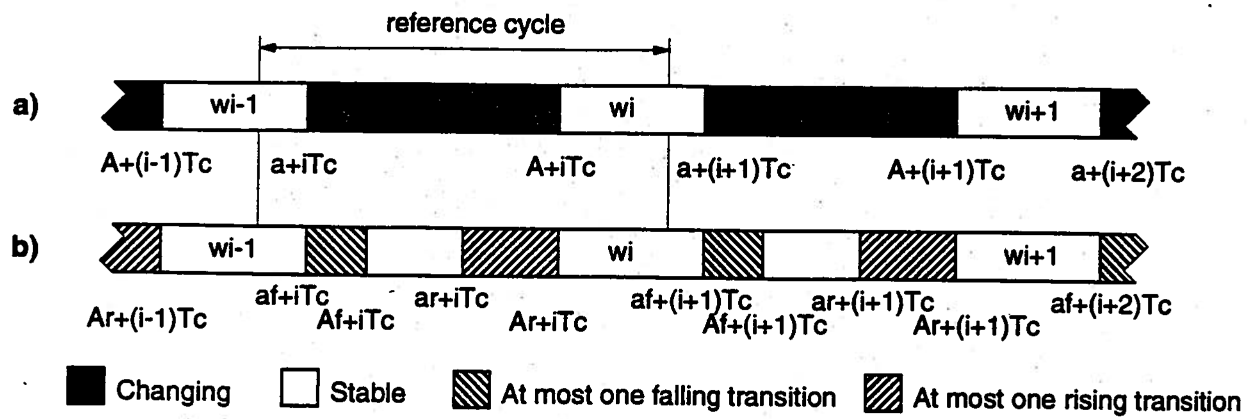

Figure 1: b) A periodic 4-event waveform; a) corresponding periodic 2-event waveform

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 1a. The stable values in each cycle are labeled w_i.

3 Domino logic

Figure 2 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.

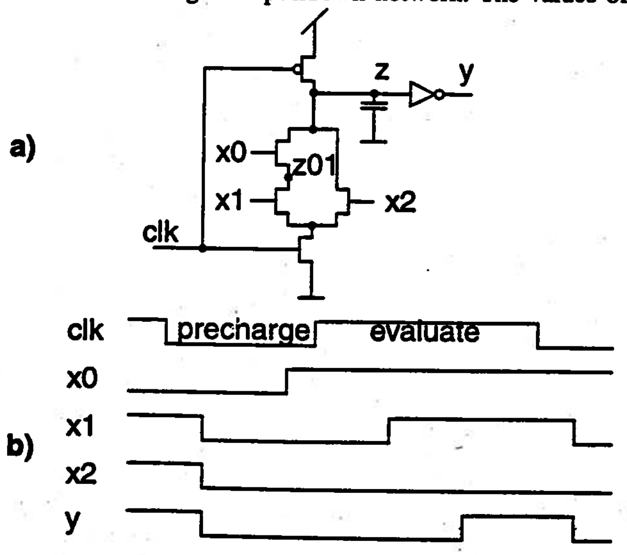

Figure 2: Domino ANDOR21 gate: a) transistor-level schematic b) sample waveforms

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, thus alleviating charge-sharing problems and minimizing capacitive coupling. A consequence of the inverting buffer is that a domino gate can only implement a non-inverting function of its inputs. The dual circuit of that shown in Figure 2 is also a valid domino gate. However, in many CMOS technologies, the performance of such a dual gate is far inferior to that of the original gate due to the poor characteristics of the pMOS transistors. The following discussion will only be concerned with the type of gate shown in Figure 2.

4 Domino Constraints

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 with period the clock period T_c. A waveform that is typical for the output signals of domino gates is shown in Figure 1b. 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 must be at least T_{eval};
5. the negative pulse width of the clock input must be 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 inputs. 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.

