

Simulation of Digital Circuits in the Presence of Uncertainty

Mark Linderman and Miriam Leeser*

School of Electrical Engineering
Cornell University Ithaca, NY 14853

Abstract

Current extended value set dynamic timing analyzers are not sophisticated enough to detect the subtle timing relationships upon which timing-critical systems depend, and exhaustive simulation achieves very accurate results but at tremendous computational cost. MTV is a simulator that strikes a balance between accuracy and efficiency.

MTV is more accurate than other extended value set simulators because it respects the ordering of events. It is more efficient than exhaustive simulators because it efficiently simulates overlapping events and requires only a single waveform to represent a signal. Features of MTV include: elimination of common ambiguity, symbolic delays, correlated delays, and sophisticated algorithms to detect ordered events. This paper concludes with simulation results from the ISCAS85 benchmark suite.

1 Introduction

Synchronous design is used in the vast majority of digital circuits because it permits the designer to separate the issues of timing and logical functionality into two independent problems. Unfortunately, there are many circuits that do not permit this convenient abstraction; these include interface and asynchronous circuits where correct dynamic behavior is essential. If the functionality of a circuit is dependent upon the timing of its circuit elements or the timing of external stimuli, we call the circuit *timing-critical*. The dynamic timing simulator we have developed, called MTV [6], addresses problems associated with these circuits. The goal of this research is to simulate large circuits that include timing-critical elements. Therefore, it is necessary to strike a balance between accuracy and efficiency.

1.1 Related work

Scald [8] and Clover [2] use an extended logical value set comprised of True, False, X (either True or False),

Rising, Falling, Changing and Uninitialized to represent the state of a wire at a given time. These simulators are capable of simulating very large systems but are not sophisticated enough to detect the subtle timing relationships upon which timing-critical systems depend. They do not attempt to resolve ambiguous event ordering, and so they must perform a worst-case analysis where temporal and logical information degrades rapidly. This results in an efficient, but pessimistic, simulation. A second class of min/max simulators [5, 7, 1] enumerates all possible event orderings and simulates them independently. These approaches are limited to circuits that generate few overlapping events. Another exhaustive simulator, CTSS [4], cannot handle asymmetric rise and fall times and its efficiency is very dependent upon the granularity of the time step and the efficiency of its BDD encoding.

2 Simulation in MTV

Because the behavior of timing-critical circuits depends on the delays of the circuit elements, it is important to account for uncertainty in these delays due to process variation and operating conditions. MTV assigns each gate a minimum and a maximum delay that covers the range of the possible variations. This type of simulation is called a min/max simulation. We represent a min/max delay as an interval, $[m, M]$, such that $m \leq M$.

Like Scald, MTV uses an extended value set comprised of True, False, X (either True or False), Rising, Falling, and Changing to represent the state of a wire at a given time. The value Rising indicates that *it is possible* that the state of the wire will transition from False to True during the Rising period. It does not guarantee that this will happen, only that it is possible. It does guarantee that the wire cannot transition from True to False. The value Falling is defined analogously. The value Changing indicates a possible hazard.

MTV's gate model is an instantaneous logical function followed by a lumped propagation delay. A transition that occurs within an interval is assumed to occur instantaneously at an unknown time within the interval.

*This research was supported by NSF grant MIP-9111146 and NSF National Young Investigator Award CCR-9257280.

MTV employs a symbolic delay representation that uses a combination of intervals and symbolic delay variables: $[m, M] + \sum_{i=1}^n c_i d_i$ where c_i is a non-negative integer and d_i is a delay variable representing an interval. When two delays are subtracted, symbolic delays variables (and their associated uncertainties) common to both delays are cancelled out, yielding a more accurate result than a non-symbolic approach. Albeit at some loss of information, the interval allows MTV to represent non-linear functions of delays, like minimum and maximum, without requiring a bifurcation of the simulation, unlike systems that depend upon linear programming [1, 5].

True and False valued events are allowed to occur within an interval of time, we call these *Movable* events. MTV uses an event graph to represent the temporal and logical relationships between *Movable* events. All other events occur at fixed times and are called *Fixed* events. Because MTV can represent temporal uncertainty in the event graph and logical uncertainty with the states Rising, Falling or Changing, it can efficiently simulate gates with overlapping input events.

