Troubleshooting unintended temporal behavior

Johan de Kleer
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
dekleer@parc.com

Abstract

This paper addresses the challenge of reasoning and diagnosing digital circuits which contain either intentional or unintentional cycles in the combinational logic. Such cycles can lead to oscillatory behavior or convert what seems at first to be a combinational circuit into a sequential one. Such circuits often produce instant contradictions when analyzed at the logical gate level. This paper presents a temporal extension to the GDE framework which makes it possible to analyze and successfully troubleshoot such circuits.

Introduction

This paper addresses the challenge of reasoning and diagnosing digital circuits which contain either intentional or unintentional cycles in the combinational logic. Such cycles can lead to oscillatory behavior or convert what seems at first to be a combinational circuit into to a sequential one. Such circuits often produce instant contradictions when analyzed at the logical gate level. This paper presents a temporal extension to the GDE framework (de Kleer & Williams 1987; de Kleer, Mackworth, & Reiter 1992) to represent signals over time. With these new models it's possible to analyze and successfully troubleshoot such circuits.

The temporal analyses presented in this paper extend both the basic GDE-style models and the connection models in (de Kleer 2007). In this paper we make the following simplifying assumptions.

- All gate delays are a (small) integer multiple of the shortest gate delay δ.
- Signals take no time to propagate through connections.
- Does not take advantage of non-intermittency axioms (Raiman *et al.* 1991).

Motivating examples

Consider the 3 inverter circuit of Figure 1. It is easy to wire three inverters together — the system is physically realizable and connecting the inverters in this way does not damage them. However, using the usual model for gates described in (de Kleer & Williams 1987) or the extended models for connections described in (de Kleer 2007), GDE immediately detects a contradiction concluding that at least one of the components or nodes is necessarily faulted. In fact, none of the inverters are faulted. This is a well-known common circuit used to generate clock signals in digital circuits (Wikipedia 2007).

Table 1: Outputs of the inverters of a ring oscillator after t gate delays. The oscillator takes 6 gate delays to return to its initial state, thus the output is a square wave with a period of 6 times the gate delay.

t	0	1	2	3	4	5	6
A	1	1	1	0	0	0	1
B	1	0	0	0	1	1	1
C	0	0	1	1	1	0	0

Table 2: Truth table for SR flip-flop.

$\overline{R}$	$\overline{S}$	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	?	?

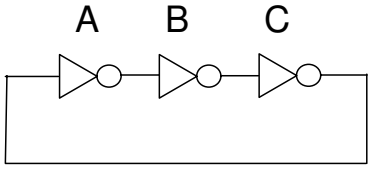

Figure 1: Simple ring oscillator.

Figure 2 illustrates an SR flip-flop. Table 2 describes the truth table for the circuit. The final line in the truth table represents two possible states the flip-flop could be in, and the values of Q and $\overline{Q}$ depend on previous history. If the preceding inputs were $\overline{S} = 0$ and $\overline{R} = 1$, then Q will become 1 and $\overline{Q}$ will become 0 (corresponding to setting the flip-flop). Symmetrically, if the inputs are flipped, the outputs will be flipped (corresponding to resetting the flip-flop). If the inputs change to 1's then Q and $\overline{Q}$ will retain their previous values. The SR flip-flop is the central way static memory elements are created in VLSI design.

Figure 3 shows a connection short which can cause un-

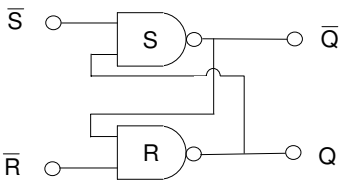

Figure 2: SR flip-flop. It consists of two cross-coupled NAND gates.

intended oscillation in c17 from (Brglez & Fujiwara 1985). With the inputs as indicated, the output at O1 should be 0. If O1 is shorted to I2, the circuit will oscillate. Modeling connections as in (de Kleer 2007), O1 will pull down I2 to 0, the output of the nand gate G1 will change from 0 to 1, and thus the output of the nand gate G5 will switch to 1. O1 is shorted to I2, so it will now return to 1. The circuit will continue oscillating in this manner forever. When modeled with GDE and the connection models of (de Kleer 2007), this circuit with these inputs are completely contradictory and the correct fault is therefore mistakenly eliminated from consideration. Many sets of design rules used in VLSI design try to minimize such hard to isolate faults by not allowing inputs to run adjacent to outputs.

