Timing Verification of Dynamic Circuits

Kumar Venkat, Liang Chen, Ichiang Lin, Piyush Mistry, and Pravin Madhani

***Abstract*— A complete set of rules is presented for timing verification of domino-style dynamic circuits. These rules include identification of dynamic nodes, generation of accurate timing constraints based on the operating environment of the gate and verification as an enhanced part of a complete timing verification process. This methodology has been implemented in a new static timing verifier and used to verify microprocessor circuits.**

I. Introduction

DYNAMIC circuits are used widely in custom VLSI circuits to achieve higher speed, smaller area, and potentially lower power consumption due to glitch-free operation [1]. There are also difficulties in designing and verifying this class of circuits. Dynamic circuits are highly sensitive to skew between input arrival times, capacitive loading, coupling between nodes, charge sharing, etc. Since timing and functionality are closely coupled in this design style, a design may not function correctly due to a small error in timing, independent of clock cycle time.

It then becomes important to verify that the logic will function correctly under the actual operating conditions. In this paper, the discussion will be restricted to combinational logic implemented with domino-style dynamic circuits [1], which is commonly used in microprocessor design. There are two major types of constraints that must be verified at the inputs of such dynamic gates:

1) Transition times of data edges relative to clock edges at inputs, where there are both setup-time and hold-time requirements.
2) Minimum width of clock and gated-clock pulses at inputs, factoring in any overlap of precharge and evaluate actions.

Considerable work has been done in the area of timing analysis and verification for MOS VLSI circuits [2]–[4]. An approach to identifying dynamic nodes is described in [4], and constraint generation rules for dynamic circuits has been reported in [5]. However, the actual operating environment of a dynamic gate has not been completely considered in these papers, so the constraints may not be sufficient for accurate timing verification. The operating environment includes the capacitive load driven by the gate (including effects of coupling with other nodes) and the width of input pulses used for switching the gate. Further, domino-style dynamic circuits have generally been modeled as transparent latches for verifying the generated constraints, which is insufficient for modeling constraints that are specific to each data transition edge.

The approach described in this paper is based on [6] and differs in several ways from previous approaches. A comprehensive set of algorithmic rules is used to robustly identify dynamic nodes during the processing of a circuit for timing verification. Then, the verification is performed in two steps. During the normal timing verification process, clock-gating is identified, and appropriate gated-clock pulses are propagated through the circuit along with other timing edges. At the end of this step, timing information (arrival times and waveform slopes) are stored at relevant circuit nodes. After this is completed, the available timing information at inputs and outputs of dynamic gates is used in a post-processing step to generate and verify constraints based on the operating environment of each gate. The verification model is more general than a transparent latch in order to verify the complex interactions at the inputs of dynamic gates.

Our approach to timing verification of dynamic circuits has been implemented in a new static timing verifier called Mips timing verifier (MTV) [6], which is being used to verify state-of-the-art microprocessor designs. Section II describes the rules for automatically identifying dynamic circuits, with several examples for illustration. Section III discusses how gated-clock circuits are handled within static timing analysis. Section IV details our approach to generation and verification of constraints. Section V presents experimental results using selected dynamic circuits and comparison with other verification methods that are either optimistic or pessimistic.

II. Identification of Dynamic Nodes

Dynamic or charge-storage nodes, defined as outputs of domino logic gates for the purposes of this discussion, can be identified using the following two steps:

1) If a path exists from any node to power supply (Vdd) through a PMOS transistor (or parallel PMOS transistors) such that none of the transistors is part of a fully complementary logic gate and there are no other series transistors in the path to Vdd, then that node is first classified as potentially dynamic.
2) During propagation of clock edges through the clock distribution network, if a clock signal controls the PMOS pullup transistor at a potential dynamic node, then that node is reclassified as a real dynamic node.

The first step ensures that output nodes of static CMOS clock buffers and clock gating circuits (with all complementary inputs) are not included in the set of dynamic nodes. However, this rule still allows weak "leakers" to pull up the dynamic

Manuscript received October 21, 1995; revised November 28, 1995.

