Localizing Transient Faults
Using Dynamic Bayesian Networks

Susmit Jha, Wenchao Li and Sanjit A. Seshia
Department of Electrical Engineering and Computer Science
UC Berkeley
Email: {jha,wenchaol,sseshia}@cs.berkeley.edu

Abstract—**Transient faults are a major concern in today's deep sub-micron semiconductor technology. These faults are rare but they have been known to cause catastrophic system-level failures. Transient errors often occur due to physical effects on deployed systems and hence, diagnosis of transient errors must be performed over manufactured chips or systems assembled from black-box components where arbitrary instrumentation of the system is not possible and hence, the system state is only partially observable. Further, these systems are often composed of components that are third party IP which further adds opaqueness to the system. In this paper, we propose a probabilistic approach to localize transient faults in space and time for such partially observable systems. From a set of correct traces and a failure trace, we seek to locate the faulty component and the cycle of operation at which the fault occurred. Our technique uses correct system traces over monitored components of the system to learn a dynamic Bayesian network (DBN) summarizing the temporal dependencies across the monitored components. This DBN is augmented with different error hypotheses allowed by the fault model. The most probable explanation (MPE) among these hypotheses corresponds to the most likely location of the error. We evaluated the effectiveness of our technique on a set of ISCAS89 benchmarks and a router design used in on-chip networks in a multi-core design.**

I. Introduction

Post-silicon validation involves exercising a manufactured chip in an actual application environment to validate its behavior under various operating conditions. Dramatic increase in design complexity in recent years is making this activity a very expensive proposition. For instance, Intel reported that in 2005 the manpower assigned to design was a third of that assigned to validation [13]. In addition, this validation process can consume 35% of chip development time on average [1]. Bugs are difficult to diagnose in the post-silicon environment due to limited observability, reproducibility and dependence on physical parameters. Electrical bugs – those that occur due to unpredictable electrical conditions – are particularly challenging since they may only manifest under certain operating conditions such as a high clock frequency in certain logic states of the system [9]. Unpredictable electrical bugs are a growing concern in today's deep sub-micron semiconductor technology[1]. The difficulty in diagnosing errors can easily lead to time-to-market slips and silicon re-spins.

The diagnosis problem is made harder by the use of third-party designs in assembled systems such as *system-on-a-chip* (SoC) designs. These third-party designs, called *intellectual property (IP) blocks*, are usually only provided as pre-synthesized components, in order to protect IP. This means a validation engineer needs to debug in a *grey-box* environment in which complete design of the system is not available. The internal operation of some of the components might be unknown and only interface signals are available for observation. If an IP block has already undergone rigorous verification audit, it is typically free of functional errors and manufacturing defects (hard errors). However, an integrated environment can still cause errors on individual components such as a critical path (speed-path) timing violation. These errors manifest themselves as a value arriving at some state-holding element at the wrong time such that it could not be latched. The effect is mostly transient, that is, it only lasts for a cycle, since the error is path dependent and it occurs only under certain operating conditions. This also means very often only a single error trace is available for diagnosis.

In this paper, we consider the problem of debugging transient errors using a set of correct traces obtained from a grey-box system and a single error trace. Our goal is to localize the error both in time and space, that is, the clock cycle in which this error happened and the part of the system where the error happened. We propose a new diagnosis framework which learns the behavior of the system from correct traces over the interface signals. The DBN is then augmented with a fault model for transient errors and the *most probable explanation* (MPE) of the error trace in the augmented model provides the diagnosis. While consistency-based [19], [5] and probabilistic diagnosis [3], [17] techniques have been studied in literature, our technique is an amalgam of the consistency and probabilistic techniques to diagnose transient errors in the grey-box scenario where the system is only partially observable. The main novel contributions made in this paper are as follows -

1) The use of dynamic Bayesian networks to learn a probabilistic abstraction of behavior (transition relation) of a system from its correct traces.
2) Fault model augmented DBNs which allow us to merge consistency based diagnosis with probabilistic techniques.

---

[1]Intel and Matsushita started mass producing 45nm chips in late 2007 and AMD started production of 45nm chips in 2008.

II. Related Work

A number of diagnosis approaches have been proposed in literature. As observed by Console et al [4], these approaches either require models that describe the correct behavior of the system or they need models for the abnormal (faulty) behavior. We discuss some of these techniques below and contrast it with our approach.