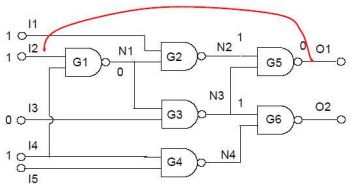

Figure 3: Short circuit causes undesired oscillation.

Figure 4 represents a commonly occuring short in CMOS which creates an SR flip-flop unintentionally. This is hard to troubleshoot because a previously simple combinational circuit now has state.

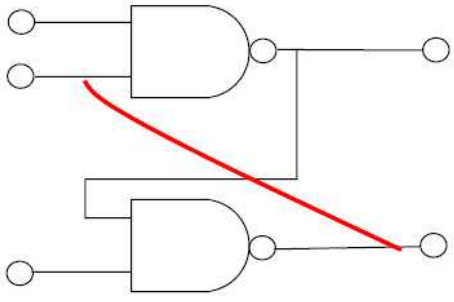

Figure 4: Short circuit which creates memory.

Modeling Components

The conventional GDE model for an inverter is:

$$INVERTER(x) \rightarrow \Big[\neg AB(x) \rightarrow [in(x,t) = 0 \equiv out(x,t) = 1]\Big].$$

Modeling an inverter as having a single delay Δ, the model changes to:

$$INVERTER(x) \rightarrow \Big[\neg AB(x) \rightarrow [in(x,t) = 0 \equiv out(x,t+\Delta) = 1]\Big].$$

To accomodate connection failures, signals on wires must be modeled at a more detailed level (de Kleer 2007). Each terminal of a component is modeled with two variables, one which models how the component is attempting to influence its output (roughly analogous to current), and the other which characterizes the result (roughly analogous to voltage). For a correctly functioning node, these voltage-like variables are equal. There are 5, mutually inconsistent, qualitative values for the influence of a component on a node (we refer to these as "drivers").

- $d(-\infty)$ indicates a direct short to ground.
- $d(0)$ pull towards ground (i.e., 0).
- $d(R)$ presents a high (i.e., draws little current) passive resistive load.
- $d(1)$ pull towards power (i.e., 1).
- $d(+\infty)$ indicates a direct short to power.

Intuitively, these 5 qualitative values describe the range of possible current sinking/sourcing behaviors of a component terminal. A direct short to ground can draw a large current inflow. A direct power to ground can drive a large current outflow.

There are three possible qualitative values for the result variable:

- $s(0)$ the result is close enough to ground to be sensed as a digital 0.
- $s(x)$ the result is neither a 0 or 1.
- $s(1)$ the result is close enough to power to be sensed as a digital 1.

With these connection models, the inverter is modeled as:

$$\begin{aligned} INVERTER(x) \rightarrow & \Big[\neg AB(x) \rightarrow \\ & [s(in(x,t)) = s(0) \rightarrow d(out(x,t)) = d(1) \\ & \wedge s(in(x,t)) = s(1) \rightarrow d(out(x,t)) = d(0) \\ & \wedge d(in(x,t)) = d(R) \\ & \wedge d(out(x,t)) = d(0) \vee d(out(x,t)) = d(1)]\Big]. \end{aligned}$$

Modeling the inverter to more accurately describe both temporal and causal behavior:

$$
\begin{aligned}
&INVERTER(x) \rightarrow \\
&\quad \Big[\neg AB(x) \rightarrow \\
&\quad [s(in(x,t)) = s(0) \rightarrow d(out(x,t+\Delta)) = d(1) \\
&\quad \wedge s(in(x,t)) = s(1) \rightarrow d(out(x,t+\Delta)) = d(0) \\
&\quad \wedge d(in(x,t)) = d(R) \\
&\quad \wedge d(out(x,t+\Delta)) = d(0) \vee d(out(x,t+\Delta)) = d(1)]\Big].
\end{aligned}
$$

