Board-Level Fault Diagnosis using Bayesian Inference*

Zhaobo Zhang†, Zhanglei Wang‡, Xinli Gu‡ and Krishnendu Chakrabarty†
†ECE Dept., Duke University, Durham, NC ‡Cisco Systems, Inc., San Jose, CA

***Abstract*—Increasing integration densities and high operating speeds are leading to subtle manifestations of defects at the board level. Board-level functional test is therefore necessary for product qualification. The diagnosis of functional failures is especially challenging, and the cost associated with board-level diagnosis is escalating rapidly. An effective and cost-efficient board-level diagnosis strategy is needed to reduce manufacturing cost and time-to-market, as well as to improve product quality. In this paper, we use Bayesian inference to develop a new board-level diagnosis framework that allows us to identify faulty devices or faulty modules within a device on a failing board with high confidence. Bayesian inference offers a powerful probabilistic method for pattern analysis, classification, and decision making under uncertainty. We apply this inference technique by first generating a database of fault syndromes obtained using fault-insertion test at the module pin level on a fault-free board, and then use this database along with the observed erroneous behavior of a failing board to infer the most likely faulty device. Results on a case study using an open-source RISC system-on-chip highlight the effectiveness of the proposed framework in terms of fault-localization accuracy and correctness of diagnosis.**

I. Introduction

Diagnosis of board-level failures is required to improve product yield and accelerate time-to-market during manufacturing. However, as hardware designers push technologies to the limit, defects-per-million rates continue to increase, and the cost of system test and field service escalates following the "rule of ten"; see Table I [1]. It is therefore necessary to detect and diagnose most failures at an early stage. However, some defects inevitably escape detection until functional test is performed on the board. Fault isolation is very difficult when a system-level functional test fails. Consequently, an effective board-level test and diagnosis strategy is urgently needed.

Test and diagnosis strategies vary depending on the assembly stage. At the chip level, most of the tests are structural tests and all chip I/Os, clocks, voltage and temperatures are controllable; diagnosis is relatively easy compared to that at the board/system level. At the board level, both structural and functional tests can be performed. Test access port (TAP)-based instructions are commonly used to control board-level tests, since the automatic test equipment (ATE) is not available at this level. In system test, all boards in one or more chassis and all devices on the boards are exercised together under functional test sequences. Many devices are only accessible via control and status registers (CSR)s (e.g. external memory) resulting in access difficulty and test time increase [2], [3].

TABLE I
Rule of ten in test economics [1].

Testing	Chip Testing	Board Testing	System Testing	Out in Field
Cost of Test	1	10	100	1000

The difficulty of board-level test and diagnosis can be attributed to the following three reasons [2]. First, ASICs that pass ATE test might still fail during board test. One reason for this kind of failure is that test conditions on ATE are different from that in board test. Additional noise, power, and clock-supply quality issues in board test affect the behavior of ASICs. Second, signal integrity test at the board level becomes a challenge as data rate continuously increases. The signals transmitted through complex telecommunication boards have reached data rate of 10 GHz today. Signal integrity issues such as crosstalk, reflection, dielectric loss make board-level test and diagnosis challenging [4]. Third, the performance of external memory devices on a board and their connectivity quality to the board are unknown until board test. Due to the lack of test structures provided in a memory device, diagnosis of memory at the board level is often difficult.

To solve these challenges, several diagnosis strategies have been published in the literature. Built-in self-test (BIST) technologies are mainly adopted for board/system test and diagnosis. In [5], a hierarchical system self-test architecture was proposed and a standard BIST interface was constructed at the system level that allows re-use of tests designed into chips and boards throughout the life cycle. In [2], an approach for extending the functionalities of structural test techniques to the board/system level was presented. Besides BIST-based structural test, inference-based diagnosis techniques have been published as a solution for board-level diagnosis. In [6], a model-based inference engine was built on the basis of the correct operation of a device, and improvements to this basic inference engine were presented as well. In [7], an automated diagnosis system named MonteJade was developed based on a combination of model-based and probabilistic approaches.

In this paper, a board-level fault diagnosis method based on Bayesian inference is presented. In order to apply Bayesian inference, we artificially create several faulty scenarios and compare the behavior of a real malfunctioning board with behaviors obtained from artificial faulty scenarios. Bayesian inference is then used to derive logical conclusions and determine the likely faulty devices. This inference-based diagnosis approach does not depend on any particular BIST design.

The remainder of this paper is organized as follows. Section II introduces fault-insertion test and fault model used in our diagnosis framework, especially for creating the database for

*The work of Z. Zhang and K. Chakrabarty was supported in part by a research grant from Cisco Systems, Inc.