Consistency-based Methods. If a system can be described using a set of constraints, then diagnosis can be accomplished by identifying the set (often minimal) of constraints that must be excluded in order that the remaining constraints are consistent with observations. This is the traditional consistency-based reasoning approach [19], [5], [16]. While this approach does not require knowledge of how a component of the system fails (a fault model), it does require the complete specification of the correct system. Consistency based approaches have been recently used to debug permanent faults in circuits [21]. More recently, the advance in satisfiability solving has refuelled interest in consistency-based reasoning approaches [21]. The correct circuit design is specified as a set of constraints in propositional logic and SAT solvers are used to detect the constraints that need to be removed or replaced in order to make them consistent with observations. These constraints which need to be removed correspond to the faulty components. our approach is better suited for localizing transient errors in *grey box* systems due to following reasons.

1) Third party IPs make it impossible to write complete logical description for the system (usual provided as netlist in SAT based techniques). Instead, we learn the transition relation underlying the system as a Dynamic Bayesian Network using correct traces of the chip. This enables *grey-box* diagnosis.
2) Transient errors produce single or few error traces. So, accuracy of consistency based diagnosis would be very low. SAT based approaches will have a large number of possible diagnosis if only few error traces are provided.
3) The SAT-based approaches can not rank different diagnoses. If the number of reported diagnoses is large, reporting all diagnoses is less helpful to a debugging engineer than reporting the set of diagnoses ranked by their likelihood.

Abduction-based methods. The other end of the spectrum is the abductive reasoning approach where a set of hypotheses of abnormal behavior must be introduced to explain the observations. The abductive approach [3], [17] requires a language in which a family of possible explanations can be expressed. The knowledge about failure modes of the system provided by experts are expressed in this language. Abduction works by entailment by identifying all explanations that entail the observation. However, it does not use any model of the correct functioning of the system. We learn a probabilistic model of the correct system and hence, our approach is not entirely abductive. Poole's work [18] on probabilistic Horn abduction proposed a new framework which incorporates both Prolog and Bayesian networks as special cases. A Prolog-like language is used to specify the possible fault models and each hypothesis is associated with some probability. The task of diagnosis is then accomplished using abduction. In contrast to this work, we do not assume any probability distribution over the different possible explanations; instead the underlying model of the system itself is represented as a probabilistic model learnt using the good behaviors of the system.

The abductive approaches are not suitable for debugging transient errors in *grey box* circuits because of the following limitations.

1) Fault models for transient errors can be provided by experts but these fault models are very specific mutations such as bit-flip for a cycle. We do not have a prior information about likelihood of faults in different components. Instead, we rely on the correct traces to derive the likelihood of possible errors.
2) The system to be diagnosed is usually not a black box and we have partial information about the system design as well as access to traces over observed interface signals.
3) Since transient errors are rare, we have abundance of correct traces. Our technique exploits this to learn the DBN describing the correct functioning of the system from the correct traces.

Bayesian networks [14] have been widely used as a probabilistic model of a system for diagnosis. Many of these approaches are anomaly detection techniques. Our approach is based on finding most probable explanation (MPE) of a DBN augmented with fault model and hence, it is not an anomaly detection technique. We use dynamic Bayesian networks to learn the temporal evolution of the system using its correct traces. This is in contrast with Bayesian networks which only represent the interdependence among the components. The key motivation for using dynamic Bayesian networks is to learn a probabilistic abstraction of the underlying state machine of the circuit to be diagnosed. Further, fault models for transient errors can be easily incorporated in DBNs.

III. Diagnosis Approach

The key insight used in our technique is to *use the abundance of correct traces of the circuit* to localize the fault causing the single error trace.

Though the system description of the circuit is not completely available, it is possible to run tests on the circuit to generate a number of traces of the system which are *correct*, i.e., the system does not fail on those tests. This can be used to *learn a probabilistic description of the correct behavior of the circuit*. Secondly, it is possible to use expert guidance to synthesize accurate fault models for the transient errors. These fault models define the space of possible hypothesis which, together with the learnt probabilistic description of the circuit, can explain the error trace.

Two sources of knowledge can be exploited:

- partial knowledge about correct functioning of the system available as the probabilistic description learnt from error traces and
- the expert knowledge available as fault models.

The diagnosis of transient errors in circuits can be best addressed using an amalgam of consistency-based reasoning (to exploit the learnt model of the correct system) and abduction (to exploit the availability of fault models).