3 Phenomena detected by MTV

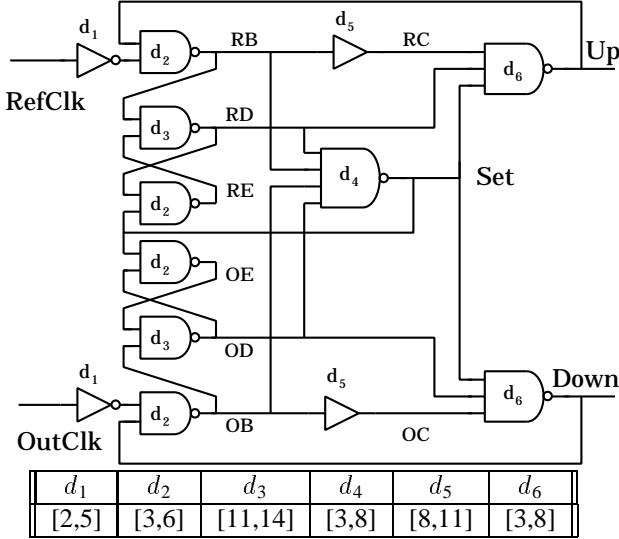


Figure 1: The phase detector

There are a number of phenomena that, if detected, can improve the accuracy of a simulation. These include: reconvergent fanout, delay correlation, and logical event ordering. This section will illustrate these phenomena using the phase detector shown in Figure 1 [3]. This is an example of a timing-critical circuit that exhibits asynchronous behavior and where correct functionality is critically dependent

on the delays of the circuit elements. We will use this circuit to illustrate important issues in simulation of circuits in the presence of temporal uncertainty. To function properly, the behavior of this circuit must be symmetric with respect to the inputs. If the rising edge of one input occurs before the rising edge of the other, then the corresponding output will go low for a period to charge or discharge the capacitor controlling the frequency of the synthesized clock. At quiescence, both outputs are True.

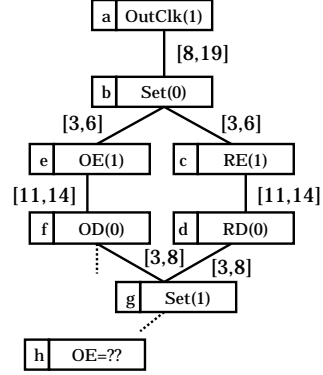


Figure 2: An event graph

Figure 2 shows a fragment of the event graph generated during the simulation of the phase detector. Each event has a letter designation, a wire name and a value. Each edge is annotated with a delay value (in nS). We denote the delay between two events x and y as $x \rightarrow y$. The delay from the event on *OutClk* to first event on *Set* (labelled b) is a summation of several gate delays not shown in this figure. The signal *OE* is the *Nand* of *OD* and *Set*. Throughout the discussion of this example, the goal is to determine what steps are necessary for the event on *OD* to precede the second event on *Set* (labelled g), because this will prevent a hazard on *OE*.

3.1 Reconvergent fanout

MTV detects reconvergent fanout and removes its associated common ambiguity. Reconvergent fanout occurs when two or more events have been caused (directly or indirectly) by the same event. When computing the separation of two events, by considering only the delays from the events back to the last common ancestor, common ambiguity due the reconvergent fanout is eliminated. This enables MTV to detect event orderings that might otherwise go undetected. The common ancestor of events f and g is event b , so the uncertainty due to the delay up to b is eliminated. Since $b \rightarrow f = [14, 20]$ and $b \rightarrow g = [17, 28]$, if a delay of $(20 - 17 = 3)$ is inserted between *OD* and *Set*, the hazard is avoided.

In Scald and Clover, the interval of an event is the accumulation of all the uncertainty leading up to that event.