Bayesian inference. Section III provides the basics of Bayesian inference and its application to board-level diagnosis. Section IV presents the development of the diagnosis framework and potential directions for improving diagnostic resolution. Section V presents the design of experiments and experimental results for Open RISC 1200 system-on-chip (SoC). Section VI concludes the paper and future directions are presented in Section VII.

II. Fault-insertion Test and Fault Model

Fault-insertion test (FIT) is a promising method for board/system reliability test and diagnosis coverage measurement [8], [9]. It facilitates the timely release of a quality diagnostic program before manufacturing and provides feedback on the fault tolerance of a complicated system. Hardware FIT verifies error detection, error handling, and recovery on a board by intentionally inserting wrong values at the pin/logic level. Artificial faults are used to model the effects of manufacturing defects and in-field intermittent and transient errors [8], [10]. In Fig. 1, a block diagram depicts a board with forcing circuitry and switches added. The forcing circuitry is used for inducing artificial error stimulus at the ASIC and memory I/O to force the net under test to an incorrect logic level, and the switches are used for fault activation. Hardware FIT can be implemented via a number of methods and the choice depends on what is available within the design; typical methods include probe-based FI, boundary-scan-based FI and multiplexer-based FI [11], [12].

An artificial faulty scenario can be created by activating a pin-level fault during runtime. Therefore, the response of a board to a known fault induced by FIT can be “learned”. The board responses under different faulty scenarios are recorded and used for future comparison with responses of an actual failing board. By learning board behaviors under various artificial faults, we construct *a priori* knowledge for Bayesian inference. To take advantage of Bayesian inference in diagnosis, we also need to define the type of faults to be used with FIT.

Generally, hardware can encounter three types of errors in operation: hard failure, transient failure, and intermittent failure [13]. Hard failure is a physical defect that is permanent and readily reproducible. It is relatively easy to screen these using BIST features. Transient failure is a one-time fault that cannot be recreated. This fault is usually attributable to high-energy particles (cosmic radiation, alpha particles) striking latches or memory cells and altering circuit logic values, which may potentially result in a soft error [14]. Intermittent failure is an occasional fault that occurs as a result of frequency, voltage or other environmental condition that is reproducible under the same condition. This type of fault is usually attributable to corner conditions in the design. While high coverage of hard failure can be achieved using state-of-the-art method, transient failure is a major concern for high-availability systems. This paper is focused on transient faults that are not easily detected by traditional DFT features. The bit-flip fault model is used to mimic the subtle effects of transient faults.

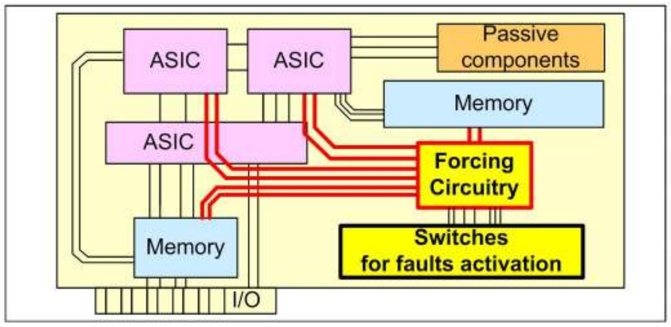

Fig. 1. Illustration of fault-insertion test.

III. Bayesian Inference and Application to Board-level Diagnosis

Bayesian theory provides a unified and intuitively appealing approach for drawing inferences from observations and *a priori* beliefs. It builds on Bayes’ theorem, shown in (1), in which F_1 through F_m are a set of mutually exclusive and jointly exhaustive events. The parameter R_k is an event that depends on the occurrence of F_1,...,F_m. $P(F_1)$ through $P(F_m)$ are probabilities ($prior$s) of the occurrence of each event F_1 through F_m. $Prior$s are *a priori* beliefs obtained before observation on event R_k is made. The conditional probability $P(R_k|F_i)$ is the probability of occurrence of event R_k given that event F_i occurs. The occurrence probability of event F_i given that event R_k occurs is denoted by $P(F_i|R_k)$. This probability is defined as $posterior$, which depends on the *a priori* belief of the occurrence probability of event F_i and the observation on event R_k.

$$P(F_i|R_k) = \frac{P(R_k|F_i)P(F_i)}{P(R_k)} = \frac{P(R_k|F_i)P(F_i)}{\sum_{j=1}^{m} P(R_k|F_j)P(F_j)} \quad (1)$$