The probabilistic model representation used by our technique is Dynamic Bayesian Networks (DBN). An DBN is a Bayesian network that represents a sequence of variables. These sequences could be time-series (for example in speech recognition) or sequences of symbols (for example protein sequences). The hidden Markov model and the Kalman Filter can be considered as the most simple dynamic Bayesian networks. A detailed discussion of Dyanamic Bayesian Networks is presented in Murphy's PhD thesis [12].

We now sketch our approach in rest of the section and identify the key components of our technique. In the next section, we describe these components in detail.

A. Knowledge Representation

We use dynamic Bayesian networks (DBN) to learn the temporal behavior of the circuit using the correct simulation traces, and thus, construct a probabilistic model describing the correct operation of the circuit. In addition, our approach makes use of a *fault model* that specifies how the normal operation of the circuit is modified in the event of an error. The fault model is used to construct a new DBN that is augmented with information on how the probabilistic dependencies change in the presence of a fault. In principle, our approach can work with any fault model. However, the focus of this paper is transient errors caused by combinations of electrical and logical conditions. We identify two examples of transient faults and observe that they can be modeled as a bit-flip.

- *Delay fault:* This fault is used to model a signal that changes its value at an incorrect time. In a sequential (stateful) circuit, it can be used to model a value being latched at the wrong time. This manifests as the value of the latch being flipped. This fault is common on speedpaths (critical paths) in a circuit when the clock speed is high. For our context, it suffices to model a delay fault in terms of its effect as a transient bit flip.
- *Single event upset (SEU):* In this fault, a state-holding element latches the opposite value of what it is supposed to get during a single cycle of circuit operation. Given a sequential circuit, this transient error might change the next state at a given time step, thus causing the resulting trace to diverge from the correct trace after this step. We do not have the complete description of circuit for diagnosis, but if we did, the location of the SEU would be the first point of divergence from the expected (correct) trace. Diagnosing the locations of single event upset can be used to make the system more robust by hardening these locations logically or physically.

Our approach can also handle transient errors with multiple bit flips at the expense of increased run-time for DBN augmentation and inference. In general, any fault model can be used to augment DBNs but more complicated fault models would increase the computational cost for computing the most probable explanation. We use a simple fault model of bit-flip to explain our approach.

Figure 1 shows the transition function of a simple circuit with state variables X_1, X_2, X_3. The primed state variables represent the next state value of the variables. The logical transition function is represented using a $0/1$ table in Figure 1. The entry for $X_1 X_2 = 00$ and $X_1' = 0$ is 1 means that X_1 is 0 in next state if X_1 and X_2 are 0 in current state. The gray box indicates that we cannot observe the variable X_1. Figure 2 shows the result after we apply the structure learning algorithm and update the parameters given a set of traces $\{(x_2^1, x_3^1), (x_2^2, x_3^2), \ldots\}$ and apply smoothening. Smoothening is done by assuming an uniform Dirichlet prior distribution on the conditional probabilities to avoid cases where valid transitions are missing from the training data. In our example, $P(X_2' = 1 | X_1 = 0)$ is 0.01 because the actual circuit can not have such a transition in any correct trace and hence, none of the learning traces would have $X_2' = 1, X_1 = 0$. But smoothening would assign some small probability to this since it might be the case that this transition was valid and missing from the set of correct traces used to learn the DBN.
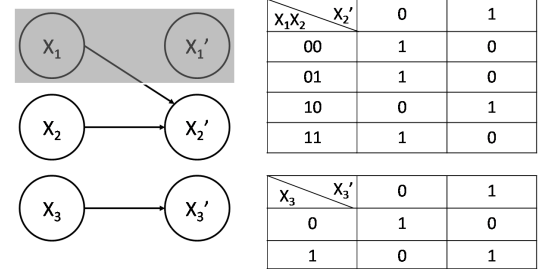

