Modeling when Connections are the Problem

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

Abstract

Most AI diagnostic reasoning approaches model components and but not their interconnections, and when they do model interconnections, they model the possibility that a connection can break, not that two connections may join (e.g., through fluid leakage or electrical short circuit). Two reasons for this limitation are (1) that modeling these interconnection failures could require an exponential number (in the number of interconnections) failure possibilities, and (2) that modeling interconnection failures requires modeling the system at a more precise level which requires far more complex models. 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. We illustrate our approach in the context of digital systems.

1 Introduction

Most of the AI diagnostic reasoning approaches for digital systems [Hamscher *et al.*, 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 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 devices which explicitly determine whether to drive outputs can be modeled as well (such as tri-state and open-collector components).

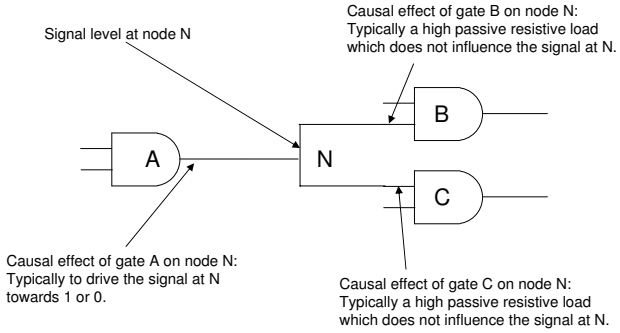

Figure 1: Instead of only modeling a node N with one signal 0 or 1, we model each potential connection which could possibly influence the signal level on the node with a qualitative variable. Each circuit node is modeled with $n + 1$ variables, one for the final signal level, and one for each of the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

This paper demonstrates the approach on a variety of circuits from the standard ISCAS-85 test suite described in [Brglez and Fujiwara, 1985]. All ISCAS-85 benchmark (up to 3512 components) circuits 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 2). The analysis will show that extremely few possible shorts can explain typical symptoms, and those that do are a small fraction of the possible diagnoses. 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 implementation of the GDE/Sherlock [de Kleer and Williams, 1987] probabilistic framework based on the HTMS [de Kleer, 1992]. All component and models are compiled (individually, not in combination) 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 propositional 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.

2 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 and 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 *et al.*, 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 *et al.*, 1995] [N.A.Snooke and C.J.Price, 1997] [Mauss *et al.*, 2000]. One methodology for using multiple variables to represent wires can be found in [Struss *et al.*, 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 node components grows linearly in the worst-case. This approach provides a way to model situations where causality changes such as short circuits, open circuits, and tri-state, open-collector, and expand gates. Structural faults are modeled as any other fault and are integrated within 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.

3 Preliminaries

This basic framework is described in [de Kleer and Williams, 1987; de Kleer *et al.*, 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):

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

The model for the second inverter of Figure 2 is:

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

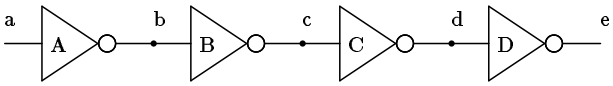

Figure 2: Four sequential inverters.

4 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, in Figure 2, 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 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.

Different logic families will have different thresholds for determining $s(0)$ or $s(1)$. For example, the voltage (v) levels for conventional TTL components are: $s(0)$ corresponds to $v \leq 0.8$, $s(1)$ to $v \geq 2.4$ and $s(x)$ to $0.8 < s(x) < 2.4$. $s(x)$ only arises when measurements are made and is never used within a component model. If some needed component input has been measured to be $s(x)$, its output is not determined by the model (i.e., can be $s(0)$, $s(x)$ or $s(1)$).

With few exceptions, correctly functioning digital devices present a high (little current drawing) 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 usual modeling regime, there is no need to model the behavior of nodes as they just pass along their signals. However, we need explicit models to describe how the sensed digital value of the node is determined from its drivers. Let $R(v)$ be resulting signal at node v and $S(v)$ be the collection of drivers of node v. For example, in Figure 2, $S(b) = \{d(out(A,t), d(in(B,t))\}$. 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:

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

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.*

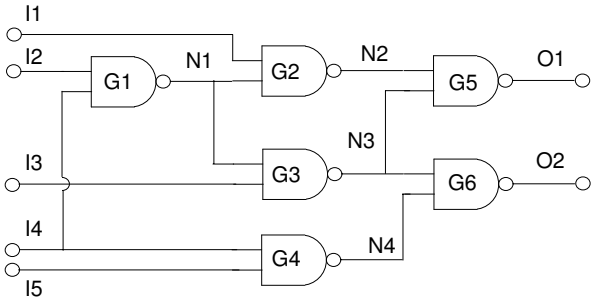

Figure 3: The simplest circuit, c17, from the ISCAS-85 test suite. Inputs are labeled "In," outputs "On," gates "Gn," and corresponding internal nodes "Nn."

