Modeling when connections are the problem

Johan de Kleer
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA

Abstract

Most of the AI diagnostic reasoning approaches for digital systems presume that digital components can be modeled as pure functions of their inputs, all signals can be represented by "1"s and "0"s, wires between components cannot fail, and do not model replacement of components or wires. None of these assumptions are valid for the challenges diagnosticians encounter in real systems. This paper presents a digital expert (DEX) that can reason over digital systems as an electrical engineer would: at a qualitative, causal level more accurate than the simple "0"/"1" level, but without incurring the costs of full-scale numerical algorithms. A fundamental contribution of this paper is a more powerful approach to modeling connections which does not require special-case post-processing and is computationally tractable.

Introduction

Most of the AI diagnostic reasoning approaches for digital systems (Hamscher, de Kleer, & Console 1992) presume that digital components can be modeled as pure functions of their inputs, all signals can be represented by "1"s and "0"s, wires between components cannot fail, and do not model replacement of components or wires. None of these assumptions are valid for the challenges diagnosticians encounter in real systems. Although digital systems can be modeled at the analog level with programs such as SPICE or, at the digital/analog level, with VHDL-based simulators, they are not designed for diagnostic use, require accurate hard-to-obtain component models and do not present results in a way a human diagnostician can understand. The objective of this research is to design a digital expert (DEX) that can reason over digital systems as electrical engineer would: at a qualitative, causal level more accurate than the simple "0"/"1" level, but without incurring the costs of full-scale numerical algorithms.

A key principle for model design is that models be *veridical*, directly linking causality with effect. Examples of non-veridical models are the "Stuck-at-0" and "Stuck-at-1" models commonly used for reasoning over digital circuits. "Stuck-at-1" could represent one of four possible faults: that the driving gate's output is stuck at 1, that some driven gate's input is stuck at 1, that the wires are shorted to power, or that the output is undriven and the signal floated to 1. In our veridical models all of these inferences are drawn from the particular model which is causing the malfunction — a wire, gate, or short. (We will use "wire" to refer to any electrical connector.)

In this paper we propose a new methodology for modeling signals in wires by explicitly representing causality for component behaviors. Instead of modeling a connection with a "1" or "0," each connection is modeled by multiple variables: one indicating the signal level of the node and the rest representing the causal drivers of the node (see Figure 1). Consequently, opens and shorts can be directly modeled, component models can be causally accurate and it is possible to extend the classes of component models to include tri-state and open-collector components.

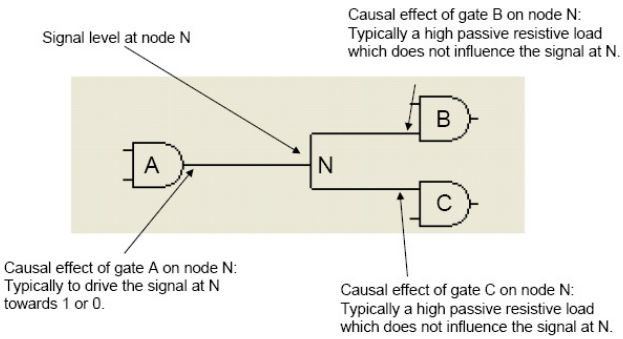

Figure 1: Instead of only modeling a node N with one signal 0 or 1, we model each potential connection which could possibly influence the signal level on the node with a qualitative variable. So each circuit node is modeled with $n + 1$ variables, one for the final signal level, and one for each of the n components connected to it.

This paper demonstrates the approach on a variety of circuits from the standard test suite described in (Brglez & Fujiwara 1985). Circuits of up to 400 nodes and 500 components can be modeled and diagnosed efficiently (< 1 minute on a modern PC). Previous work on shorts and bridge faults presumed considering all possible shorts to be computationally unreasonable and introduced special-case inference procedures to handle them (see section on Related Work). The analysis will show that extremely few possible shorts can

explain typical symptoms, and those that do are a small fraction of the possible candidates. Thus, model-based diagnosis systems containing shorts can be performed within the framework of existing algorithms using the models presented in this paper.