Bayesian inference has been successfully used in power-circuit and analog-circuit diagnosis [15], [16]. There is much interest today in applying inference-based methods to digital system diagnosis [6], [7]. In this paper, Bayesian inference is used for board-level fault diagnosis. Suppose that we want to locate the faulty ASIC in a malfunctioning board. To apply Bayesian inference to this practical problem, let F_1 through F_m in (1) be a set of faults at the pin level of ASICs. These faults are candidates for the root cause of board failure. Let R_k be the kth observation point. Specifically, observation points are observable registers (e.g. CSRs) on the board. The artificial faults and the observation points are denoted on a real board shown in Fig. 2. Note that $prior$ $P(F_i)$ is the *a priori* occurrence probability of fault F_i; $P(R_k|F_i)$ is the probability of an error occurring on the kth register given that fault F_i occurs. $P(F_i|R_k)$ is the occurrence probability of F_i given an error in the kth register. This is the $posterior$ based on the observation on the kth register.

To understand this inference process better, suppose that there are two possible faults (F_1 and F_2) on a failing board and one register (R_1) is observable. The problem is to determine the more likely fault resulting in the board failure (out of F_1 and F_2). Assume that when fault F_1 occurs, the occurrence probability of error on register R_1 is 0.8; when fault F_2 occurs, the occurrence probability of error on register R_1 is 0.4. Note that we find an error on register R_1 on the actual failing board. In this scenario:

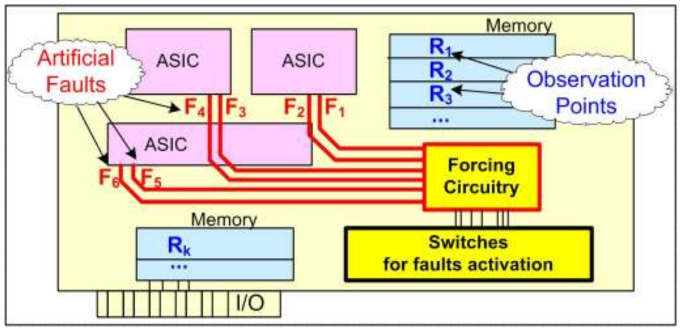

Fig. 2. FIT for applying Bayesian-inference.

$$P(F_1) = P(F_2) = 0.5$$
$$P(R_1|F_1) = 0.8,\ P(R_1|F_2) = 0.4$$
$$P(F_1|R_1) = \frac{P(R_1|F_1)P(F_1)}{P(R_1|F_1)P(F_1)+P(R_1|F_2)P(F_2)} = 0.67$$
$$P(F_2|R_1) = \frac{P(R_1|F_2)P(F_2)}{P(R_1|F_1)P(F_1)+P(R_1|F_2)P(F_2)} = 0.33.$$

Therefore, the probability that F_1 occurs is 0.67 and the probability that F_2 occurs is 0.33. Based on the given conditional probabilities and the observations from the bad board, we infer that fault F_1 is the more likely suspect fault causing board failure.

In Bayesian inference-based diagnosis, we initialize the *prior* for each fault in an equiprobable manner, i.e., $1/m$. The parameter m is the number of all the possible faults. There is only one observable register in the above example. When multiple registers are observable, the $posterior$ is updated iteratively by observation on each register. The calculated $posterior$, based on the observation of register R_k, is used as the $prior$ of the next calculation based on the observation of register R_{k+1}. Thus, there is no requirement on the number of observation points (observable registers). Decisions can be made based on existing observations and when new observation is available, the inference results can be updated. This adaptive attribute is one of the main advantages of Bayesian inference. Another advantage of using Bayesian inference in diagnosis is efficiency, since it eliminates the debug time for devices that are determined to be fault-free by the inference engine. Moreover, an inference-based engine such as this allows for an automated diagnostic process. Assuming that the Bayesian framework is accurate, diagnosis results can be directly gathered by sending the fault syndrome and database to the inference engine. The fault syndrome refers to the observations on registers from bad boards, and database includes all the conditional probabilities needed in the $posterior$ calculation, which are learned from fault-insertion test.

IV. Diagnosis Framework

The Bayesian inference based diagnosis framework consists of a learning step and the diagnosis process. In the learning step, pin-level faults are intentionally inserted to a fault-free (reference) board. The inserted faults and corresponding behaviors of the board are recorded. Conditional probabilities ($P(R_k|F_i)$) are computed and saved to a database. In the second step, according to the conditional probabilities in the database and bad-board syndrome, the probability of each artificial fault occurring or being the root cause is inferred using Bayes' formula and the fault with the highest occurrence probability is deemed to be the most-likely candidate. Therefore, the ASIC where the most suspect fault resides is considered to be the faulty ASIC leading to board failure. This framework can be hierarchically used to diagnose faulty components inside ASICs as well.