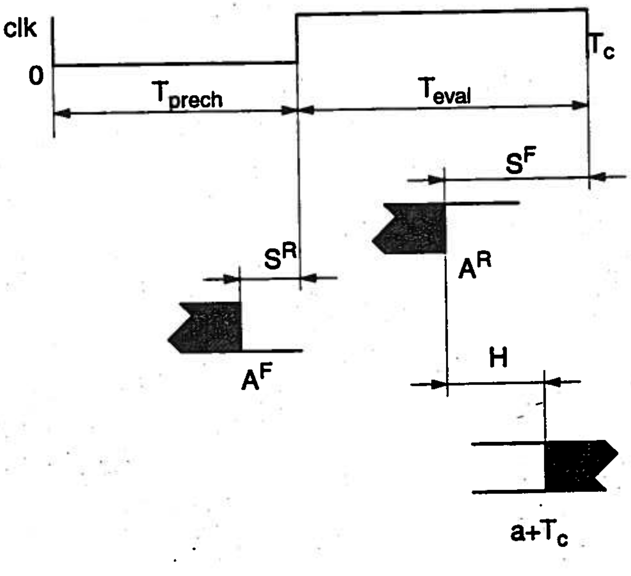

Figure 3: Constraints for a domino gate

4.1 Gate delay model

We assume that for each gate a set of min/max, input to output delays are available for each input/output pair. The minimum delay from a transition of type T_i (either rising or falling) on input i that triggers a transition of type T_y on output y is denoted by $\delta_{i,y}^{T_iT_y}$. Similarly for the maximum delay $\Delta_{i,y}^{T_iT_y}$. Depending on the gate type not all combinations apply. For example, a 2-input domino AND-gate is characterized by the set $\{\delta_{a,y}^{RR}, \delta_{b,y}^{RR}, \delta_{clk,y}^{RR}, \delta_{clk,y}^{FF}, \Delta_{a,y}^{RR}, \Delta_{b,y}^{RR}, \Delta_{clk,y}^{RR}, \Delta_{clk,y}^{FF}\}$. For a static XOR-gate, any transition on the output can be triggered by any transition on an input. Hence all transition-pairs (T_i, T_y) apply. Note that the differentiation between minimum and maximum delays reflects the differences in delay due to different conditions of the side inputs. This can be illustrated with the domino ANDOR21-gate in Figure 2. Consider the propagation of a rising event on input x_2 to the output. In order for the event to propagate, the clock *clk* has to be high and at least one of the other data inputs needs to be low. In case both x_0 and x_1 are low, the only capacitance that needs to be discharged is that at internal node z. The same is true in case x_1 is low and x_0 high. In this case the capacitance at node z_{01} was already discharged before x_2 rises. However when x_1 is high and x_0 low, the capacitance at node z_{01} needs to be discharged through the x_2 transistor as well. This results in a larger delay. For complex gates this effect can be significant. The dependence of propagation delay on the input conditions is certainly not unique to domino gates, but it tends to be more significant in domino gates. The example also suggests that in general, the delay from a transition on input x_i to output y can be dependent on the logic value of the sideinputs. However, taking into account this dependency greatly complicates the timing analysis.

5 Timing Verification

For timing verification we propose two approaches. The first approach extends the SMO model by explicitly modeling the domino gates. The alternative approach makes use of the basic SMO model. Domino gates are taken into account during combinational delay computation and in a postprocessing step which checks domino-specific constraints.

5.1 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 signals of synchronizers and domino gates are expressed as a function of their corresponding input signals. The input signals in turn are expressed in terms of output signals of the predecessor synchronizers and domino gates, 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.

5.2 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 feasibly 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. Referring to Figure 4, this can be seen as follows. The figure shows a domino gate connected to a flip-flop through some combinational logic whose minimum delays are δ^R and δ^F starting from a rising and falling transition respectively. The early arrival of *u* is determined by three timing paths: *<Rx0, Ry, u>*, *<Fclk1, Fy, u>*, and *<Rclk1, Ry, u>*. The symbols *R* and *F* refer to a rising and falling transition respectively. The first path needs indeed be considered since we allow input signal to domino gates to start changing near the end of the evaluate phase. Normally, the third path will not contribute to the result since the difference between δ^R and δ^F is usually much smaller than the duration of the precharge phase. Now consider the case in which the earliest arrival of *x0* arrives δx (smaller than the duration of the precharge phase) after the falling transition of *clk1*. In this case path1 is false. If in addition $\delta^R + \delta x < \delta^F$, method 2 will compute a smaller value for the early arrival than 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.

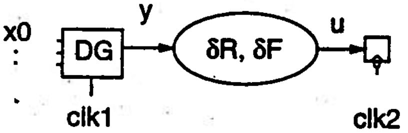

Figure 4: Short path situation

5.3 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.