DEX utilizes a re-implementation of the GDE/Sherlock (de Kleer & Williams 1987) probabilistic framework based on the HTMS (de Kleer 1992). All the models in this paper are compiled to their prime-implicates with unmeasurable variables eliminated. The primary inference mechanism is local constraint-propagation where completeness (for conflicts and value propagation) is ensured though the introduction of additional assumptions for unassigned measurable variables, propagating these and subsequently using logical resolution to eliminate the ambiguities.

In this paper, we make the following simplifications, which we intend to relax these in future work:

- Components behave non-intermittently. Later in the paper we will generalize definition of non-intermittency from that given in (Raiman *et al.* 1991): A component behaves non-intermittently if its outputs are a function of its inputs. As the models in this paper allow causality to change, inputs and outputs are no longer well-defined.
- No causal loops in the combinatorial logic.
- No logical memory elements.
- No model of transient behavior. All signals are presumed to have reached quiescence after the input vector has been applied. Thus DEX cannot reason about hazards, race conditions, or situations in which a node's value switches too slowly because its not driven with enough current or there are too many loads connected to it.
- No fault propagation. A fault in one component cannot cause a fault in another.

Related work

Early work on model-based diagnosis (Davis 1984) addresses bridge faults. However, this early research treats shorts as a special case, hypothesizing bridge faults only when all single faults were eliminated. (Preist & Welham 1990) inserts additional insulating components at places where shorts may occur and uses stable-model semantics to identify candidate diagnoses. This approach is too inefficient, as the number of possible insulator components to consider grows quadratically with system size. (Boettcher, Dague, & Taillibert 1996) model structural shorts in analog systems. Again this approach uses the possibility of multiple-faults to invoke an additional algorithm to match observed behavior to known hidden interaction models.

The broadest system modeling techniques come from the QR and MBD work in the automotive diagnosis and the FMEA construction domains (Struss, Malik, & Sachenbacher 1995) (N.A.Snooke & C.J.Price 1997) (Mauss, May, & Tatar 2000). One methodology for using multiple variables to represent wires can be found in (Struss, Malik, & Sachenbacher 1995).

A fundamental contribution of this paper is a more powerful approach to modeling connections which does not require special-case post-processing and is computationally tractable. Only one additional component needs to be added to model each node, so the number of additional components grows linearly in the worst-case. This method provides a new way to model digital components which can model short circuits, open circuits, and tri-state, open-collector, and expand gates. Structural faults are modeled much like any other fault and are integrated with the GDE/Sherlock approach to measurement selection and component replacement policies. Most of the potential computational complexity introduced by the more detailed causal models is avoided by generating candidate diagnoses in best-first order. Candidates are ordered by their posterior probabilities, not the number of faults they contain.

Preliminaries

This basic framework is described in (de Kleer & Williams 1987; de Kleer, Mackworth, & Reiter 1992).

Definition 1 *A system is a triple (SD,COMPS, OBS) where:*

1. *SD, the system description, is a set of first-order sentences.*
2. *COMPS, the system components, is a finite set of constants.*
3. *OBS, a set of observations, is a set of first-order sentences.*

Definition 2 *Given two sets of components Cp and Cn define $\mathcal{D}(Cp, Cn)$ to be the conjunction:*

$$\Big[\bigwedge_{c \in Cp} AB(c)\Big] \wedge \Big[\bigwedge_{c \in Cn} \neg AB(c)\Big].$$

Where $AB(x)$ represents that the component x is ABnormal (faulted).

A diagnosis is a sentence describing one possible state of the system, where this state is an assignment of the status normal or abnormal to each system component.

Definition 3 *Let $\Delta \subseteq$COMPS. A diagnosis for (SD,COMPS,OBS) is $\mathcal{D}(\Delta, COMPS - \Delta)$ such that the following is satisfiable:*

$$SD \cup OBS \cup \{\mathcal{D}(\Delta, COMPS - \Delta)\}$$

In this framework, a typical model for an inverter is (assuming the appropriate domain axioms for variables):

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

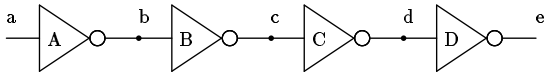

Figure 2: Four sequential inverters.

The model for the second inverter of Figure 2 is:

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

Introducing causality

Models of the type described in the previous section implicitly assume that components have distinguished input terminals that only sense their inputs and then cause an output variable value. Both of these assumptions can be faulty. For example, B's input can be shorted to ground internal to the gate. In this case, despite the 1 signal from A, b would be measured to be 0. If $a = 0$, then the classic models would dictate that A is faulted, when in fact it is actually B that is faulted.