TABLE II
An example of fault syndromes.

	F_1	F_2	F_3	F_4
R_1	1	0	1	1
R_2	1	1	1	1
R_3	0	1	0	1
R_4	0	0	0	1
R_5	1	1	1	0

A. Learning step

The purpose of the learning step is to study board responses to the artificial faults. In (1), we can see that the conditional probability $P(R_k|F_i)$ is the key to *posterior* calculation. The computation of this conditional probability is the main task in the learning step. More details of the conditional probability calculation are presented below.

In FIT, an artificial fault can be inserted on the board by turning on fault-insertion logic. During fault simulation, logic values of observable registers are recorded, and a set of comparison results of register values is called the fault syndrome. The logical values of registers in fault simulation are compared with golden values. For each register, the comparison result is either a match or a mismatch. Specifically, if there is a match, we record a "0"; if there is a mismatch, we record a "1". Thus, the fault syndrome can be denoted by a binary vector. The length of this vector is equal to the number of observable registers n. Therefore, the number of all possible comparison outcomes is 2^n. A fault syndrome table is shown in Table II. There are 5 observable registers on the board and four pin-level faults are inserted. For example, when pin-level fault F_1 is inserted, the fault syndrome is (1,1,0,0,1), which means that detectable errors appear in register R_1, R_2 and R_5 in the presence of pin-level fault F_1.

Thus far, fault syndromes in the presence of different pin-level faults have been obtained, but we cannot conclude that the probability of error occurring on register R_1 is 1 when pin-level fault F_1 occurs. The reason for this is that the effect of defects on modern boards is subtle and error responses are not always reproducible. To get an accurate error-occurrence probability, fault simulation needs to be run multiple times and an average is used to compute the conditional probability. Therefore, we repeat fault simulation multiple times, and accumulate the count of errors occurring on each register. The relative frequency of error is equal to the count of error occurrence divided by the total number of simulations. When the simulation times is large enough, our premise is that the relative frequency of an error converges to the corresponding conditional probability. For example, in Table III, fault simulation is performed 100 times for bit-flip fault F_1. During these 100 simulation runs, the bit-flip fault is inserted at various time instants during test execution, and the number of error occurrences in Register R_1 is found to be 90. Thus, the conditional probability $P(R_1|F_1)$ is 0.9. In this learning step, by

TABLE III
DATABASE OF CONDITIONAL PROBABILITIES.

	F_1	F_2	F_3	F_4
R_1	90/100	2/100	99/100	71/100
R_2	82/100	68/100	21/100	71/100
R_3	0/100	92/100	10/100	65/100
R_4	4/100	0/100	7/100	100/100
R_5	86/100	46/100	51/100	30/100

recording the faults and corresponding syndromes, the database of conditional probabilities is obtained. This database is reusable for the diagnosis of different failing boards.

B. Diagnosis Process

In the diagnosis step, the probability of each fault being the root cause is inferred using Bayes' formula. According to the fault syndrome on the bad board, if an error occurs on register R_k, the possibility of fault F_i to be root cause can be directly calculated using (1). The conditional probability $P(R_k|F_i)$ can be computed from the database constructed in the first step. If no error occurs on register R_k, the probability that fault F_i is the root cause is calculated using (2):

$$P(F_i|\overline{R_k}) = \frac{P(\overline{R_k}|F_i)P(F_i)}{\sum_{j=1}^{m} P(\overline{R_k}|F_j)P(F_j)} \tag{2}$$

where $P(\overline{R_k}|F_i) = 1 - P(R_k|F_i)$. The probability $P(F_i|\overline{R_k})$ denotes the occurrence probability of F_i when the observation from register R_k does not contain an error. The computation is similar for the case of error occurrence.

Due to the adaptive attribute of Bayesian inference, we know that the $posterior$ can be updated when new observations are made. Therefore, if the number of observable registers on the board is q, the $posterior$ of each fault will be updated q times. The $posterior$ in the final update is considered to be the probability of the fault being the root cause. In the first round, the $prior$ is initialized in an equiprobable manner. Afterwards, the $prior$ is updated by the $posterior$ calculated in the previous round.

Fault syndromes for a failing board might also not be reproducible due to the subtle manifestations of transient faults. To describe the fault syndrome of the failing board more realistically and get accurate diagnosis, simulation on the failing board is performed multiple times as well. The final $posterior$ is the average of all the $posterior$ computed from each fault syndrome.