$X_1 X_2$ \ X_2'	0	1
00	1	0
01	1	0
10	0	1
11	1	0

X_3 \ X_3'	0	1
0	1	0
1	0	1

Fig. 1. Original Transition Function

X_2 \ X_2'	0	1
0	0.7	0.3
1	0.99	0.01

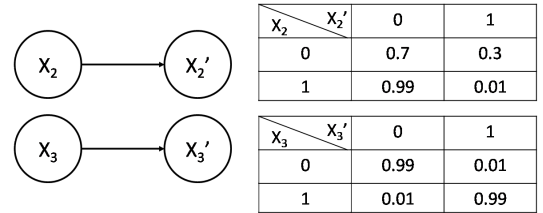| X_3 \ X_3'	0	1
0	0.99	0.01
1	0.01	0.99

Fig. 2. Probabilistic Abstraction of the Original Transition Function as a DBN

With the learnt DBN as a probabilistic description of the transition relation, we can now extend it with a fault model and set up for diagnosis. Our fault model is the following. We say that at any cycle, if a fault occurs to a latch X, X takes the opposite value of what it is supposed to get. Suppose in the original sequential circuit, $x = \delta_X(\pi(X), i(X))$, where δ_X is the deterministic next state update for X, $\Pi(X)$ are the

parents of X (the set of latches that feeds X), and $I(X) \subseteq I$ is the set of inputs that feeds X ($i(X)$ is an evaluation on $I(X)$). Under this fault model, $x' = \delta'_X(\pi(X), i(X), e_{X,t})$, where $e_{X,t} = 1$ means there is a fault at X at time t.

Since the learnt DBN is effectively a probabilistic abstract transition function over the observed latches, the effect of a fault at latch X is that X takes the complement value with the original probability. Formally, given a node X and its parents $\Pi(X)$, we augment the dependence of X by a binary error node E_X by adding an edge from E_X to X, such that

$$P(X = 1|\pi(X), E_X = 1) = P(X = 0|\pi(X))$$
$$P(X = 1|\pi(X), E_X = 0) = P(X = 1|\pi(X))$$
$$P(X = 0|\pi(X), E_X = 1) = P(X = 1|\pi(X))$$
$$P(X = 0|\pi(X), E_X = 0) = P(X = 0|\pi(X))$$

Figure 3 shows the augmented DBN for the same example in Figure 2.

$E_2X_2 \backslash X_2'$	0	1
00	0.7	0.3
01	0.99	0.01
10	0.3	0.7
11	0.01	0.99

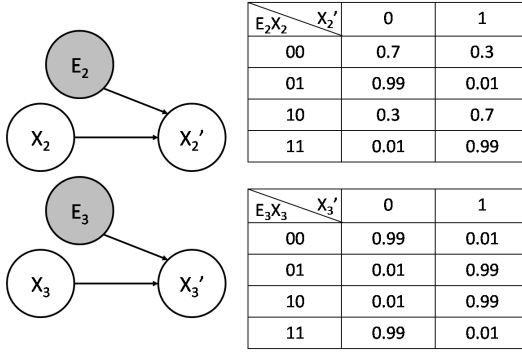| $E_3X_3 \backslash X_3'$	0	1
00	0.99	0.01
01	0.01	0.99
10	0.01	0.99
11	0.99	0.01

Fig. 3. Augmented DBN with Error Nodes

B. Inference

The fault models are used to define a space of hypothesis which can be used to explain the observed error trace. Inference step in our diagnosis technique requires us to compute the probability of observing the error trace under different hypotheses. We do this by augmenting the learnt probabilistic model (DBN) with the fault model by suitably modifying the conditional probability tables of the DBN and then computing the likelihood of producing the error trace. The error hypothesis which is most likely to produce the error trace is the most probable explanation of the error.

In the above example, the fault could be in any of the two variables at some clock cycle. Figure 4 shows the unrolled DBN given an error trace and a fault location hypothesis. The fault location hypothesis which is most likely to produce the error trace is the MPE and is reported as the most likely fault location.

IV. DBN LEARNING AND INFERENCE

A *sequential circuit* C_s is formally modeled as a tuple $\langle I, O, L, \delta, \rho \rangle$, where I is the set of input signals, O is the set of output signals, L is the set of state variables (latches) that induce the state space 2^L, $\delta : 2^I \times 2^L \to 2^L$ is the next state

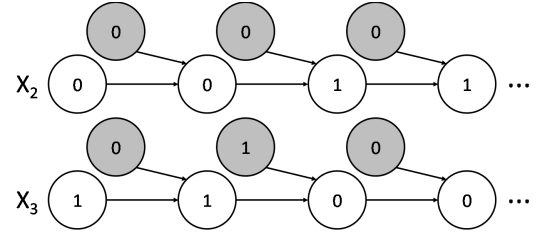Fig. 4. MPE Evidenced on an Error Trace and a Location Hypothesis

function of C_s, ρ is the output function. A sequential circuit is usually initialized to some state s_i. If $|\delta(s, i)| = 1 \ \forall s \in 2^C$ and $i \in 2^I$, then C_s is deterministic. In this paper, we consider deterministic circuits.