It is not possible for them to attribute portions of the uncertainty to any circuit element, so common ambiguity cannot be eliminated. Since the worst-case bounds on event f is [22,39] and on g is [25,47], Scald requires an inserted delay of $(39-25=14)$. This is clearly not as desirable as a delay of 3. In fact, because of the symmetric nature of the circuit and its operation, modifying the delay of the *Nor* gate that drives *OD* will solve this hazard but exacerbate the potential hazard on *OE*. It is not possible, by modifying the delays of the given gates, for Scald to simulate this circuit as correct. It is, however, possible to simulate the circuit by inserting buffers between *OD* and *Set* and between *RD* and *Set*. According to Scald, the maximum frequency at which the circuit can operate correctly is 10.5 MHz.

3.2 Delay correlation

Because of the symmetric nature of the phase detector, the designer may “mirror” the layout (as is common in analog design) such that the delays of corresponding gates are nearly equal. To allow full or partial delay correlation, MTV allows gate delays to be expressed as symbolic delay expressions. If the delays of corresponding gates use the same symbolic expressions, MTV will detect that f occurs before g and there is no hazard. Assume that the delay of the gates driving *OE* and *RE* is Y , and the delay of the gates driving *OD* and *RD* is Z , then the delay from b to f is $Y+Z$, and the delay from b to g is $Y+Z+[3,8]$. Since the difference is positive ($[3,8]$), the events are ordered. According to MTV, the maximum frequency at which the circuit can operate correctly is 20.0 MHz. The fact that frequency attained by Scald is just half of that obtained by MTV is a direct result of the pessimism of the Scald algorithm and its inability to incorporate correlation information.

In a second type of correlation, MTV allows for correlated process/environmental variation by means of a correlation factor [9]. This stems from the observation that if a single gate is slow, then all gates exposed to the same environmental and processing variations are likely to be slow as well. The amount of correlation is indicated by a number ($0.0 \leq K \leq 1.0$); zero indicates no correlation and one indicates perfect correlation. In the example, if $K \geq 0.73$, then event d occurs before event g . Since MTV detects this, the occurrence of g will not cause a transition on *OE*. If the ordering is not detected (like in a Scald simulation), a Changing event is generated on *OE*.

3.3 Logical event ordering

Unlike Scald and Clover, MTV detects event orderings due to logical constraints as well as those due to temporal constraints. A logical ordering does not exist between events f and g , because only one of the inputs (f or d) need fall for the output to rise since g is the *Nand* of f and d .

Therefore, it cannot be assumed that the output follows a particular input. However, if the polarity of each of the signals is reversed (so that events d and f are both True), events d and f must both occur before event g , because both inputs must transition to True before the output can transition to False. MTV will detect this and simulate the circuit accordingly.

4 Simulation of event orderings

This section briefly highlights MTV’s simulation of ordered events and overlapping events; details can be found in the thesis describing this work [6].

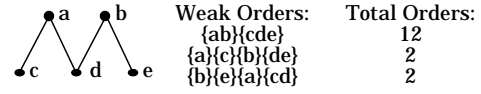


Figure 3: Partial, weak and total orders

There is a temporal partial order among the events at the inputs to a gate. Figure 3 illustrates a partial order and a set of weak orders to which it extends. In the weak order, events enclosed in braces are unordered. The number of total orders to which a weak order extends is indicated to the right of the weak order. A weak order allows some events to overlap while allowing others to be ordered. By simulating the weak order $\{ab\}\{cde\}$, MTV handles 12 total orders simultaneously. The ability to simulate overlapping events may result in far fewer cases than the exhaustive systems that generate all possible *total* orders.

Each weak order is simulated independently. To simulate a weak order, the sets of overlapping events are simulated, in turn, from first to last. Since the weak order “obeys” the partial order, any event that occurs before another in the partial order will be simulated before that event in the weak order. In this way, orderings between the events are respected in the simulation. MTV does not bifurcate when it simulates weak orders; after the weak event orders are simulated, the results are merged together to generate a single new event. Therefore, the behavior of a wire is represented by a single waveform.

It is desirable to respect event orderings in simulation for two reasons. First, if it can be determined that *controlling* input events (e.g. a False input to an *And* gate) occur before non-controlling input events, no Changing event will be generated on the output. This was observed in the example from the phase detector, but the same principle extends to gates with any number of inputs. Second, it may eliminate some events from the list of potential parents of an output event, thus preserving more temporal information.

