2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

A Methodology for SAT-based Electrical Error Debugging during Post-silicon Validation

Binod Kumar[1], Masahiro Fujita[2] and Virendra Singh[1]
Indian Institute of Technology Bombay, India[1], University of Tokyo, Japan[2]
Email: {binodkumar, viren}@ee.iitb.ac.in, fujita@ee.t.u-tokyo.ac.jp

***Abstract*—Satisfiability(SAT)-based error diagnosis during pre-silicon verification has been quite successful. Recently, SAT-based debugging for error localization in the post-silicon environment has also been attempted. Analysis of silicon debug data is challenging because of the limited observability of the internal signals, which leads to elongating the error localization process. Therefore, the success of the debug process in the post-silicon scenario depends on effective on-chip tracing of internal signals of the design. This paper proposes a grouping based signal tracing methodology for collecting useful debug data from chip execution. The proposed methodology analyzes logical connectivity in the design netlist for deriving different signal groups from which a limited number of signals are traced. Effective post-silicon debug data is collected with the proposed signal selection scheme and an SAT-based debugging methodology is applied for analysis of these traces. Experiments on benchmark circuits show that the proposed internal signal grouping and the debug process is effective for localizing functional and electrical errors.**

***Index Terms*—Satisfiability, Post-silicon validation, Bit-flips, Error localization, Trace signals.**

I. Introduction

Restricted observability and controllability of the internal design states and lack of failure repeatability of the observed failures are some of the major obstacles during the post-silicon validation of modern integrated circuits [1]. Scan chains can be utilized for a periodic monitoring scheme where in their contents are taken out after some pre-defined intervals [2]. However, for subtle electrical bugs, this scheme becomes insufficient as the scan dumps can not provide necessary debug information. This necessitates DfD (Design-for-Debug) features for continuous tracing of internal signals of the design for a fixed number of clock cycles. Since the area overhead attributed to such tracing mechanisms is limited, the decision of internal signals to be traced becomes a crucial issue for the debug process [1], [3]. This paper proposes a signal tracing methodology to solve the signal selection problem and then utilize satisfiability-based debugging of the obtained post-silicon signal dumps. For SAT-based diagnosis, after creating a SAT instance, we need to apply some constraints to solve it. These constraints assist in the logical inconsistencies in the SAT instance leading to the root-cause discovery. Since post-silicon execution has limited visibility, obtaining constraints for solving the SAT instances is a challenging task and involves trade-offs of debug time, deployed design-for-debug features and required off-chip dumping. The constraints required for SAT-based error localization of data collected from the chip execution are analyzed. With different observability enhancement mechanisms, the constraints required for SAT solving differ. The proposed signal selection methodology is agnostic of the type of error and the deployed observability enhancement scheme. The main contributions of the paper are:

- a methodology is proposed for tracing of appropriate signals for collecting debug data through different observability enhancement schemes.
- a satisfiability based debugging methodology is presented for handling large error traces.

The remainder of the paper is organized as follows. Section II summarizes the related work on post-silicon observability expansion and error localization methodologies. Section III presents the proposed signal selection methodology and its application to different observability expansion techniques. Section IV describes the satisfiability based error localization methodology and the proposed adaptations to it. Section V explains the details of the experiments. Section VI finally concludes the paper with directions on some future research.

II. Related work

A. Post-silicon Observability Enhancement Techniques

The signals utilized for tracing in the post-silicon environment are decided at the design stage. When reconfigurability is intended in the tracing of internal signals, a multiplexed scheme needs to be employed [4], [5]. Majority of the research done in the field of automatic trace signal selection in the last decade pertains to measure of state restorability [1], [5]–[7]. This selection metric computes the ratio of unknown signal states (with the help of principles of logic implication) over traced signal states. However, the metric of state restoration fails to capture other important issues such as ease in error localization/debugging with the assistance of traced and restored signal states. Temporal variability in the tracing of internal signals can be achieved by stitching a group of signals into small chains where in the length of chain decides the tracing frequency. This variability in temporal tracing of internal signals leads to a higher state restoration ratio [1] and different mechanisms to introduce the temporal variation assists in error detection. Some of the signal selection techniques have attempted trace signal selection aimed at error localization in the post-silicon environment [4], [8].