Assume we can only observe $L_o \subseteq L$ latches. A trace τ of length l is then a finite sequence of states $s_0, s_1, \ldots, s_{l-1}$, where $s_t \in 2^{L_o}$. A trace can be a correct trace or an error trace.

A *Bayesian network* (BN) B is a tuple $\langle G, \theta \rangle$. G is a directed acyclic graph (DAG). A node X in G is a random variable. An edge in G from X_a to X_b indicates X_b is (potentially) conditionally dependent on X_a. In our framework, X is a binary variable $\in L$. θ is the set of parameters that defines the conditional probabilities amongst variables.

Given a set of correct traces over the observed variables L_o, we learn a DBN using these traces. The DBN is over the observed variables and structural learning is done to determine the dependence edges in the DBN.

A. Structure Learning

Structure learning is the problem of finding the Bayesian network B that best fits some given data D. This is commonly done by choosing the free parameters θ that maximizes the posterior probability $P(\theta|D)$. $P(\theta|D)$ can be approximated by the Bayesian Information Criterion (BIC) [20] or the likelihood-equivalent Bayesian Dirichlet metrics (BDe) [8]. In BIC,

$$log(P(\theta|D)) \approx L(D|\theta) - \frac{|\theta|}{2} log(|D|) \qquad (1)$$

where $L(D|\theta)$ is the log-likelihood of the observation in the estimated model. The regularization term $-\frac{|\theta|}{2} log(|D|)$ in BIC lessens the chance of overfitting.

We use REVEAL [10] with BIC for learning the DBN structure. The REVerse Engineering ALgorithm (REVEAL) works by using the *mutual information* measures to learn the structure of the dynamic Bayesian network. The mutual information $M(X, Y)$ between variables X and Y is defined as

$$M(X, Y) = H(X) + H(Y) - H(X, Y) \qquad (2)$$

More details on learning DBN can be found in [6], [10], [12].

Once the structure is learnt, we can then update the parameters θ of the network. We assume a uniform Dirichlet prior distribution [7] on the conditional probabilities to avoid cases where valid transitions are missing from the training data. This essentially assigns a small non-zero probability to

the unseen transitions. For all the observed nodes, we can update the conditional probabilities as follows,

$$P(x|\pi(X)) = \frac{N_{x,\pi(X)} + \alpha_{x,\pi(X)}}{N_{\pi(X)} + \alpha_{\pi(X)}} \quad (3)$$

where x is an evaluation of variable X, $\pi(X)$ is an evaluation on the parents $\Pi(X)$, $N_{x,\pi(X)}$ is the configuration count for $(x, \pi(X))$, and $\alpha_{x,\pi(X)}$ is the positive hyper-parameter that determines the strength of the regularization.

Once the DBN representing the correct traces is constructed, we modify it to augment it with the fault model. The key idea is to introduce new nodes in the DBN which are binary domain decision variables for whether a particular location and time-step is faulty or not. The mutation caused by the fault is reflected by suitably modifying the probability table of the DBN. For a single bit-flip fault model, the effect of a transient error at latch i is i will take the opposite of its initial value.

B. Inference and Diagnosis

Given the augmented DBN with error nodes $\{E_X\}$ and an error trace τ on the observable variables, the diagnosis problem is an instance of statistical inference of finding the *Most Probable Explanation* (MPE). MPE is the problem of finding the instantiation of the Bayesian network that has the highest probability given the observed evidence.

$$argmax \ \{P(\{e_X\}|\tau) = \frac{P(\tau|\{e_X\})P(\{e_X\})}{P(\tau)}\} \quad (4)$$

That is, we want to find an assignment to the error nodes such that the posterior probability is maximized. Given an error trace τ with length k, we first unroll the DBN $k-1$ times and then ask for the MPE. For a large graph, MPE computation can be expensive. In fact, MPE is NP-hard [11].

Computing MPE is relatively inexpensive for the SEU model. For each fault location (latch/cycle pair) hypothesis X_h, t_h, we can simply compute the probability of the observations under this hypothesis, denoted as

$$P(E_{X_h,t_h} = 1, \{E_{X,t} = 0 | X \neq X_h, t \neq t_h\}, \tau).$$

We assume the same $P(E_{X,h} = 1)$ for all possible fault locations. This is a parameter to the algorithm. Let $\{Y_1, \ldots, Y_n\}$ be the set of random variables (including the error nodes) in the unrolled network. For each error hypothesis, we calculate this probability

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i|\pi(Y_i)) \quad (5)$$

The MPE then corresponds to the hypothesis with the largest $P(y_1, \ldots, y_n)$.