Due to the lack of observability and limitation of the inference model, some faults cannot be distinguished from one another, resulting in an ambiguity group. In order to retain diagnosis accuracy, we not only consider the fault with the highest $posterior$ as the root cause, but also several faults with high $posterior$ values. A discussion on how to improve diagnosis accuracy and shrink the ambiguity group size is presented in Section V.

V. EXPERIMENT DESIGN AND RESULTS

A. Experiment Design

To evaluate the proposed framework, experiments are performed on the OpenRISC 1200 SoC [17]. The OpenRISC 1200 is a 32-bit RISC with Harvard microarchitecture, 5 stage integer pipeline, virtual memory support (MMU) and basic DSP capabilities. It consists of a CPU, memory management units, data cache, instruction cache. Flash memory, SRAMs, audio/video/ethernect connections, etc. The system block diagram is shown in Fig. 3. Although there are differences between a SoC and a manufactured board, the SoC here is used to establish the feasibility and effectiveness of the proposed method. In this simulation framework, the functional units and memory cells on the SoC can be viewed as the ASICs and memory devices mounted on a real board. In future, the proposed method will be applied to a manufactured board.

The CPU block is the critical part in this system. Therefore, artificial faults are inserted at the output pins of six functional modules inside the CPU, namely ALU, Ctrl, Insn_fectch, Operand_MUX and Gen_PC, as shown in Fig. 4. These six modules are the suspect modules causing system malfunction. We assume that registers in six memory units are observable, namely IMMU, DMMU, DC_top, IC_top, register files and special purpose register (SPR). Four functional tests, *Dhry*, *Basic*, *CBasic* and *Multi*, are pre-loaded in flash for diagnosis. *Dhry* is a synthetic benchmark workload, *Basic* and *Cbasic* include most basic instructions, and the test *Multi* mainly consists of multiply instructions [17].

In our experiments, a total of 23 pins from six functional modules are selected for fault insertion. In the 23 pins, some of them are bits of data buses and the others are flag signals. More details of the pin-level fault selection technique can be found in [10]. The fault syndrome is a combination of observations from 17 registers. Fault simulation with pin-level faults is used for constructing the database of error-occurrence probabilities for a specified pin-level fault. A malfunctioning system is created by inserting bit-flip faults within one of the six functional modules. Fault simulation is repeated multiple times for each fault due to the subtle effects of transient faults. The number of simulation runs can be adjusted according to the length of the test sequences. In our experiments, fault simulation is repeated 100 times for each fault. The time instants of fault insertion are randomly selected during test execution.

B. Experimental Results

In this subsection, experimental results for the OpenRISC1200 are presented. All the experiments were performed on a pool of state-of-the-art servers running Linux. The Verilog simulator for all simulations is VCS (Y-2006.06-SP1-16). Four functional tests were applied to the OpenRISC system. A single run of the tests takes 0.5 min, 1 min, 4 min and 4.5 min respectively. According to the experiment design above, fault simulation is repeated 100 times for each fault. The total simulation time is nearly 600 hrs. Bayesian inference is implemented using Matlab and it takes only a few seconds to obtain the inference results.

Due to the irreproducible nature of transient fault and practical limits on the number of simulations, faults at the pins of an actual faulty module may not always have the highest occurrence probability, which necessitates the inclusion of faults with smaller occurrence probability in an ambiguity group. In our experiments,

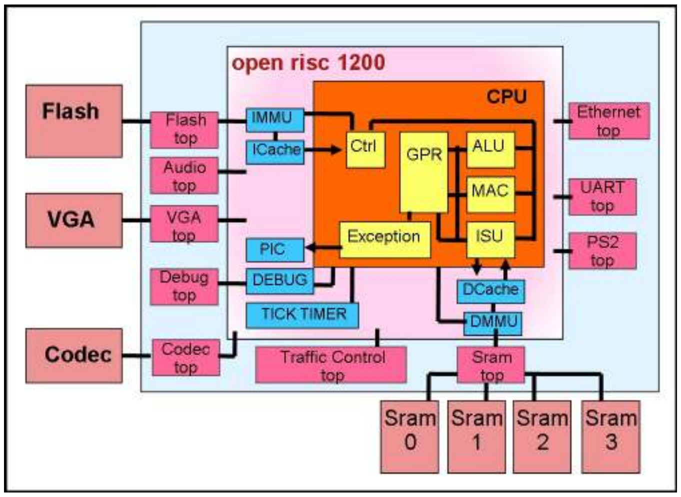

Fig. 3. Block Diagram of the OpenRISC 1200 SoC.