Representing Time

Our implementation is based on the ATMS/HTMS structure. Each time instant is modeled by an explicit assumption $t = i$ and any two time assumptions contradict each other. These assumptions separate all inferences into their respective times. However, to instantiate the component models which incorporate evolving time, a non-monotonic ATMS rule is required. An instantiation of the model for a particular inverter A is encoded in the following two clauses (the implication $x \wedge y \wedge \cdots \rightarrow z$ is equivalent to the logical clause $\neg x \vee \neg y \vee \cdots \rightarrow z$ and the literal $\neg(q = 0)$ is equivalent to literal $q = 1$):

$$
\begin{aligned}
AB(A) \vee in(A,t) = 1 \vee [out(A,t+\Delta) = 1] \\
AB(A) \vee in(A,t) = 0 \vee [out(A,t+\Delta) = 0].
\end{aligned}
$$

However, time is implicitly represented, by explicit assumptions. So the assumption for $t = i$ would be implicit in any deduction for $in(A,i)$. Any deduction of $in(A)$ will have a single $t = i$ assumption. We introduce a new modal operator N which specifies that its argument holds in the next point beyond its antecedent. In its two argument form $N[x,i]$ specifies x holds in i gate delays. So the two inverter clauses become:

$$
\begin{aligned}
AB(A) \vee in(A) = 1 \vee N[out(A) = 1] \\
AB(A) \vee in(A) = 0 \vee N[out(A) = 0].
\end{aligned}
$$

These clauses are modeled in the ATMS by the following non-monotonic inference rule: Every (final or intermediate) prime implicate of the form,

$$AB(c_1) \vee \cdots \vee AB(c_n) \vee \neg(t = i) \vee N(x),$$

is rewritten:

$$AB(c_1) \vee \cdots \vee AB(c_n) \vee \neg(t = i+1) \vee x.$$

This rule is non-monotonic because the result no longer depends on the existence of the previous time instant. Without this inference rule, it would not be possible to model evolution of time as time is inherently non-monotonic.

An advantage of this scheme to represent time is that it is not necessary to make multiple copies of component models for each time. Once the clauses representing the system model are represented, all further propagations are accomplished through the ATMS.

Suppose the input to the first inverter of the ring oscillator is observed to be 0 at t_1. Table 3 lists the sequences of deductions which follows. All subsequent inferences follow the same pattern. No new assumptions are added.

Table 3:

t	deduction
1	$AB(A) \vee \neg(t = 1) \vee N[out(A) = 1]$
2	$AB(A) \vee \neg(t = 2) \vee [out(A) = 1]$
2	$AB(A) \vee AB(B) \vee \neg(t = 2) \vee N[out(B) = 0]$
3	$AB(A) \vee AB(B) \vee \neg(t = 3) \vee [out(B) = 0]$
3	$AB(A) \vee AB(B) \vee AB(C) \vee \neg(t = 3) \vee N[out(C) = 1]$
4	$AB(A) \vee AB(B) \vee AB(C) \vee \neg(t = 4) \vee [out(C) = 1]$
4	$AB(A) \vee AB(B) \vee AB(C) \vee \neg(t = 4) \vee N[out(A) = 0]$
5	$AB(A) \vee AB(B) \vee AB(C) \vee \neg(t = 5) \vee out(A) = 0$

Modeling the ring oscillator

Applying the temporal models produces the envisionment of Figure 5. The usual states associated with a ring oscillator are described by the states on the outside of the envisionment figure. This corresponds to a stable oscillation of period 6 gate delays. This oscillation is stable because there are no ambiguous transitions. However, there are two spurious states in a meta-stable oscillation at the center of the envisionment diagram. Over time one of the gate delays will be slightly longer or shorter and the others and the system will transition to one of the 6 stable states. All transitions out of 000 and 111 are ambiguous because multiple transitions are happening simultaneously. While in the 000-111 oscillation will look like Figure 10. For this reason, most practical ring oscillators have an enable input (Figure 8 so that the circuit starts in a known stable state.