V. EXPERIMENTAL RESULTS

We evaluated our diagnosis approach on a set of 10 IS-CAS89 circuits and a router [15] used in on-chip networks in a multi-core design.

The set of 10 ISCAS89 benchmarks [2] included 8 benchmarks which are controllers and hence, DBN can be used to
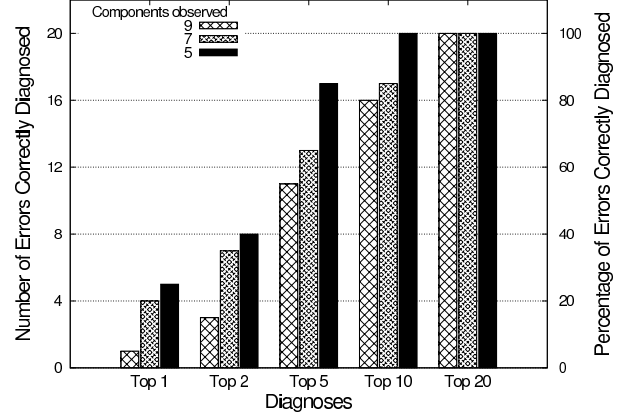

Fig. 6. Accuracy of diagnosis with number of DBN nodes

learn probabilistic abstraction of the underlying state-machine of the controller. A correct simulation trace of length 200 was used to learn the DBN. This trace was observed by simulating the circuit using random input vectors. To evaluate our approach, we performed a set of fault injection experiments.

In a single *fault injection experiment* on a benchmark, we randomly selected 5 latches for constructing the DBN. Single bit-flip fault was injected at one of these latches in exactly one clock cycle. An error trace of length 100 was obtained from the faulty circuit. We repeated this 25 times by inserting faults at each of the 5 latches - and randomly selecting 5 different clock cycles to insert the fault for each latch. So, we had 25 error traces from each fault injection experiment on a benchmark.

Further, we repeated the fault injection experiment on each benchmark 10 times by randomly selecting 5 different latches. Thus, we had 250 error traces for each of the 10 benchmarks.

Figure 5(a) shows for each of the 10 benchmarks, the accuracy with which our approach identified the injected faults. Our tool outputs a ranked list of fault locations (latch/cycle pair). Figure 5(a) shows, for each of the 10 benchmarks, how the injected faults are ranked by the tool. For example, for the benchmark s444, 155 of the 250 injected faults were ranked 1, 17 faults were ranked 2, 13 were ranked 3 and 36 were ranked between 4 and 10.

Figure 5(b) shows data aggregated over all the 10 benchmarks. From this plot, we see that the accuracy of the diagnoses increases only 5.7% while going from considering the top 5 ranked diagnoses to the top 10 diagnoses.

We also did a case study on a chip multiprocessor (CMP) router which is described in [15]. We inserted errors in 5 different latches in the input-controller component of the router out of a total of over 100 latches. Each error trace was 56 cycles long and the errors were inserted at 4 different time steps for each location. Thus, a total of 20 error traces needed to be diagnosed. For building the DBN, we chose 5,7 or 9
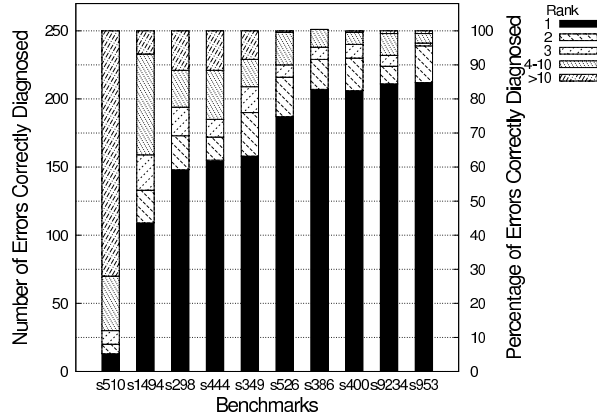
(a) Rank of correct diagnoses
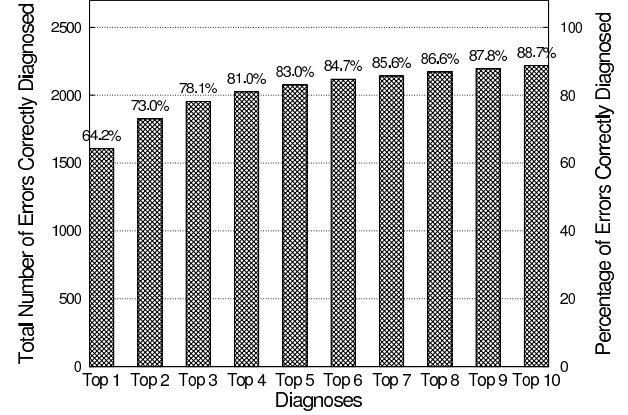
(b) Accuracy of top 10 diagnoses