To model causality more accurately, we must model wires with more accuracy. Each terminal wire 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 equivalent. There are 5 possible qualitative values for a component driver:

- $d(-\infty)$ 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 cause a large current outflow.

There are two possible qualitative values for the result variable:

- $s(0)$ the result is close enough to ground to be sensed as a digital 0.
- $s(1)$ the result is close enough to power to be sensed as a digital 1.

With few exceptions, correctly functioning digital devices present a high (little current) resistive load on all their inputs and drive all their outputs. Unless otherwise noted these axioms will be included in every component model.

Every output drives a signal (except in special cases described later):

$$\neg AB(x) \rightarrow [d(out(x,t)) = d(0) \vee d(out(x,t)) = d(1)].$$

Every input presents a resistive load:

$$\neg AB(x) \rightarrow d(in(x,t)) = d(R).$$

Under this modeling regime, an inverter is modeled as follows:

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

Under the classical modeling regime, there is no need to model the behavior of nodes as they just pass through their signals. However, we need explicit models to describe how the sensed digital value of the node is determined by its drivers. Let $R(v)$ be resulting signal at node v and $S(v)$ be the collection of drivers of node v. For example, $S(b) = \{d(out(A,t), d(in(B,t))\}$. Nodes are modeled as follows (sometimes referred to as 0-dominant models):

- If $d(-\infty) \in S(v)$, then $R(v) = s(0)$.
- If $d(+\infty) \in S(v)$, then $R(v) = s(1)$.
- If $d(0) \in S(v)$, then $R(v) = s(0)$.
- Else, if all drivers are known, and the preceding 3 rules do not apply, then $R(v) = s(1)$.

For example, node b of Figure 2 is modeled as follows:

$$\begin{aligned} &\neg AB(b) \rightarrow \\ &\quad \Big[d(out(A,t)) = d(-\infty) \rightarrow s(b) = s(0) \\ &\quad \wedge d(in(B,t)) = d(-\infty) \rightarrow s(b) = s(0) \\ &\quad \wedge d(out(A,t)) = d(+\infty) \rightarrow s(b) = s(1) \\ &\quad \wedge d(in(B,t)) = d(+\infty) \rightarrow s(b) = s(1) \\ &\quad \wedge d(out(A,t)) = d(0) \rightarrow s(b) = s(0) \\ &\quad \wedge d(in(B,t)) = d(0) \rightarrow s(b) = s(0) \\ &\quad \wedge [d(out(A,t)) = d(1) \wedge d(in(B,t)) = d(1) \rightarrow s(b) = 1] \\ &\quad \wedge [d(out(A,t)) = d(1) \wedge d(in(B,t)) = d(R) \rightarrow s(b) = 1] \\ &\quad \wedge [d(out(A,t)) = d(R) \wedge d(in(B,t)) = d(R) \rightarrow s(b) = 1] \\ &\quad \wedge [d(out(A,t)) = d(R) \wedge d(in(B,t)) = d(1) \rightarrow s(b) = 1]\Big] \end{aligned}$$

A naive implementation would require the construction of $2^n + 3n$ clauses for n-terminal nodes. DEX constructs these clauses only when they are needed to analyze a leading diagnosis using the consumer architecture of the HTMS.

We have now laid the groundwork for a new definition of non-intermittency. The definition from (Raiman *et al.* 1991) is:

Definition 4 *(Raiman* et al. *1991) A component behaves non-intermittently if its outputs are a function of its inputs.*

This definition succinctly captures the intuition of non-intermittency: (1) a component has exactly one output value for a particular set of input values, (2) even though other circuit values may change, the same inputs yield the same outputs, (3) the inputs are clearly identified — so no "hidden" input can be effecting the output value. We use these same intuitions for our new definition, except that the notion of "input" and "output" is changed.