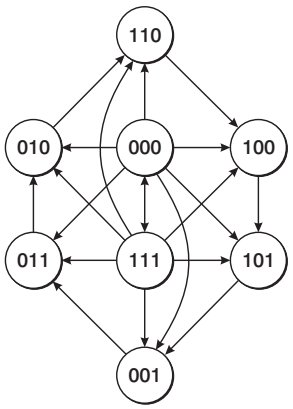

Figure 5: Envisionment for the ring oscillator. The vertices are labeled with the values for the outputs of A, B, C respectively.

Signals

The notation $f(g,t)$ (e.g., $in(A,t)$) to represent values is a clumsy approach to represent a changing value for time (a fluent). We introduce a notion of signal. A signal represents the evolution of a value over time (at the granularity of a gate delay). It is represented by a sequence of symbols drawn from `0,1,?,>,*.`

- `0,1` indicate their respective values,

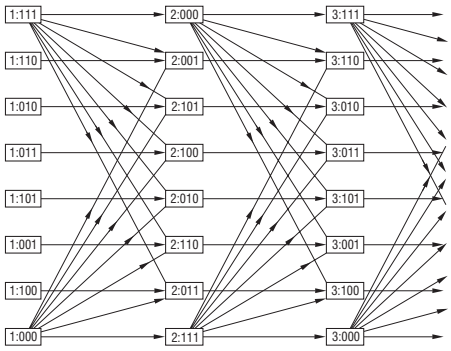

Figure 6: Qualitative simulation of the ring oscillator. The vertices are labeled: t:A,B,C where t is the time in gate delays and A,B,C are the outputs of the respective inverters.

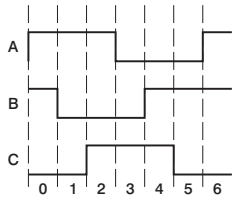

Figure 7: Desired output of the ring oscillator.

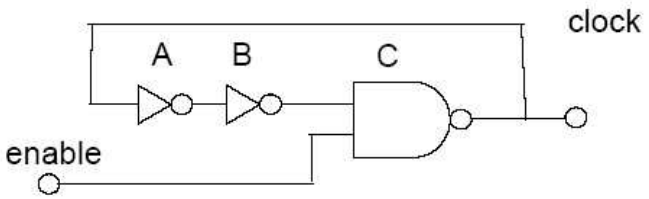

Figure 8: A better designed ring oscillator. When the clock is enabled, the ring oscillator will start in a known good state: 011

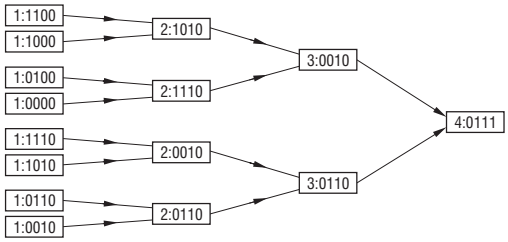

Figure 9: The oscillatory behavior of the better ring oscillator. Vertices are labeled t:nnnn: t is the time, nnnn are the outputs of gates A,B,C and the enable input. The enable signal is 000*1.

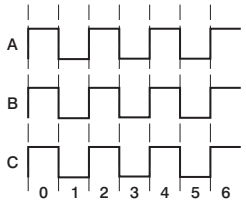

Figure 10: Undesired spurious oscillation of the ring oscillator.

- ? indicates the value is unknown,
- > indicates a large (unknown) number of gate delays,
- $\star$ means that the following pattern repeats.