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00085

B. Satisfiability-based Post-silicon Error Localization

Satisfiability methods have been utilized in the process of pre-silicon verification with appreciable success. Applying satisfiability (SAT) during design debugging, Suelflow et al. [9] proposed the usage of UNSAT core (clauses which are responsible for unsatisfiability of a SAT instance) for the purpose of error localization. Yang et al. [10] extended this proposal to post-silicon error traces, based on a hierarchical debugging scheme of RTL design descriptions. A signal selection scheme was proposed based on the UNSAT core obtained from the CNF formula. This method heavily relies on the knowledge of design details as hierarchical debugging scheme is employed. Vali et al. [8] proposed bit-flip detection aware selection of trace signals. Additionally, a SAT-based methodology of localization (of particular flip-flop and flip cycle) for traces of 100 clock cycles is proposed. Their methodology involves unrolling the netlist and computing UNSAT core after forcing constraints (obtained from the trace history) on the CNF expression of unrolled netlist. However, solving the SAT instance for a large design and a larger error trace (consequently unrolling for larger number of cycles) becomes cumbersome. Zhu et al. [3] proposed a sliding window (corresponding to a small number of clock cycles of chip execution) based technique to localize stuck-at constant faults. Debugging is assisted by computation of *backbones* which are set of signals (in the unrolled netlist) that are immutable under the chip execution constraints such as the signal values from the trace buffer or primary input (PI) and primary output (PO) values. Leveraging *backbones* computation, a sliding-window based approach is presented in [11] also for localizing design errors like gate replacements in post-silicon environment.

III. Proposed Signal Grouping Methodology

A. Motivation behind the proposed methodology

As mentioned earlier, for application of SAT-based debugging, chip execution history is needed which can be utilized in creating a Boolean instance for the localization problem. In post-silicon environment, a detailed execution history is impossible to obtain. Thus, to extract important history from the chip execution is of crucial importance. This can be done by off-dumping the contents of on-chip trace buffers. However, whether the collected debug information is profitable or not depends on the trace signals selected for this purpose. We propose a methodology which can be utilized in either static (fixed tracing of signals) or dynamic manner (through usage of multiplexers). The grouping of the signals is aimed at minimizing the number of trace signals required for SAT-based error localization. The proposed methodology explores the logical connectivity among different signals of the design. Such a methodology has been adopted in [2] for determining the position of multiple input signature registers which are used for compacting responses stored in shadow flip-flops of the design. In comparison to a selection methodology [8] aimed at certain kind of errors such as bit-flips, the proposed methodology is agnostic of the targeted error scenarios.

B. Description of the proposed methodology

The methodology segregates all the flip-flops (signals) in the design into certain number of groups. After the formation of signal groups, depending on the trace buffer width, the required number of trace signals can be selected from these signal groups. The proposed methodology consists of two parts- first, the netlist connections are represented by a graph (G) where the flip-flops are represented by nodes, second, a merging procedure is followed to group the candidate signals and finally select trace signals. Typically, each flip-flop has some gates common between its logic cone and logic cone of other flip-flop(s). This overlap of logic cones is quantified as *cone sharing*. For any graph (G), the *cone sharing* is represented as weights of edges between the different nodes. Thus, the weight on the edges of G corresponds to the logic cones of the corresponding nodes measured in terms of the number of gates that are shared between these two cones. The graph, G obtained from the analysis of a netlist (not shown here) is illustrated in Fig. 1 which has eight nodes and eleven edges. Since a trace signal becomes highly useful for debugging if the error propagates to it, we need to find effective flip-flops from this perspective. Conditional error probability based error propagation methods rely on assumption of 0/1 at the inputs of the combinational gates. Thus, we measure the error transmission capability of signals (flip-flops) from the heuristic of logical connectivity, without accounting for sensitization of the errors. One of the reasons behind the proposed approach is that different type of errors (such as bit-flips, stuck-at etc.) have varied propagation conditions. With proposed methodology, we aim to reduce the tracing overhead. There are two inputs to the proposed grouping methodology- the number of groups, which depends on trace buffer width (TBw), and the other is the minimum size of each group, $MinSize$. Both these parameters are to be provided by the design engineer. The edge of G with the largest weight is selected as the initial edge for the formation of first signal group. Since the edge-weight in G represents the logic cone overlap size, the largest edge-weight has a higher probability of error propagation to both the graph nodes.

C. Illustration of proposed methodology

Let's consider the graph, G for illustration in Fig. 1. The edge between node F4 and F7 is selected since this edge has the highest weight, and node F4 and F7 are merged to begin forming the first group. Subsequently, the edge weights in G is updated after these two nodes are merged. This is shown in Fig. 2 where two such groupings are shown. In addition to the merging of F4 and F7, F2 and F5 are merged to form another group. The process of adding additional nodes to the obtained bigger groups continues in a greedy manner by selecting the edge with the largest weight attached to the current bigger group until the size of these groups reaches minimum size.