K. Venkat is with the Surya Technologies, Inc., San Jose, CA 95148 USA.
L. Chen, I. Lin, P. Mistry, and P. Madhani are with the Silicon Graphics, Inc., Mips Technologies Division, Mountain View, CA 94039 USA.

Publisher Item Identifier S 0018-9200(96)02469-9.

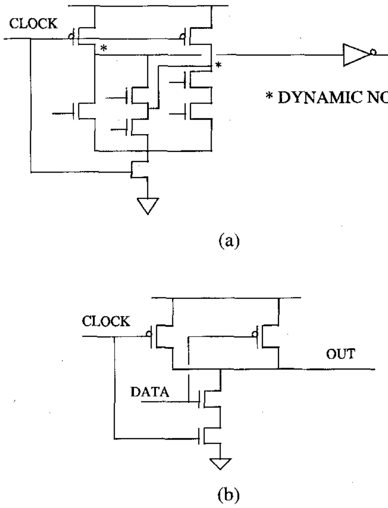

Fig. 1. (a) Domino-style dynamic gate. (b) Static clock-gating circuit.

nodes. The second step confirms that dynamic nodes are charged to Vdd only during particular clock phases.

Fig. 1(a) illustrates an example of a domino-style dynamic gate where dynamic nodes are identified by this approach. This particular gate contains a complex NMOS evaluate network and two PMOS pullup transistors. The PMOS transistor on the left is the conventional precharge device, while the other PMOS transistor serves to eliminate charge-sharing effects at internal nodes. According to the rules outlined above, two dynamic nodes are identified in this circuit (at the drain terminal of each PMOS device). Fig. 1(b) shows a static clock-gating circuit whose output node is properly excluded from the dynamic classification.

III. Clock-Gating Circuits

Gated clocks are often used to generate inputs to dynamic gates in two ways:

1) Data signals may be gated with clocks externally to drive domino gates if the evaluate networks are designed without clock-controlled NMOS transistors.
2) Clocks may be gated with delayed clocks to produce clock pulses of desired widths, so that precharge and evaluate operations can be tightly controlled.

In both cases, the resulting pulses must be verified to be of sufficient width so as to switch the dynamic gates, as will be seen in the next section. In normal critical-path timing analysis, earliest and latest arrival times (for both rising and falling edges) are propagated from one stage to the next, in order to find the longest and shortest paths. In the case of clock-gating circuits within the context of static timing analysis, the functionality of the gating logic must be considered in selecting the rising and falling edges (or equivalently, an output pulse of the correct width) to propagate.

There are some commonly used clock-gating circuits, where selecting the correct pair of rising and falling edges at the output of the gate is feasible using the following criteria:

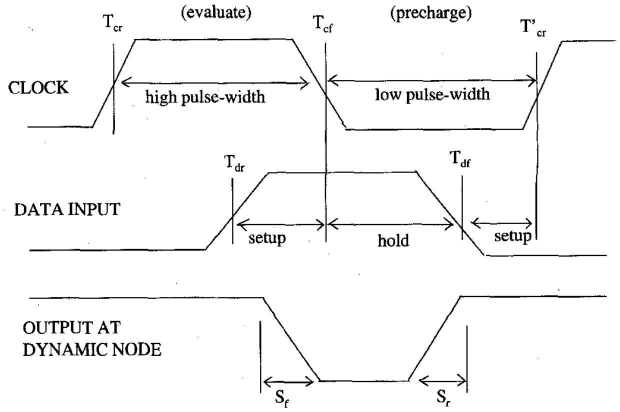

Fig. 2. Representative input/output waveforms of a domino-style dynamic gate.

1) NAND function: Since the controlling value at the input is a logic 0, the output goes to logic 0 (logic 1) when the last (first) of the inputs goes to logic 1 (logic 0). Hence, the earliest rising edge and the latest falling edge must be selected at the output to properly represent the gated-clock pulse in static analysis.
2) NOR function: Since the controlling value at the input is a logic 1, using similar reasoning as above, the earliest falling edge and the latest rising edge must be selected at the output.