Definition 5 *The causal inputs to a component are the signal levels at all the circuit nodes the component is connected to. The causal outputs of a component are the driving signals*

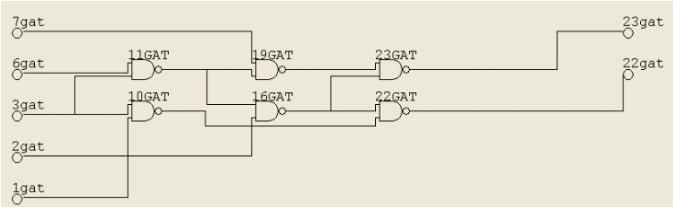

Figure 3: The simplest circuit, c17, from the test suite (Brglez & Fujiwara 1985). The schematic has been, crudely, automatically generated from the benchmark file. All wire crossings do not connect directly. All gates are labeled with upper-case "nnnGAT" and all nodes are labeled with their corresponding (except for inputs) lower-case "nnngat". When the label is unambiguous, node names are elided to reduce clutter.

on all of the wires to the nodes it connects to. A component is causally non-intermittent if all its driving outputs are a function of its sensed inputs.

Under this definition, a "2-input and gate" has 3 sensed inputs and 3 driven outputs. This general definition captures all possible faults of the 2-input and gate including such extreme possibilities of installing the wrong gate or installing it backwards. Correctly functioning components will drive all their outputs, but most will not reference the signal level on its teleological output (what the logic designer would call the "output.")

Consider the slightly more complex circuit of Figure 3 — the simplest examples in the test suite (Brglez & Fujiwara 1985). Suppose a test vector (1gat=0, 2gat=0, 3gat=1, 6gat=1, 7gat=0) is applied and 22gat is measured to be 1 (correct is 0). Figure 4 highlights the faulty components under the simple GDE models. Using the new causal models, 23GAT can fail with its input stuck to ground. Therefore, in Figure 5 gate 23GAT is also possibly faulted.

In order to model that nodes can fail, we model all nodes as components with the same model described earlier. For example, Figure 6 highlights the more probable node and component faults.

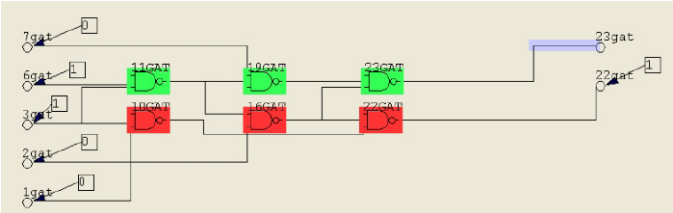

Figure 4: The inputs and symptom added. The squares on the figure contain the known signal levels on the respective nodes. Output 22gat should be 0 but is measured to be 1. All components fail with equal prior probability. After observing the symptom, the more likely faulted gates are 10GAT, 16GAT and 22GAT (which are highlighted in red or darker shading).

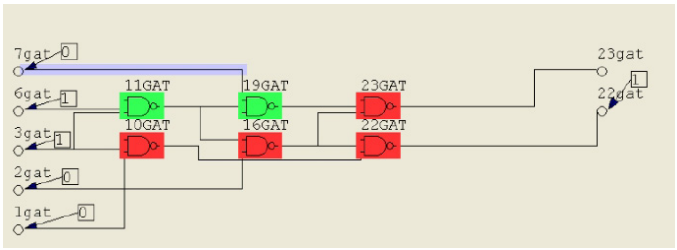

Figure 5: Using the expanded node model, gate 23GAT could have its input shorted to ground. Causing a symptom at 22gat (should be 0 but 1 is measured). The likely faulted gates are now 10GAT, 16GAT, 22GAT and 23GAT.

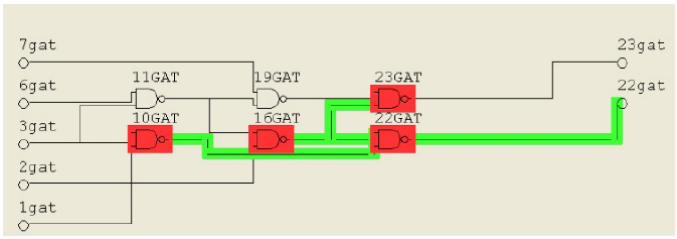

Figure 6: Modeling nodes as possibly faulted components. The highlighted components are more likely to be faulted given the observations. Component faults (in red or darker shading) are more likely than connection faults (in green or lighter shading). Lower probability components and nodes are not shaded. Prior fault probabilities of components are all equal, as are the component probabilities (in this case with higher priors).