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 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.")

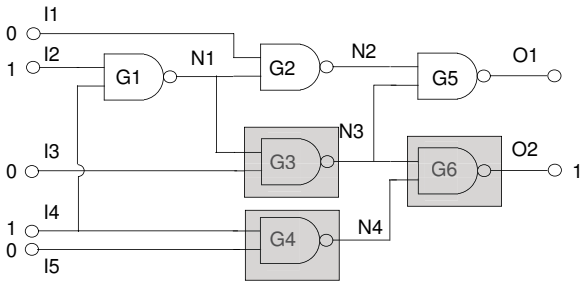

Figure 4: The inputs and symptom added. Output O2 should be 0 but is measured to be 1. All components fail with equal prior probability. After observing the symptom, the more likely faulted gates are G3, G4 and G6 (which are highlighted).

Consider the slightly more complex circuit of Figure 3 — the simplest examples in the ISCAS-85 test suite. Suppose a test vector (I1=0, I2=1, I3=0, I4=1, I5=0) is applied and O2 is measured to be 1 (correct is 0). Figure 4 highlights the only component singlefaults under the simple GDE models. When our node models are included, G5 is the only node singlefault with its input stuck to ground. Therefore, in Figure 5 gate G5 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.

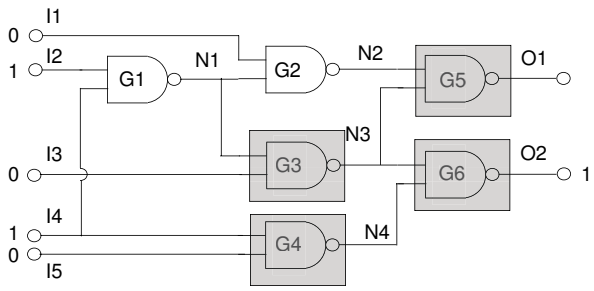

Figure 5: Using the expanded node model, gate G5 could have its input shorted to ground causing the symptom at O2 (should be 0 but 1 is measured). The likely faulted gates are now G3, G4, G5 and G6.

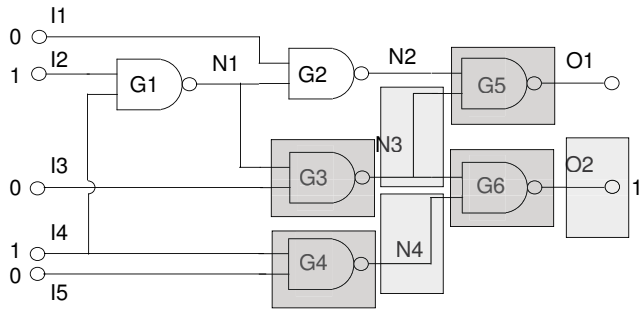

Figure 6: Modeling nodes as possibly faulted components. The highlighted components are more likely to be faulted given the observations. Component faults (in darker shading) are more likely than connection faults (in lighter shading). Lower probability components and nodes are not shaded. Prior fault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

5 Bridges and shorts

Having laid out the causal modeling paradigm the extension needed to model bridges and shorts is relatively straightforward. 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:

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

Table 1: Combined drivers of nodes N1 and N3.

io	gate	drive
output	G1	d(0)
input	G2	d(R)
input	G3	d(R)
output	G3	d(1)
input	G6	d(R)
input	G5	d(R)

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.

Each of the nodes in a shorted 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 N1 and N3 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 N3 of 0 which propagates through G6 to produce 1 which is the observed symptom. Therefore, a N1-to-N3 short is a candidate diagnosis which explains all the symptoms.

Table 2: Combined drivers of nodes I4 and N1.

io	gate	drive
driven	I4	d(1)
input	G1	d(R)
input	G4	d(R)
output	G1	d(0)
input	G2	d(R)
input	G3	d(R)

Notice that the only possible short with node O2 which explains the symptom is a short to power (assuming all other components and nodes are working correctly). As the output driver of G6 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 I4 and N1. The drivers of this combined node are listed in Table 2. So, the signal at the combined node I4-N1 will become 0, which produces an inconsistency with the correct model of the nand gate G1 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, the inconsistency will guarantee that no candidate diagnosis will include the I4-N1 short.

Finally, consider the case where the nodes I5 and O2 are shorted (with same observations of Figure 5). The drivers of the combined node are listed in Table 3. Gates G4 and G6 appear to be in a loop. Thus the signal levels on nodes I5, N4 and O2 cannot be determined by only considering the driver values. Fortunately, the nand model for G6 resolves the ambiguity:

Table 3: Combined drivers of nodes I5 and O2.

io	gate	drive
driven	I5	d(0)
input	G4	d(R)
output	G6	?