IV. Timing Constraint Verification

Dynamic gates require that latest data inputs to the evaluate network arrive sufficiently early such that the load capacitance can be discharged within the evaluate clock phase. *Consideration of the load capacitance (including effects of coupling to adjacent nodes) is critical for correct constraint generation.* Further, latest data inputs should be in an inactive state prior to the beginning of the next evaluate phase, in order to prevent inadvertent discharge of the output. Both of these constraints are considered to be *setup-time requirements,* as illustrated in Fig. 2.

Further, the earliest data inputs to the evaluate network should not become inactive before the end of the evaluate phase. This constraint is a *hold-time requirement.* For clock and gated-clock inputs, the width of the signal pulse must be sufficient to allow the output to charge or discharge properly. (If the active time of a gated-clock evaluate input overlaps with some portion of the precharge time in a domino gate, then the available pulse-width for the precharge operation is effectively reduced.) This constraint is a *pulse-width requirement.* Fig. 2 also illustrates these two constraints.

All of these constraints depend heavily on accurate timing information at all relevant circuit nodes, including minimum/maximum arrival times for rising/falling edges and the corresponding rise/fall transition times. (The rise/fall transition times are also referred to as waveform slopes in this paper.) Once normal timing analysis has been performed (including propagation of gated-clock pulses) and all the critical timing information has been preserved, the dynamic gates can be

TABLE I
PULSE-WIDTH AND SETUP CONSTRAINT VERIFICATION

Circuit	Output rise/fall slope (ps)	MTV's low pulse-width/ margin (ps)	MTV's high pulse-width/ margin (ps)	MTV's setup margin (ps)	Optimistic setup margin (ps)	Pessimistic setup margin (ps)
faexpc2c1	305/115	3000/2756	--	--	--	--
faexpc2c1	3337/1413	3000/330	--	--	--	--
faexp2c2	399/150	3000/2680	3000/2880	380	500	-2500
faexp2c2	2654/970	3000/876	3000/2224	-276	500	-2500
aqdynor16b	207/230	3000/2834	3000/2816	316	500	-2500
aqdynor16b	2716/1331	3000/827	3000/1935	-564	500	-2500
aqdynor16b_gc	292/437	3000/2555	3000/122	--	--	--
aqdynor16b_gc	1220/1822	3000/1813	3000/985	--	--	--

verified as a post-processing step. Note that such timing information reflects the total effect of loading as well as device sizes throughout the circuit. This approach can be added on to most existing timing analysis systems.

Constraint Generation Rules

We now provide details on the constraint generation rules, using Fig. 2 for reference. Arrival times of clock or data edges (or events) are denoted by T, with subscripts indicating clock (c) or data (d), and rising (r) or falling (f). T' indicates a clock edge one cycle later as shown in Fig. 2. Constants are denoted by K with a suffix to distinguish different constants, and chosen for each technology or design to meet the required conservatism in the design methodology. Waveform slopes are denoted by S, with a subscript indicating rising (r) or falling (f) transition.

i) Rising edge of data input to evaluate network should meet setup-time requirement to the falling edge of evaluate clock so that the output can be discharged

$$T_{cf} - T_{dr} >= K_1 * S_f. \tag{1}$$

Note that event T_{cf} occurs chronologically later than event T_{dr} in a good design by at least the required margin on the right hand side of (1), so T_{cf} is a larger number than T_{dr} in such a case.

ii) Falling edge of data input to evaluate network should meet setup-time requirement to rising edge of evaluate clock on the next cycle

$$T'_{cr} - T_{df} >= K_2. \tag{2}$$

iii) Falling edge of data input to evaluate network should meet hold-time requirement to falling edge of evaluate clock

$$T_{df} - T_{cf} >= K_3. \tag{3}$$

iv) Effective pulse-width of precharge clock or gated-clock (conservatively reduced by the overlap time of precharge and evaluate operations, when there is no explicit evaluate clock-controlled transistor) must be capable of pulling up the output node

$$T'_{cr} - T_{cf} >= K_4 * S_r. \tag{4}$$

The time between adjacent falling and rising edges of clock is checked here.

v) Pulse-width of evaluate clock or gated-clock must be capable of pulling down the output node

$$T_{cf} - T_{cr} >= K_5 * Sf. \tag{5}$$

The time between adjacent rising and falling edges of clock is checked here.