A signal may have only one $\star$ and it must occur at the earliest possible place. We call signals containing a $\star$ followed by both 1's and 0's as definitely oscillatory. We call signals in which $\star$ is followed by exactly one symbol 0 or 1 as steady. Examples of valid signals are:

- $\star$0 the steady signal always 0.
- >$\star$1 the steady signal eventually 1.
- >$\star$000111 the oscillatory output of a 3 gate ring oscillator

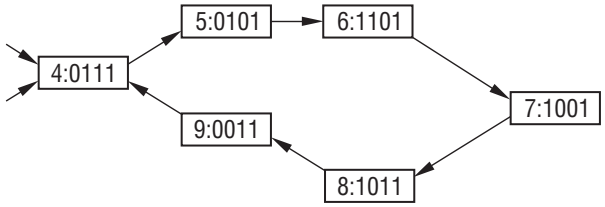

Figure 11: Qualitative simulation of the better ring oscillator. Vertices are labeled t:nnnn: t is the time, nnnn are the outputs of gates A,B,C and the enable input. The enable signal is 000*1.

Signals are used to describe all of a system's variables. Figure 11 shows a continuation of the simulation of Figure 9. It shows that the resultant signal at the output of the ring oscillator is: ?$\star$111000. Our extended GDE, infers not just the values of variables at particular time points (such as $in(A, t_1)$) but also all the most general signals (presuming all signals eventually have a repeating pattern). The assumptions supporting this derivation are the assumptions of its constituent components:

$$AB(A) \vee AB(B) \vee AB(C) \vee (C =? * 111000).$$

On the surface, these derivations don't seem to have much diagnostic power, as in most cases the signals will be depend on the non-ABnormal behavior of all the components of the system.

To illustrate the diagnostic power of inferring complex signals consider a slightly more complex ring oscillator consisting of 4 inverters and one nand gate as illustrated in Figure 12. Assume the input is `00000*1`. GDE will compute an output of `>*0000011111`. Suppose the output is observed to be `*0`. Then we immediately have the conflict:

$$AB(A) \vee AB(B) \vee AB(C) \vee AB(D) \vee AB(E) \vee AB(G).$$

The obsevation `*0` is propagated (as it would if a simple 0 were observed). If all components are equally likely to fail, the usual GDE probing strategy will choose the next measurement (either the output of B or C) and the gate causing the failure to oscillate can be isolated as usual. More intuitively, as we are observing the system when all observations and inputs are steady, the circuit looks like that of Figure 13 which can be easily diagnosed.

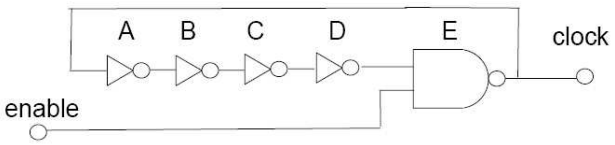

Figure 12: A 5 gate ring oscillator.

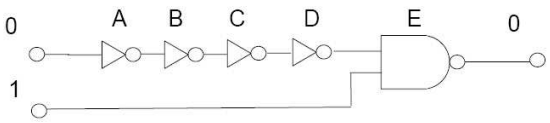

Figure 13: A view of the 5 gate ring oscillator observed at quiescence.

Consider the circuit of Figure 14 which combines an SR flip-flop with a ring oscillator. The purpose of this circuit is to ensure the clock is on (it might be on already) when the flip-flop sees a single negative pulse at S. Suppose the clock output is observed to be quiescent 0. Observed at quiescence every component is obeying its io-behavior: S=*0, R=*1, C=*0, B=*1, A=*1, yet the circuit is faulted. With the two given inputs at quiescence being 1, the outputs of S and R cannot be definitively inferred. Even though we observe S to be *0, at quiescence that cannot be inferred. So that observation provides no information about whether S or R are faulted or not. Fortunately, the fault at S and R can be directly determined by the basic propagation of signals. Given the observations as shown, the signal at the output of S should be ???1* hence one of S or R are faulted. For example, S stuck at 0, or R stuck at 1 would explain the observation. The reason its hard to diagnose at quiescence is that S stuck at 0 is the same output as if it were working correctly. The fault occurred in the past.