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.

	I1	I2	I3	I4	I5	N1	N2	N3	N4	O1	O2
I1								S	S		
I2											
I3						S	S				
I4											
I5						S	S				
N1								S	S		
N2											
N3											
N4										S	
O1											
O2											

$$\neg AB(G6) \rightarrow [d(out(G6,t)) = d(0) \vee d(out(G6,t)) = d(1)].$$

Thus the output driver of G6 cannot be $d(+\infty)$ and thus it can be immediately inferred that shorting I5 and O2 does not explain why O2 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.

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.

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 ISCAS-85 test suite. c432 is a 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	638	5
c499	201	242	562	6
c880	384	442	4959	8

6 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 pos-

sible when modeling gates as purely digital). 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):

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

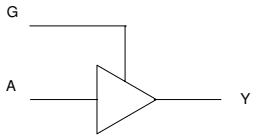

Figure 7: Tri-state Buffer.

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.

7 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 propose measurements 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.

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.

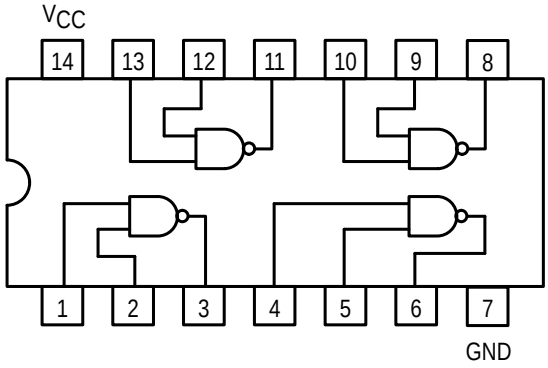

Figure 8: 7400.

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.

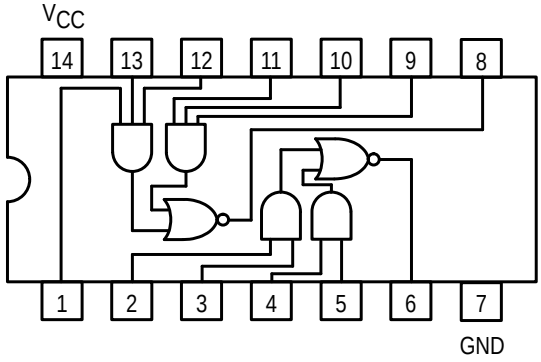

Figure 9: 7451.

8 Algorithm

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. A simplistic implementation of node models would require the construction of $2^n + 3n$ clauses for n-terminal nodes. Shorted nodes can have a large number of terminals. Therefore, DEX constructs these clauses only when they are needed to analyze a leading diagnosis using the consumer architecture of the HTMS.

Intuitively, our candidate generator operates 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 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. Clauses are added to ensure the signal levels at the shorted nodes are all equal (i.e., modeled as one node). One added circumscription 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 added clause circumscribes the conditions under which the node model is applied to the node set. This is important to ensure that if the shorted set is ruled out as a diagnosis, that supersets may be considered as shorts without utilizing any additional clauses or inferential machinery. 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.

Our algorithm for generating the best k diagnoses uses a variant of A* [de Kleer and Williams, 1989; Williams and Ragno, 2002]. For brevity we will only describe the basic algorithm without optimizations and not include the Bayes' rule updates of candidate posterior probability. The candidate diagnosis assigns a behavior mode to every component (including the nodes) of the system. Any GDE-style models which characterize only good behavior are modeled as "G" and "U." Assuming all components fail independently the prior probability of a particular candidate C_l is:

$$p(C_l) = \prod_{m \in C_l} p(m),$$

where $p(m)$ denotes the prior probability of behavior mode m. The objective of the algorithm is to discover the k most probable diagnoses (the "G" modes have higher priority probabilities than the "U" modes). To cast this search in the familiar A* framework we define cost as the negative sum of the logarithms of the fault mode probabilities. Every search node n is represented by a sequence of assignment of modes to components, including the empty model ϕ if that mode has not been assigned. Thus,

$$g(n) = - \sum_{c=m \neq \phi \in n} \ln p(m),$$

$$h(n) = - \sum_{c=\phi \in n} \ln max\{p(x) | x \in m(c)\},$$

where $m(c)$ are the possible modes for component c. h clearly is an underestimate and thus admissible for A*. Algorithm 1 describes the basic algorithm. $CONSISTENT$ is a subprocedure which returns a conflict when the candidate is inconsistent, otherwise it succeeds.