In Fig. 3, these two groups have been constructed, a grouping procedure is repeatedly performed to merge the remaining nodes of G together using the largest weight on an edge. Node F6 is connected to the node containing F2,F5 and F8 through

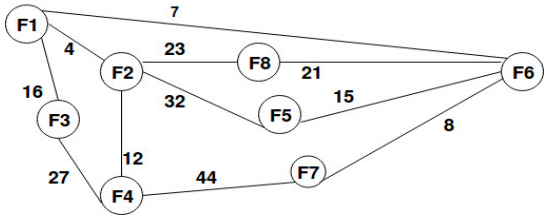

Fig. 1. Circuit graph illustration

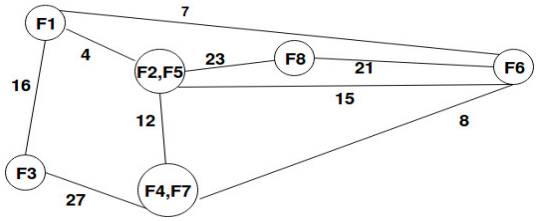

Fig. 2. Illustration of grouping of nodes

an edge where two edge-weights (i.e., 15 and 21) are possible because of the merging. Out of this, the highest edge-weight (i.e., 21) is selected as the final weight for this edge because of the larger extent of overlap.

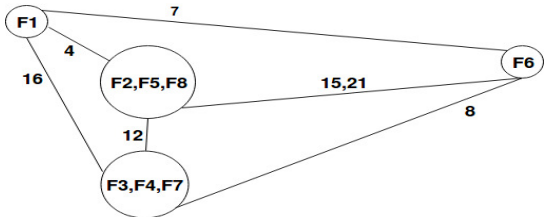

Fig. 3. Final grouping illustration

Suppose, for this graph (G), we have fixed $MinSize$ as 4 and TBw as 2. Since the two groups have 3 nodes in them, both of them need one node to reach the minimum size. The final grouping is shown in Fig. 4 from where 2 flip-flops are to be selected as trace signals. As per Algorithm 1, F4 and F2 are selected for tracing considering a static signal tracing scheme, where in these signals are dumped for a fixed trace buffer depth (denoted by Cy). If a multiplexed signal tracing scheme [5] is available, content of two (or, more) signals from each group can be dumped into the trace buffer. Suppose we

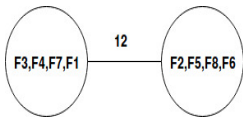

Fig. 4. Two groups at the end of $GroupSignals$ procedure

have two final groups and the trace buffer width (TBw) is more than 2, N_1 signals can be selected from the first group and N_2 from the second group such that sum of N_1 and N_2 equals TBw. A simple distribution of N_1 and N_2 occurs when each one of them is chosen as half of TBw.

D. Discussion regarding design choices

The proposed methodology of trace signal selection is formally presented as Algorithm 1. One of the questions regarding this algorithm is the choice of $MinSize$ parameter. It can be chosen by considering the circuit size in terms of the number of flip-flops. The goal here is to select the trace signals which are able to catch the flip even if it does not occur in their vicinity. Furthermore, once the groups are formed, we need to perform the selection of trace signals which can be done based on the logic connectivity (in terms of edges) of the candidate signals. Since majority of the previous signal selection techniques [1], [5]–[7] focus on restoration of untraced signal states from the traced ones, they do not perform much better than a random signal selection in most of the cases in the experiments. Detailed results regarding impact of various parameters on the error localization process are presented in the experiment section.

Algorithm 1: $GroupSignals$

Input: G, TBw, $MinSize$
Output: $TraceSignals$

```
numGroup ← ∅;
Nodes_G ← nodes in G;
TraceSignals ← ∅;
while numgroup != TBw do
    Edge_hwt ← edge with highest edge-weight;
    N1, N2 ← any 2 nodes of Edge_hwt edge;
    {wt1, wt2} ← weights of edges connecting to N1,
      N2 with the remaining nodes of G;
    Node_m ← Merge nodes {N1, N2} into one node;
    Choose maximum out of {wt1, wt2} for edge
      connecting Node_m to other nodes of G;
    Remove N1, N2 from Nodes_G;
    Add Node_m to Nodes_G;
    TempGroup ← Node_m;
    if size of TempGroup == MinSize then
        numGroup = numGroup+1;
    end
end
for each group g_i in numGroup do
    Rank signals (nodes) of g_i in the decreasing order
      of sum of their edge-weights;
end
TraceSignals ← top signals from each numGroup;
```