Bridges and shorts

In order to model bridges and shorts we add one fault mode to the model of a node. A node can either be in mode G (working correctly $\neg AB$), S for shorted or U (unknown or no model). For every node:

$$AB(n) \rightarrow S(n) \odot U(n).$$

The G model is unchanged from that described earlier in the paper. There are two additional nodes, power and ground whose output drivers are $d(-\infty)$ and $d(+\infty)$ to model shorts to power and ground.

We make the simplifying assumption that we are looking at one set of shorts at a time. All of the nodes which are shorted in any candidate diagnosis are considered shorted together. Each of the nodes in this set will have the same signal level, which is determined as if the combined node were functioning in an overall G mode. For example, consider the candidate diagnosis of circuit c17 in which 11gat and 16gat are shorted together and all other components and nodes function correctly. Table 1 lists the drivers of the combined node, and by the node-model this will produce a signal at 16gat of 0 which propagates through 22GAT to produce 1 which is the observed symptom. Therefore, a 11gat-to-16gat short is a candidate diagnosis which explains all the symptoms.

Table 1: Combined drivers of nodes 11gat and 16gat.

io	gate	drive
output	11GAT	d(0)
input	19GAT	d(R)
input	16GAT	d(R)
output	16GAT	d(1)
input	22GAT	d(R)
input	23GAT	d(R)

Notice that the only possible short with node 22gat which explains the symptom is a short to power (assuming all other components and nodes are working correctly). As the output driver of 22GAT is $d(1)$ it cannot pull up any 0 node.

Most combinations of shorts make no causal sense and these are eliminated as a consequence of our models (no additional machinery is required). A trivial instance of a nonsensical short is between nodes 3gat and 11gat. The drivers of this combined node are listed in Table 2. So, the signal at the combined node 3gat-11gat will become 0, which produces an inconsistency with the correct model of the nand gate 11GAT and will not be considered a possible short (again assuming all other components and nodes are working correctly). If this short had happened in a physical circuit, the circuit would probably oscillate. As we are not modeling oscillation, we let the component and node models detect the inconsistency and so no candidate diagnosis will include it.

Consider the case where the nodes 1gat and 22gat are shorted (with same observations of Figures 5 and 6). The drivers of the combined node are listed in Table 3. Gates 10GAT and 22GAT appear to be in a loop. Thus the signal

Table 2: Combined drivers of nodes 3gat and 11gat.

io	gate	drive
driven	3GAT	d(1)
input	11GAT	d(R)
input	10GAT	d(R)
output	11GAT	d(0)
input	19GAT	d(R)
input	16GAT	d(R)

Table 3: Combined drivers of nodes 1gat and 22gat.

io	gate	drive
driven	1GAT	d(0)
input	10GAT	d(R)
output	22GAT	?

levels on nodes 1gat, 10gat and 22gat cannot be determined by only considering the driver values. Fortunately, the nand model for 22GAT resolves the ambiguity:

$$\neg AB(22GAT) \rightarrow [d(out(22GAT, t)) = d(0) \vee d(out(22GAT, t)) = d(1)].$$

Thus the output driver of 22GAT cannot be $d(+\infty)$ and thus it can be immediately inferred that shorting 1gat and 22gat does not explain why 22gat is observed to be 1 instead of 0. The only shorts which explain the symptom are shown in Table 4. It is interesting to note that the majority of possible shorts are ruled out by just measuring one output signal. Only $\frac{9}{66}$ of the possible shorts explain the evidence.

Table 4: Upper diagonal of this matrix gives the only possible two node shorts for our c17 example which explain the symptoms. For brevity nodes are indicated by their integers. Shorts to ground and power are not included.