5 Results from the ISCAS85 benchmarks

We have run MTV on a number of examples from the ISCAS85 benchmark suite. These examples were run using fifty randomly generated Boolean test vectors. All inputs transition at time 0. These results are presented to demonstrate the scalability of the algorithms used by MTV, and its improvements in accuracy over an approach like that of Scald and Clover. The results for “Scald” are approximated by using MTV with the appropriate optimizations disabled.

The results for the circuits in the ISCAS85 benchmark suite are shown in Table 1. The columns labelled “[10,15],” “[10,20]” and “[10,50]” show the percentage accuracy improvement of MTV over Scald for minimum gate delays of 10 and maximum of 15, 20 and 50, respectively. The accuracy of a simulator is defined as the summation over all signals of the amount of time the signal has the value Changing (a lower number indicates more accuracy). The percentage improvement is $(1 - (\text{Acc}_{\text{MTV}} / \text{Acc}_{\text{Scald}})) * 100$. The last column shows the execution time in seconds of the slowest of the test cases on an HP715/50. These results indicate that, as expected, MTV performs no worse than Scald, and may be more than twice as accurate, even on combinational circuits that are not timing-critical. Timing-critical circuits can yield far greater accuracy improvements.

Table 1: Percentage accuracy improvement with symbolic delays.

Circuit	[10,15]	[10,20]	[10,50]	Max Time
c432	1.30	3.15	1.99	0.04 sec
c499	25.23	44.85	56.45	0.78 sec
c880	15.43	17.85	8.89	0.61 sec
c1355	4.94	3.11	4.00	1.43 sec
c1908	17.80	11.58	7.41	2.61 sec
c2670	10.57	14.62	16.15	5.63 sec
c3540	4.26	5.86	3.95	6.39 sec
c5315	9.88	12.93	12.13	8.39 sec
c6288	8.35	7.38	5.35	14.51 sec
c7552	19.91	18.35	12.53	19.19 sec
Avg.	11.76	13.97	12.88	5.42 sec

6 Conclusion

We have presented MTV, a simulator combining symbolic min/max delay simulation and an extended value set. It is more accurate than Scald-based simulators and more efficient than exhaustive simulators. MTV eliminates common ambiguity, detects logically ordered events, and

simulates correlated delays without resorting to exhaustive simulation. MTV implements an accuracy/efficiency tradeoff between timing verifiers that simulate all possible orderings and those that do not resolve ordering ambiguities.

References

- [1] Srinivas Devadas, Kurt Keutzer, Sharad Malik, and Albert Wang. Event suppression: Improving the efficiency of timing simulation for synchronous digital circuits. *IEEE Trans. on CAD*, June 1994.
- [2] Dimitris Doukas and Andrea S. LaPaugh. CLOVER: A timing constraints verification system. In *Tau*, 1990.
- [3] Michael Franz, T. C. Whang, and Wallace Chou. A 240 MHz phase-locked circuit implemented as a standard macro on CMOS SOG gate arrays. In *CICC*, pages 25.1.1--25.1.4, 1992.
- [4] Nagisa Ishiura, Yutaka Deguchi, and Shuzo Yajima. Coded time-symbolic simulation using shared binary decision diagrams. In *DAC*, pages 130--135, 1990.
- [5] Nagisa Ishiura, Nizuki Takahashi, and Shuzo Yajima. Time-symbolic simulation for accurate timing verification of asynchronous behavior of logic circuits. In *DAC*, pages 497--502, 1989.
- [6] Mark H. Linderman. *Simulation of Digital circuits in the presence of uncertainty*. PhD thesis, Cornell University, 1994.
- [7] Alan R. Martello, Steven P. Levitan, and Donald M. Chiarulli. Timing verification using HDTV. In *DAC*, pages 118--123, 1990.
- [8] Thomas M. McWilliams. Verification of timing constraints on large digital systems. In *DAC*, pages 139--147, 1980.
- [9] Kostas Siomalas, Philippe Piche, and Giovanni Mancini. Correlated delays and hazard detection in digital simulation. Bell-Northern Research internal report, 1988.