Algorithm 1: Basic A* search for diagnoses

```
begin
    OPEN ← {c_i = φ}, DIAGNOSES ←
    ∅, CONFLICTS ← ∅
    while OPEN ≠ ∅ do
        n ← arg min_{x∈OPEN} f(x)
        OPEN ← OPEN \ {n}
        if ∀_{ng∈CONFLICTS} ng ⊄ n then
            if n assigns modes to all components then
                if ng ← CONSISTENT(n) then
                    CONFLICTS ←
                    CONFLICTS ∪ {ng}
                else
                    DIAGNOSES ←
                    DIAGNOSES ∪ {n}

            else
                pick any s = φ ∈ n
                o ← n \ {s = φ},
                foreach f ∈ m(s) do
                    OPEN ← OPEN ∪ {o ∪ {s = f}}
end
```

All the axioms of Section 4 are included in the initial system description (SD). Nodes are just modeled as components, and the conventional model-based algorithm is adequate. The extension to model shorts requires a modification the function $CONSISTENT$ which adds the necessary shorted axioms relevant to any candidate diagnosis which is ultimately reached in the search. With this extension, the same algorithm identifies the highest probability candidates correctly.

Nodes typically do not short independently — they short to each other. A more realistic prior probability of a candidate is:

$$p(C_l) = p(s_1, \ldots, s_k) \times \prod_{m \in \text{ components of } C_l} p(m),$$

where $p(s_1, \ldots, s_k)$ is the prior probability that nodes s_1 through s_k are shorted together. To generalize the algorithm to shorts, only f and h need to be modified. Let $p^*(n)$ be the maximum probability of any shorts in search node n. This is computed from the assigned short modes as well as the unassigned node modes (which could be shorted).Then,

$$g(n) = - \sum_{c=m \neq \phi \in n \wedge m \neq s} \ln\ p(m),$$

$$h(n) = - \sum_{c=\phi \in n} \ln max\{x | x \in m(c)\} - \ln p^*(n).$$

As the additional term in h is derived from a maximum, it underestimates and the heuristic is admissible. This algorithm can be greatly optimized and h and g better estimated. These details are beyond the scope of this paper.

DEX uses the same probe-selection strategy as [de Kleer and Williams, 1987]. Utilizing our A* algorithm with $k = 10$ best diagnoses, it can troubleshoot all structural failures (all opens and 2-node-shorts) in all the circuits of the ISCAS-85

test suite in less than a minute on a modern PC. The troubleshooting task has become considerably more complex because there are now a far higher number of possible faults to distinguish among. Therefore the average number of measurements needed to isolate a fault is increased for every circuit.

9 Conclusion

In this paper, we focused on identifying connection faults. This approach has been combined with the existing ability to replace components, repair nodes, select new measurements, and generate new test vectors. DEX can now perform these same tasks on circuits which contain connection faults.

A few of the possible directions for research on modeling connections for diagnosis are as follows. First, our models presume that two nodes either short completely or do not influence each other directly (by far the prevalent fault model for digital circuits). Sometimes the correct model lies in between — nodes could be shorted with a small resistance. Second, introducing analog measurements to track down shorts. Third, shorts can change the dynamic behavior of a system (e.g., introduce oscillation). Fourth, connection faults can be intermittent.

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 combined 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.

10 Acknowledgments

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

References

[Boettcher *et al.*, 1996] C. Boettcher, P. Dague, and P. Taillibert. Hidden interactions in analog circuits. In Suhayya Abu-Hakima, editor, *Working Papers of the Seventh International Workshop on Principles of Diagnosis*, pages 36–43. Val Morin, Quebec, Canada, October 1996.

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

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

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

[de Kleer and Williams, 1989] J. de Kleer and B.C. Williams. Diagnosis with behavioral modes. In *Proc. 11th IJCAI*, pages 1324–1330, Detroit, 1989.

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

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

[Hamscher *et al.*, 1992] W. C. Hamscher, J. de Kleer, and L. Console, editors. *Readings in Model-based Diagnosis*. Morgan Kaufmann, San Mateo, Calif., August 1992.

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

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

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

[Raiman *et al.*, 1991] O. Raiman, J. de Kleer, V. Saraswat, and M. H. Shirley. Characterizing non-intermittent faults. In *Proc. 9th National Conf. on Artificial Intelligence*, pages 849–854, Anaheim, CA, July 1991.

[Struss *et al.*, 1995] P. Struss, A. Malik, and M. Sachenbacher. Qualitative modeling is the key. In *6th International Workshop on Principles of Diagnosis*. Goslar, Germany, 1995.

[Williams and Ragno, 2002] B. C. Williams and J. Ragno. Conflict-directed a* and its role in model-based embedded systems. *Journal of Discrete Applied Math, Special Issue on Theory and Applications of Satisfiability Testing*, 2002.