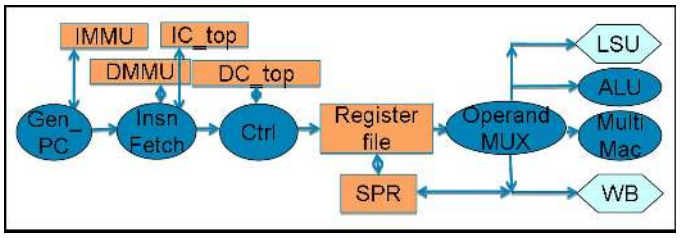

Fig. 4. Block Diagram of the CPU in OpenRISC 1200.

if the $posterior$ value of a fault is greater than 0.05 and this fault ranks in the list of faults with the top five $posterior$ values, we place this fault in the ambiguity fault group. The threshold for selection depends on the number of fault candidates, but a characterization of the optimal threshold is left for future work.

Table IV shows functional modules and the corresponding faults at the ports. These faults are used for locating faulty modules inside the system. For example, F_{19}, F_{20}, F_{21} are faults inserted at ALU output ports for learning. According to the diagnosis results, if F_{19} has the highest occurrence probability, our conclusion is that the ALU module is the most likely faulty module causing system failure. Faults at the ports of the actual faulty module are called target faults. Diagnosis is considered to be correct, when at least one target fault falls in the ambiguity fault group.

Table V shows diagnosis results for six different failing cases using the first functional test *Basic*. Each row is related to one failing case. The first column in Table V lists the actual faulty module in the failing system. The second column is the ambiguity fault group. Faults are placed in the ambiguity group if they meet the criteria defined above. Target faults are highlighted in bold. The third column shows the highest rank of target faults out of 23 fault candidates. The fourth column indicates the correctness of diagnosis. From Table IV, we can see 5 cases out of 6 are correctly diagnosed based on only one functional test. Unfortunately, if the faulty module is Ctrl in the failing system, we cannot correctly locate it using test *Basic*.

TABLE IV
FUNCTIONAL MODULES AND CORRESPONDING FAULTS AT THE PORTS.

Functional modules	Corresponding faults at the ports
ALU	F_{19}, F_{20}, F_{21}
Ctrl	F_{9}, F_{10}, F_{11}, F_{12}
IF	F_{5}, F_{6}, F_{7}, F_{8}
O_MUX	F_{13}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}
Multi	F_{22}, F_{23}
Gen_PC	F_{1}, F_{2}, F_{3}, F_{4}

TABLE V
DIAGNOSIS RESULTS USING FUNCTIONAL TEST *Basic*.

Faulty module	Ambiguity fault group	Rank	correctness
ALU	$\mathbf{F_{21}}$,F_{6},$\mathbf{F_{20}}$,F_{4},F_{3}	3	Yes
Ctrl	F_{4},F_{3}	7	No
IF	F_{20},F_{17} ,F_{13},F_{18},$\mathbf{F_{5}}$	1	Yes
O_MUX	$\mathbf{F_{17}}$,$\mathbf{F_{18}}$,F_{4},F_{3}	3	Yes
Multi	$\mathbf{F_{23}}$,F_{17},F_{14},F_{13},F_{11}	5	Yes
Gen_PC	$\mathbf{F_{1}}$,$\mathbf{F_{4}}$,$\mathbf{F_{3}}$	1	Yes

TABLE VI
DIAGNOSIS RESULTS USING FUNCTIONAL TEST *Dhry*.

Faulty module	Ambiguity fault group	Rank	correctness
ALU	F_{15},$\mathbf{F_{21}}$,F_{8},F_{2},F_{3}	3	Yes
Ctrl	$\mathbf{F_{11}}$,F_{2},F_{3}	3	Yes
IF	F_{21} ,F_{15},F_{20},F_{2}	9	No
O_MUX	$\mathbf{F_{15}}$,F_{21},F_{2},F_{3},F_{5}	5	Yes
Multi	—	—	—
Gen_PC	$\mathbf{F_{2}}$,$\mathbf{F_{3}}$	1	Yes

Likewise, diagnosis results for six failing cases using the second functional test *Dhry* are shown in Table VI. In Table VI, the system with the faulty IF module cannot be correctly diagnosed, but the faulty Ctrl module can be distinguished by *Dhry*. We get blank results for faulty multiplier case, because the internal defects inserted in the multiplier do not cause errors on any of the observable registers for the *Dhry* test. In other words, *Dhry* cannot be used to detect/diagnose the faults occurring inside the multiplier module.