	1	2	3	6	7	10	11	12	16	19	22	23
1						S			S			
2						S			S			
3												
6												
7						S		S				
10							S					S
11									S			
12												
16												
19												
22												
23												

DEX implements this shorting paradigm as follows. Candidate diagnoses are generated in best-first posterior probability order as in the GDE/Sherlock framework. The nodes are simply components. This candidate generator is modified to never generate candidate diagnoses of only one

Table 5: Number of possible shorts of 2 nodes for a typical symptom for the worst-case where all 2 shorts are equally likely. The percentages characterize how many of all possible 2 or smaller candidates are node shorts. All circuits come from the (Brglez & Fujiwara 1985) test suite. c432 is 27-channel interrupt controller, c499 is a 32-bit single-error-correcting-circuit, and c880 is an 8-bit arithmetic logic unit.

circuit	components	nodes	2-shorts	%
c17	6	11	9	7
c432	160	195	976	6
c499	201	242	562	2
c880	384	442	4959	8

shorted node. Whenever a new candidate diagnosis is identified with two or more shorted nodes, any needed additional clauses are added dynamically to model the combined set of shorted nodes. One added clause disjoins the correct node model with the clause with all the U modes of all the possibly remaining shortable nodes in the current diagnosis. This is important to ensure that shorts of size n will not incorrectly eliminate shorts of size $n+1$. Other clauses are added to ensure the signal levels at the shorted nodes are equal. DEX can also be provided matrix of prior probabilities of all shorts to incorporate into its Bayes rule calculation of the posterior probability of a candidate given evidence.

As we saw in circuit c17, surprisingly few shorts explain the observations that have been collected on the circuit. Table 5 shows that, as compared to all possible candidates which explain the symptoms, the percentage of shorted node candidates is relatively small. In addition, the best-first candidate generator will focus towards the most likely shorts, and not generate candidate diagnoses which contain unlikely shorts. Thus, the probabilistic framework does the main work in avoiding considering exponentially many combinations of shorted nodes.

Modeling tri-state and open-collector devices

With a modeling paradigm which distinguishes signal levels from drivers, it is simple to model a tri-state device (not possible when modeling gates as purely logical). When $G = 1$ in Figure 7, the gate acts as any other buffer. However, if $G = 0$, the tristate output only presents high resistive load at its output no matter the value of input A (an exception to the usual output driver):

$$TRISTATEBUF(x) \wedge \neg AB(x) \rightarrow$$

$$\Big[s(G(x,t)) = s(1) \rightarrow$$
$$[s(A(x,t)) = s(0) \rightarrow d(Y(x,t)) = d(0)$$
$$\wedge s(A(x,t)) = s(1) \rightarrow d(Y(x,t)) = d(1)$$
$$\wedge d(Y(x,t)) = d(0) \vee d(Y(x,t)) = d(1)]\Big]$$
$$\wedge d(A(x,t)) = d(R)$$
$$\wedge d(G(x,t)) = d(R)$$
$$\wedge G(s(x,t) = s(0) \rightarrow d(Y(x,t)) = d(R).$$

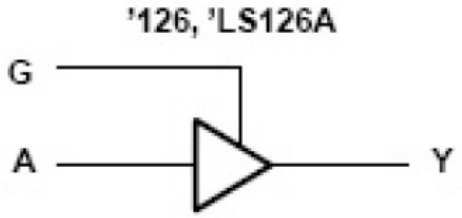

Figure 7: Tri-state Buffer.

Open-collector devices do not need an additional model as these are 0-dominant as well. Without an external pull-up resistor these lines rise slowly to their signal level but DEX does not yet model this transient behavior.

IC Components

The methodology in this paper can be used to analyze a system to the gate level. However, in many cases the system is composed of integrated circuits that each contain many gates, and troubleshooting need only identify the faulty IC. Figure 8 is the familiar IC which contains 4 2-input nand gates. Intuitively, it looks like we can utilize a single AB-literal for the IC and all the nand-models depend on its negation. This has two problems. First, the extension to fault models is cumbersome. If we model an individual nand gate with 3 faults (e.g., SA0, SA1, U), then the IC would have 255 fault modes. Second, the ability to suggest measurement is impeded because 4 components are removed when considering the IC faulty. Therefore, DEX models all ICs with sets of prime-implicates containing only IC terminal variables and the one IC AB-literal, and simply replaces these clauses with individual gate AB-literals and associated clauses whenever the IC AB-literal occurs in any candidate leading diagnosis.

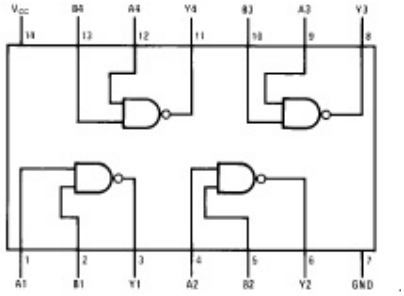

Figure 8: 7400.