V. RESULTS

In order to demonstrate the verification methodology, results from four different test circuits are shown in Table I: i) $faexpc2c1$ is a domino gate with no evaluate clock; ii) $faexp2c2$ and $aqdynor16b$ are standard domino circuits with evaluate clocks; iii) $aqdynor16b\_gc$ is a domino gate with gated-clock inputs and no evaluate clock.

The test conditions included a clock period of 6000 ps in all cases, with a duty cycle of 50%. Data inputs were assumed to go high 500 ps before the falling edge of clock, and go low 500 ps after the falling edge of clock. Constants K_1 and K_4 were set to 0.8 for these experiments. Each circuit was run twice, once with the dynamic nodes loaded lightly and the second time with heavier load. The output waveform slopes are longer with heavier loading, as seen in the second column of Table I.

MTV was used to verify the low pulse-width for all the precharge clocks, and high pulse-width for all evaluate clocks and gated clocks. MTV was also used to verify the setup constraint for all data inputs on circuits with explicit evaluate clock inputs. In each case reported, the "margin" is the positive or negative slack available after meeting the constraint.

The last three columns compare MTV's setup margin computation, based on (1), with two other methods that other tools have used. The optimistic margin requires only that data inputs become valid before the end of the evaluate phase. The pessimistic margin requires that data inputs become valid before the start of the evaluate phase. Such methods generally model a domino gate as a transparent latch, with all data transitions (edges) required to meet setup constraints to a particular edge of clock, which is unrealistic. In real circuits, rising transitions at data inputs may occur either before or during the evaluate phase, but they must occur before the end of the evaluate phase. Falling transitions at data inputs must occur before the start of the next evaluate phase. Therefore, MTV's setup margins are more accurate and realistic than the other two methods. (Note that MTV's hold-time and pulse-width verifications do not have similar or equivalent methods in other tools for comparison.)

VI. Conclusion

We have presented a comprehensive approach to timing verification of domino-style dynamic circuits. Dynamic nodes are automatically identified based on topology and clocking information. Then, accurate and complete constraints (setup, hold, and pulse-width) are generated for inputs at each gate based on its actual operating environment. Constraint generation and verification are performed as a post-processing step following a conventional timing analysis process. This methodology has been implemented in MTV, a new static timing verifier that is being used on microprocessor designs. The methodology is also applicable as an enhancement to most existing timing analysis systems.

References

[1] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design.* Reading, MA: Addison-Wesley, 1985.

[2] J. K. Ousterhout, "A Switch-level timing verifier for digital MOS VLSI," *IEEE Trans. Computer-Aided Design,* vol. CAD-4, no. 3, pp. 336-349, July 1985.

[3] N. P. Jouppi, "Timing analysis and performance improvement of MOS VLSI designs," *IEEE Trans. Computer-Aided Design,* vol. CAD-6, no. 4, pp. 650–665, July 1987.

[4] J. J. Cherry, "Pearl: A CMOS timing analyzer," in *Proc. 25th Design Automation Conf.,* 1988, pp. 148–153.

[5] J. Grodstein, J. Pan, B. Grundmann, B. Gieseke, and Y. Yen, "Constraint identification for timing verification," in *Int. Conf. on Computer-Aided Design,* 1990, pp. 16–19.

[6] K. Venkat, L. Chen, I. Lin, P. Mistry, P. Madhani, and K. Sato, "Timing verification of dynamic circuits," in *Proc. Custom Integrated Circuits Conf.,* 1995, pp. 271–274.