In Table VII, two functional tests are used to improve diagnostic resolution. Here diagnostic resolution refers to the size of the ambiguity fault group. Four correctly diagnosed cases using the *Basic* test are selected for this experiment. First, the test *Basic* is applied to the system and ambiguity fault groups obtained under this test are shown in the second column in Table VII, which are the same as the results shown in Table V. Subsequently, the *Dhry* test is applied to distinguish/rank faults in the ambiguity fault group obtained based on the previous test. In the second round inference, the candidates are only those faults in the previous ambiguity fault group. If the $posterior$ value of a candidate is greater than 0.2 and it ranks in top-3 highest $posterior$, it is placed in the second-round ambiguity group, shown in the third column. The correctness is shown in the fourth column in Table VII. The average of ambiguity group size shrinks from 4.25 to 2.75 without loss of diagnosis correctness.

Fig. 5 shows the improvement of diagnosis accuracy for the Ctrl module when multiple functional tests are used. In Fig. 5, the y-axis is the rank of a corresponding fault of the Ctrl module, and the x-axis refers to the functional tests applied for diagnosis. To diagnose a system with faulty Ctrl module, we start by using the *Basic* test. The rank of fault F_{11} is 7. Next we apply the *Dhry* test based on the first test, and fault F_{11} is now rank third. When the third test *Multi* is also used, the rank becomes 2. This shows that when multiple tests are applied in diagnosis, diagnosis accuracy can be improved using Bayesian inference.

TABLE VII
IMPROVED DIAGNOSIS USING THE *Basic* AND *Dhry* TESTS.

Faulty module	Ambiguity group in *Basic*	Ambiguity group in *Dhry*	Correctness
ALU	$\mathbf{F_{21}}$,F_6,$\mathbf{F_{20}}$,F_4,F_3	F_4,$\mathbf{F_{20}}$,F_3	Yes
IF	F_{20},F_{17} ,F_{13},F_{18},$\mathbf{F_5}$	F_{20},$\mathbf{F_5}$	Yes
O_MUX	$F_{\mathbf{17}}$,$\mathbf{F_{18}}$,F_4,F_3	$\mathbf{F_{18}}$,F_4,F_3	Yes
Gen_PC	$\mathbf{F_1}$,$\mathbf{F_4}$,$\mathbf{F_3}$	$\mathbf{F_1}$,$\mathbf{F_4}$,$\mathbf{F_3}$	Yes

TABLE VIII
MULTIPLE FAULT DIAGNOSIS USING TEST *Basic*.

Faulty module	Ambiguity fault group	Inferred suspect module
ALU & IF	F_{18} $\mathbf{F_5}$,F_4,F_3	IF, O_MUX,Gen_PC
O_MUX & Gen_PC	$\mathbf{F_2}$,$\mathbf{F_3}$,$\mathbf{F_4}$	Gen_PC
IF & Gen_PC	$\mathbf{F_1}$,$\mathbf{F_2}$,$\mathbf{F_4}$,$\mathbf{F_3}$	Gen_PC
ALU & O_MUX	$\mathbf{F_{20}}$,$\mathbf{F_{17}}$,F_4,$\mathbf{F_{18}}$,F_3	ALU,O_MUX,Gen_PC

We next study the diagnosis of multiple faults using the proposed framework. Two faults inside different functional modules are inserted per trial to mimic the situation that multiple faults exist in a malfunction system. The database of conditional probability is the same as before. The diagnosis results are shown in Table VIII. They are not as accurate as before, since the conditional probabilities used in the computation are obtained from single fault simulation. Multiple fault diagnosis is difficult and computationally expensive, as the volume of fault hypotheses grows exponentially with the number of faults in the system. A promising solution is to partition the overall system into subsystems, within which there is likely to be a single fault. This partitioning enables the application of single-fault diagnosis, which has only linear complexity, to the subsystems without the need to handle the exponential hypothesis explosion.

VI. CONCLUSION

We have presented a board-level Bayesian inference-based fault diagnosis strategy. This approach offers two key advantages. First, it does not depend on the structural test circuity, which eliminates the requirement of building standard BIST features for all devices on a board. Second, by learning the behavior of board/system under different faulty scenarios, the inference engine can automatically perform diagnosis without manual effort. Experiments have been performed on an open-source RISC SoC to illustrate the use of Bayesian inference in fault diagnosis. Our results show that Bayesian inference-based diagnosis is a promising strategy to tackle the challenges involved in quality assurance of complex boards. By being adaptive, automated and self-learning, the approach is expected to lead to a new breakthrough in board/system diagnosis.