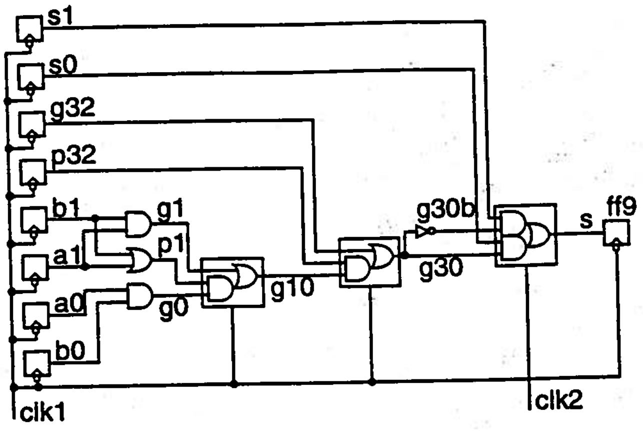

Figure 5: Example circuit

Table 1: Min/max gate delays: δ:Δ [ps]

gate	input	RR	FR	FF	RF
INV	x0	N/A	50:50	N/A	15:15
AND2	x0,x1	75:100	N/A	50:75	N/A
OR2	x0,x1	50:75	N/A	100:125	N/A
ANDOR21	x0	80:90	N/A	N/A	N/A
	x1,x2	90:100	N/A	N/A	N/A
	clk	75:100	N/A	100:125	N/A
ANDOR22	x0,x2	90:100	N/A	N/A	N/A
	x1,x3	80:100	N/A	N/A	N/A
	clk	80:115	N/A	80:110	N/A
DFFN	clk	N/A	150:150	125:125	N/A

Table2: Setup/Hold[ps]

Parameter	DFFN	ANDOR21 ANDOR22
Tsetup	150	N/A
Thold	0	50
Tsetup1	N/A	50
Tsetup2	N/A	50
Tprecharge	N/A	200
Tevaluate	N/A	200

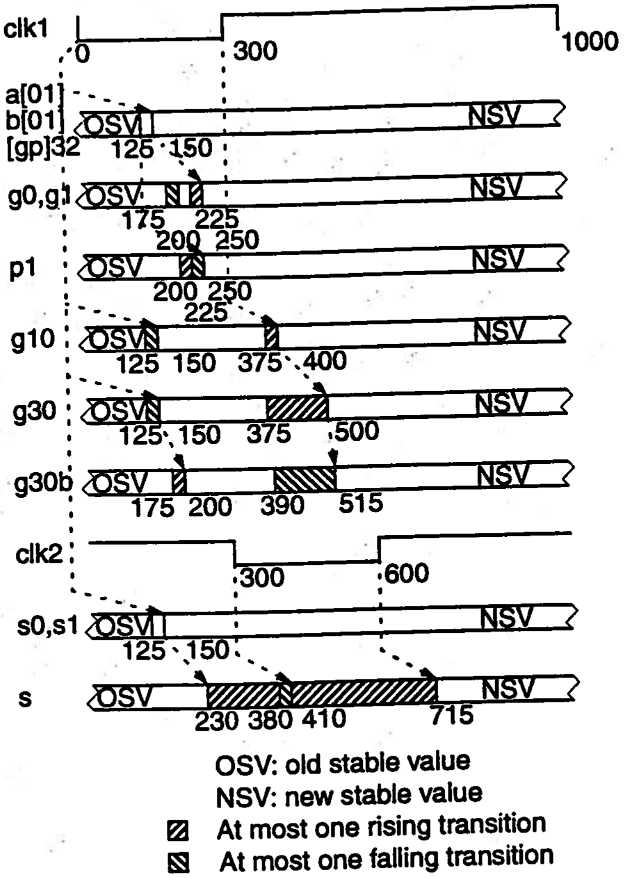

Figure 6: Waveforms for example (times in ps)