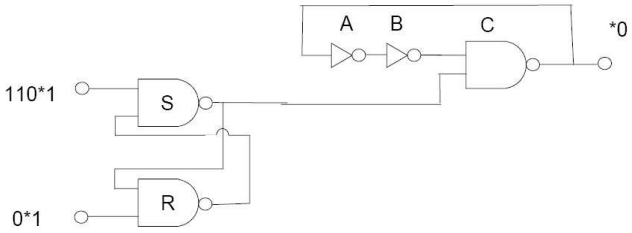

Figure 14: A flip-flop combined with a ring oscillator. The figure shows the initial inputs and it is observed at quiescence: S=*0, R=*1, C=*0, B=*1, A=*1 which is consistent with all the component models, yet the output is incorrect.

Related work

(Dague *et al.* 1991) focused on troubleshooting analog oscillators which stopped oscillating. (Kaul, Biswas, & Bhuva 1994) focused on simulating CMOS designs with QSIM (Kuipers 1986). Neither approach generalized to explicitly recording assumptions of good behavior in order to isolate the malfunctioning component through more observations. (Dressler 1990) incorporates a non-monotonic inference rule in the ATMS but does not employ it to diagnose signals over time. More details of digital circuit behavior can be found in (Weste & Eshraghian 1993) and (Johnson & Graham 1993).

Conclusions

This research has extended the GDE framework with a vocabulary to represent signals over time. The same architecture and inferential machinery as GDE can be used to propagate these generalized signals. Discrepancies between observed and predicted generalized signals can guide diagnosis to isolate the faulty system component(s). These generalized signals capture temporal behavior over time so they can be used to troubleshoot sequential circuits (e.g., containing flip-flops) as well. Sequential circuits can be hard to diagnose because no symptom maybe observable at quiescence. With these extensions GDE can be used to troubleshoot a far wider range of circuits than previously.

References

Brglez, F., and Fujiwara, H. 1985. A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In *Proc. IEEE Int. Symposium on Circuits and Systems*, 695–698.

Dague, P.; Jehl, O.; Deves, P.; Luciani, P.; and Taillibert, P. 1991. When oscillators stop oscillating. In *Proc. 12th Int. Joint Conf. on Artificial Intelligence*, 1109–1115.

de Kleer, J., and Williams, B. C. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32(1):97–130. Also in: *Readings in NonMonotonic Reasoning*, edited by Matthew L. Ginsberg, (Morgan Kaufmann, 1987), 280–297.

de Kleer, J.; Mackworth, A.; and Reiter, R. 1992. Characterizing diagnoses and systems. *Artificial Intelligence* 56(2-3):197–222.

de Kleer, J. 2007. Modeling when connections are the problem. In *Proc 20th IJCAI*, 311–317.

Dressler, O. 1990. Problem solving with the NM-ATMS. In *Proc. 9th Eur. Conf. on Artificial Intelligence*, 253–258.

Johnson, H. W., and Graham, M. 1993. *High-speed digital design: a handbook of black magic*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Kaul, N.; Biswas, G.; and Bhuva, B. 1994. An artificial intelligence approach to multi-level mixed-mode qualitative simulation of cmos ics. *Comput. Electr. Eng.* 20(5):369–382.

Kuipers, B. J. 1986. Qualitative simulation. *Artificial Intelligence* 29(3):289–338. Also in: D. S. Weld and J. de Kleer (eds) *Readings in Qualitative Reasoning about Physical Systems* (Morgan Kaufmann, Los Altos, Calif., 1990).

Raiman, O.; de Kleer, J.; Saraswat, V.; and Shirley, M. H. 1991. Characterizing non-intermittent faults. In *Proc. 9th National Conf. on Artificial Intelligence*, 849–854.

Weste, N. H. E., and Eshraghian, K. 1993. *Principles of CMOS VLSI design: a systems perspective, second edition*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Wikipedia. 2007. Ring oscillator — Wikipedia, the free encyclopedia. [Online; accessed 12-February-2007].