Fig. 5.   Results on ISCAS Benchmarks

components to build a DBN which included the component with failure. The training data for the DBNs was a correct trace of length 232. The results are presented in Figure 6. The top 5 diagnoses output by the tool included over 50% of the injected faults. Prior information about the error can be used to construct DBNs over smaller number of components and this example illustrates that our technique can exploit the prior knowledge. It yields higher accuracy with fewer components.

VI. CONCLUSION

In this paper, we have proposed a practical approach to diagnose transient error using dynamic Bayesian networks to learn and represent the model of partially observable circuit system. Our error localization technique can point to the pair of observed system variable and time-cycle to which the error first propagates to. Our work is essentially a first step in combining consistency and probabilistic approaches to diagnose difficult to reproduce errors occurring in deployed systems which have only limited observability.

VII. ACKNOWLEDGMENT

REFERENCES

[1] M. Abramovici, P. Bradley, K. Dwarakanath, P. Levin, G. Memmi, and D. Miller. A reconfigurable design-for-debug infrastructure for socs. In *DAC '06: Proceedings of the 43rd annual conference on Design automation*, pages 7–12, New York, NY, USA, 2006. ACM.

[2] F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. *Circuits and Systems, 1989., IEEE International Symposium on*, pages 1929–1934 vol.3, May 1989.

[3] L. Console, L. Portinale, and D. T. Dupr'e. Using compiled knowledge to guide and focus abductive diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 8:690–706, 1996.

[4] L. Console and P. Torasso. A spectrum of logical definitions of model-based diagnosis. *Comput. Intell.*, 7(3):133–141, 1991.

[5] J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnosis and systems. *Artificial Intelligence*, 56, 1991.

[6] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. 1998.

[7] T. J. Harald Steck. On the dirichlet prior and bayesian regularization. In *NIPS 2002*, pages 697–704.

[8] D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. In *Machine Learning*, pages 197–243, 1995.

[9] D. D. Josephson. The manic depression of microprocessor debug. In *ITC '02: Proceedings of the 2002 IEEE International Test Conference*, page 657, Washington, DC, USA, 2002. IEEE Computer Society.

[10] S. Liang, S. Fuhrman, and R. Somogyi. Pacific symposium on biocomputing 3:18-29 (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures.

[11] M. Littman. Initial experiments in stochastic search for bayesian networks. in procedings of the sixteenth national conference on artificial intelligence. (1999) 667-672.

[12] K. P. Murphy. *Dynamic bayesian networks : representation, inference and learning*. PhD thesis, UC Berkeley, 2002.

[13] P. Patra. On the cusp of a validation wall. *IEEE Des. Test*, 24(2):193–196, 2007.

[14] J. Pearl. Bayesian networks. pages 149–153, 1998.

[15] L.-S. Peh. Flow control andmicro-architecturalmechanisms for extending the performance of interconnection networks. phd thesis, stanford university, august 2001.

[16] B. Peischl and F. Wotawa. Model-based diagnosis or reasoning from first principles. *IEEE Intelligent Systems*, 18(3):32–37, 2003.

[17] D. Poole. Explanation and prediction: an architecture for default and abductive reasoning. *Comput. Intell.*, 5(2):97–110, 1989.

[18] D. Poole. Representing diagnostic knowledge for probabilistic horn abduction. *Readings in model-based diagnosis*, pages 467–473, 1992.

[19] R. Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.

[20] G. Schwartz. Estimating the dimension of a model. annals of statistics 6, 461-464 (1978).

[21] A. Smith, A. Veneris, and A. Viglas. Design diagnosis using boolean satisfiability. *ASP-DAC*, 0:218–223, 2004.