VII. FUTURE WORK

In the construction of our diagnosis framework, several parameters have yet to be optimized. Our future work includes selecting output pins that are the most capable of distinguishing faulty modules, and choosing observation points (registers) to create the least-correlated fault syndromes in the presence of different faults. We will also focus on selecting the most effective observation intervals instead of saving all logic values at runtime. In order to verify the effectiveness of the proposed method in practice, experiments will be performed on manufactured boards in the near future.

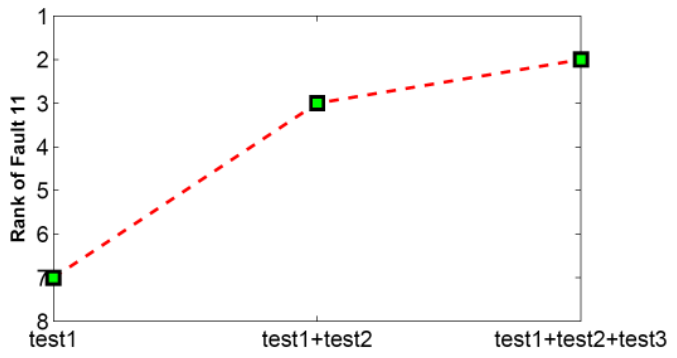

Fig. 5. Improvement of diagnosis accuracy using more tests for the Ctrl module.

REFERENCES

[1] Y.-T. Lin, "Economic design for manufacturing system test and field maintenance," in *Southeastern Symposium on System Theory*, 2005, pp. 40–44.
[2] T. Vo, Z. Wang, T. Eaton, P. Ghosh, L. Young, W. Wang, H. Jun, D. Singletary, and X. Gu, "Design for board and system level structural test and diangosis," in *IEEE International Test Conference*, 2006, pp. 1–10.
[3] S. Martin, R. Bleck, C. Dislis, and D. Farren, "The evolution of a system test process," in *IEEE International Test Conference*, 1999, pp. 680–688.
[4] Z. Mu and K. Willis, "SI and design consideratins for Gbps PCBs in communication systems," in *Electrical Performance of Electronic Packaging*, 2001, pp. 287–290.
[5] S. Pateras and P. McHugh, "BIST: A test & diagnosis methodology for complex, high reliability electronics systems," in *Autotestcon*, 1997, pp. 398–402.
[6] C. O'Farrill, M. Moakil-Chbany, and B. Eklow, "Optimized reasoning-based diagnosis for non-random, board-level, production defects," in *IEEE International Test Conference*, 2005, pp. 173–179.
[7] L. Barford, V. Kanevsky, and L. Kamas, "Bayesian fault diagnosis in large-scale measurement systems," in *IEEE Instrumentation and Measurement Technology Conference*, 2004, vol. 2, pp. 1234-1239.
[8] B. Eklow, A. Hosseini, K. Chi, S. Pullela, T. Vo, and H. Chau, "Simulation based system level fault insertion using co-verification tools," in *IEEE International Test Conference*, 2004, pp. 704–710.
[9] B. Huang, M. Rodriguez, M. Li, and C. Smidts, "On the development of fault injection profiles," in *Reliability and Maintainability Symposium*, 2007, pp. 226–231.
[10] Z. Zhang, Z. Wang, X. Gu, and K. Chakrabarty, "Physical defect modeling for fault insertion in system reliability test," in *IEEE International Test Conference*, 2009, pp. 1–10.
[11] R. Sedmak, "Boundary-scan: beyond production test," in *IEEE VLSI Test Symposium*, 1994, pp. 415–420.
[12] S. Chau, "Fault injection boundary scan design for verification of fault tolerant systems," in *IEEE International Test Conference*, 1994, pp. 677–682.
[13] C. Constantinescu, "Impact of deep submicron technology on dependability of VLSI circuits," in *International Conference on Dependable Systems and Networks*, 2002, pp. 205–209.
[14] N. Karimi, M. Maniatakos, A. Jas, and Y. Makris, "On the correlation between controller faults and instruction-level errors in modern microprocessors," in *IEEE International Test Conference*, 2008, pp. 1–10.
[15] F. Liu, P. K. Nikolov, and S. Ozev, "Parametric fault diagnosis for analog circuits using a bayesian framework," in *IEEE VLSI Test Symposium*, 2006, pp. 1–6.
[16] B. Ye, Z. Luo, W. Zhang, and C. Piao, "Fault diagnosis for power circuits based on SVM within the Bayesian framework," in *World Congress on Intelliggent Control and Automation*, 2008, pp. 5125–5129.
[17] "OR1200 Architecture," http://www.opencores.org/project,or1k.