IC components present a second challenge for our models. Consider the 7451 IC of Figure 9. It contains two distinct sets of gates which compute and-or-invert of their inputs. We could model both of these functions with one AB-literal as we did for the 7400. A wire bond from the semiconductor die could short with some other internal metal trace, or two metal traces could short. For example, the output of one of the and gates of the first and-or-invert logic could short with the output of an and gate of the second and-or-invert logic. In

this case, both pieces of logic would behave intermittently. The output of the first and-or-invert logic is no longer a true function of its inputs, but is also a function of the inputs of the second and-or-invert logic. In these cases, we model the IC at the gate level with node models which allow shorts. With node models, internal shorts appear as non-intermittent faults.

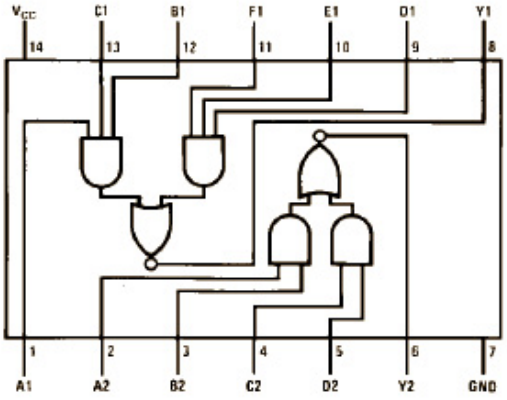

Figure 9: 7451.

Conclusion

DEX is able to model the faulty behavior of components (de Kleer & Williams 1989). The shorted node model relies on a modeled faulty mode. DEX uses a decision-theoretic structure to decide the next action to take: generate more candidate diagnoses, gather additional measurements, replace a component, or generate a new test vector. Although efficiency is not the main goal of DEX, it can troubleshoot most structural failures in simpler circuits, such as c432, c499, c880 in less than a minute on a modern PC.

DEX also includes the ability to replace components, repair nodes, select new measurements, generate new test vectors, and handle component fault modes. In this paper, we focused on identifying shorts between and among nodes. We believe the framework presented in this paper and the notion of causal non-intermittency will generalize to other qualitative models. The concepts of sensed and driven variables and of using the best-probability first candidate generator can be used to reason about structural faults in many domains (e.g., leaks or breaks in pipes and containers, additional linkages in mechanical systems) without incurring the computational complexity previous schemes require.

Acknowledgments

Elisabeth de Kleer, Minh Do, Haitham Hindi, Wheeler Ruml, Peter Struss, Rong Zhou provided many useful comments.

References

Boettcher, C.; Dague, P.; and Taillibert, P. 1996. Hidden interactions in analog circuits. In Abu-Hakima, S., ed., *Working Papers of the Seventh International Workshop on Principles of Diagnosis*. 36–43.

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.

Davis, R. 1984. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence* 24(1):347–410. Also in: Bobrow, D. (ed.) *Qualitative Reasoning about Physical Systems* (North-Holland, Amsterdam, 1984 / MIT Press, Cambridge, Mass., 1985).

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., and Williams, B. 1989. Diagnosis with behavioral modes. In *Proc. 11th IJCAI*, 1324–1330.

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

de Kleer, J. 1992. A hybrid truth maintenance system. PARC Technical Report.

Hamscher, W. C.; de Kleer, J.; and Console, L., eds. 1992. *Readings in Model-based Diagnosis*. San Mateo, Calif.: Morgan Kaufmann.

Mauss, J.; May, V.; and Tatar, M. 2000. Towards model-based engineering: Failure analysis with mds. In *Workshop on Knowledge-Based Systems for Model-Based Engineering, European Conference on AI (ECAI-2000*.

N.A.Snooke, and C.J.Price. 1997. Challenges for qualitative electrical reasoning in automotive circuit simulation. In *Proceedings 11th international workshop on Qualitative Reasoning*, 165–180.

Preist, C., and Welham, B. 1990. Modelling bridge faults for diagnosis in electronic circuits. In *Working Notes First International Workshop on Principles of Diagnosis*, 69–73.

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.

Struss, P.; Malik, A.; and Sachenbacher, M. 1995. Qualitative modeling is the key. In *6th International Workshop on Principles of Diagnosis*.