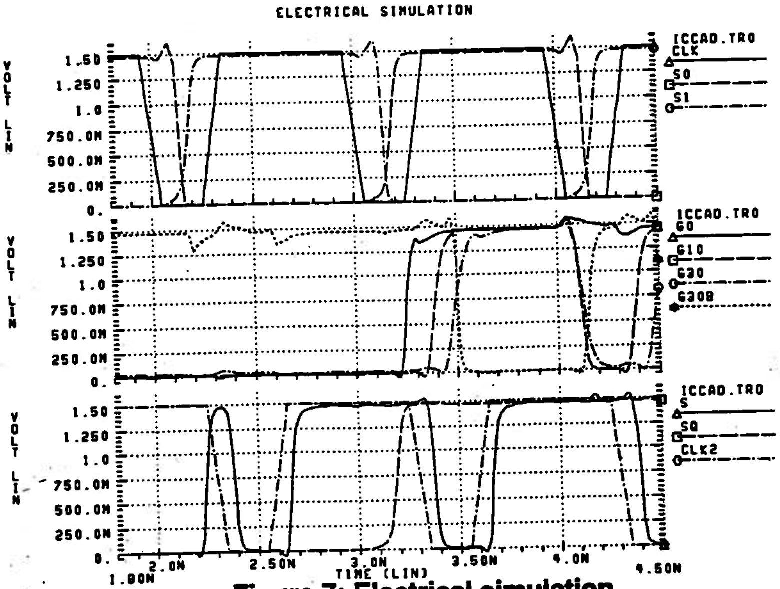

Figure 7: Electrical simulation

6 Example

An example circuit is shown in Figure 5. Gate delays are listed in Table 1, setup and hold parameters in Table 2. The waveforms exhibited on each circuit node are shown in Figure 6. Causality is indicated by the arrows. The synchronizers at the boundaries of the circuit are negative-edge triggered, and are clocked by the same phase. One clock cycle is available to propagate the signals between the synchronizers. The logic in between the synchronizers consists of both static and domino logic. The domino ANDOR22 gate driving s is of special interest. This gate implements a multiplexer. However it requires both the complemented and the uncomplemented version of the select signal (*g30*), since domino gates can only implement non-inverting functions. Given that the uncomplemented select signal is generated by domino logic, there are two alternatives for generating the complemented version. The first one is to produce the complemented signal by a separate chain of domino logic. This effectively drives back the inversion to the synchronizers. In this case either the complemented or the uncomplemented select signal will stay low throughout a clock cycle, whereas the other signal makes a single rising transition during the evaluate phase. The advantage of this implementation is that the ANDOR22 gate can be clocked by the same phase as the other domino gates. The area overhead, however, can be significant and might motivate the second alternative adopted in the example. A static inverter produces the complemented select signal. In order to prevent a falling transition on *g30b* to cause incorrect operation of the circuit, another clock phase, *clk2*, is used to clock the ANDOR22 gate. With respect to this clock phase, the select signals, *g30* and *g30b*, settle well before the onset of the evaluate phase.

The example also illustrates that early signal changes pertaining the next clock cycle, might arrive during the evaluate phase. The conservative domino rules (see Section 5.3) would not allow this behavior. The input signals *s0* and *s1* assumes their new value at $t = 150$. At that time the clock steering the ANDOR22, *clk2*, is still high. Hence, if the ANDOR22-gate evaluated to a zero, the early arrival of *s0* might cause the zero to be overwritten by a one. As can be seen from the waveforms, the output s of the ANDOR22 is affected at $t = 230$. This is not a problem provided that the stable value of s propagates to the synchronizers. This is the case as at $t = 1000$, the stable value of s, is captured by the output synchronizer. This is well before the early arrival of *s0* corrupts s. According to the conservative domino rules we would be forced to make the falling edge of *clk2* occur earlier so that the early arrival of *s0* occurs during the precharge phase. This requires more complex circuitry. In our case *clk2* can simply be generated by delaying *clk2*

To demonstrate that these results are realistic, we also performed an electrical simulation of the circuit from Figure 5. The actual circuit was implemented in a complementary gallium-arsenide technology. Some representative waveforms are shown in Figure 7. Note that at $2.2ns$ the ANDOR22 gate (whose output is s) switches. This is due to the effect of the early arrival of the *s1* signal as was described above.

7 Conclusion

We addressed static timing verification for sequential circuit implemented in 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 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 at a smaller computational cost.

Acknowledgments

This work was supported in part by SRC contract 95-DJ-338, ARPA contract DAAH04-94-G0327, and NSF grant MIP-9404632.

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, June 1982. p 614-619; 1982.

[2] K. Venkat et al, "Timing verification of dynamic circuits," *Proceedings of the Custom Integrated Circuits Conference* 1995. p271-274; 1995.

[3] K. Sakallah, T. Mudge, O. Olukotun, "checkTc and minTc: Timing Verification and Optimal Clocking of Synchronous Digital Circuits," in *ICCAD-90 Digest of Technical Papers*, pp. 552-555, November 1990.

[4] 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.

[5] T. Burks, K. Sakallah and T. Mudge, "Critical Paths in Circuits with Level-Sensitive Latches," *IEEE Trans. VLSI Systems*, Vol. 3, No. 2, pp. 273-291, June 1995.