E. Temporal variability in post-silicon signal tracing

As mentioned in section II, to increase the post-silicon observability, some flip-flops can be combined into fixed length scan chains [1] which assist in obtaining temporal variability in signal tracing. This needs the usage of shadow flip-flops and the tracing frequency of each signal is determined by the length of the chain. Table I shows a scenario where trace buffer width is 2 (which are denoted by TB_1 and TB_2) and buffer depth is 6 (denoted by the six clock cycles). Let us consider that FFA, FFB, FFC and FFD are four trace signals whose contents are to be dumped into TB_1 and TB_2. Note that FFA_1 denotes the signal state of FFA in the first clock cycle. Since four signals can be traced in this scheme, we need to select four signals from the grouping shown in Fig. 4, F4, F7 and F2, F6 are respectively selected from the final

TABLE I
COMBINATION OF SIGNALS FOR TEMPORAL VARIABILITY

No.	Cycle1	Cycle2	Cycle3	Cycle4	Cycle5	Cycle6
TB_1	FFA_1	FFB_1	FFA_3	FFB_3	FFA_5	FFB_5
TB_2	FFC_1	FFD_1	FFC_3	FFD_3	FFC_5	FFD_5

group(s) because of the ranking based on sum of edge weights of signals inside the clusters. In a static signal tracing scheme, only 2 signals can be dumped for 6 clock cycles, whereas in this scheme, 4 signals (flip-flops) meaning that all the signals are dumped in alternate cycles. Considering the illustration in Fig. 4, FFA and FFB can be represented by F4 and F7 respectively. Similarly, FFC and FFD can be represented by F2 and F6 respectively. Since larger number of internal signals can be covered for signal tracing with this observability enhancement scheme, error localization can be more effective. Note that the dumping frequency is fixed here which is being decided by the length of the smaller scan chains, eliminating the need of any controller [5] for the purpose of dumping.

IV. SAT-BASED POST-SILICON ERROR LOCALIZATION

A. Preliminaries for error localization based on UNSAT core

Given a Boolean formula in Conjunctive Normal Form (CNF), if it is UNSAT, any subset of clauses in the instance that is also unsatisfiable is called as an *UNSAT core* [9], [10]. Let's consider the following unsatisfied CNF formula:

$$\phi = (e + g) \cdot (e + f) \cdot (\bar{g}) \cdot (\bar{e}) \cdot (\bar{f} + \bar{g}) \quad (1)$$

For the above formula, the UNSAT core include $\{(e+g),(\bar{g}),(\bar{e})\}$. Analysis of these clauses leads to the root-cause of the error. For bit-flip kind of errors, we need to localize the infected flip-flop (and the corresponding flip cycle). For other kind of error scenarios such as stuck-at, only particular flip-flop (in the logic cone of which the *sa*-0 or *sa*-1 nets/wires are present) is localized. An accurate localization of these net(s) requires a detailed analysis as is done in [11].

In the post-silicon environment, one challenge is to force constraints on the SAT instance in such way that the resulting CNF formula turns out to be UNSAT. The generated UNSAT core from the unsatisfiable SAT instance hint towards error localization. However, if the error has not propagated to the obtained signal dumps (from traced signals) or primary input/output values, the error localization process is ineffective. For a sequential circuit (C) enrolled for T time-frames, where T is the length of the error trace, the complete Boolean formula is given by following equation:

$$\phi_C = \prod_{i=1}^{i=T} (C_i \cdot PI_i \cdot PO_i \cdot tb_i \cdot resto_i) \quad (2)$$

In the above equation, C_i represents the unrolled circuit in i^{th} time-frame, PI_i and PO_i respectively represent the primary input and primary output values in i^{th} clock cycle. tb_i and $resto_i$ respectively represent the signal values of traced signals and restored signals in i^{th} clock cycle. Note that for each value of i, the number of signals is fixed in tb_i whereas the number of signals is variable in the case of $resto_i$. If the restored signal values are not utilized for creating the Boolean formula (ϕ_C), $resto_i$ is dropped from the above equation. However, we show in the experimental section that usage of state restoration technique assists in obtaining a lesser number of *suspect* candidates. By providing ϕ_C to the SAT solver, the UNSAT core can be obtained which hints towards the error root-cause. Since the sequential circuit needs to be unrolled for T time-frames (i.e., clock cycles), the size of ϕ_C in terms of number of clauses increases which can lead to serious implications on the solver time (and memory usage) for large designs. For example- in the case of $s38584$ circuit, the number of clauses after the netlist is unrolled for 256 clock cycles reaches close to 13.4 million. It has also been observed that a ten times larger SAT instance requires approximately hundred times larger runtime and ten times higher memory usage while solving [8]. Therefore, to perform error localization in the case of large error traces, we break the complete error trace ($ETrace$) into smaller sections of the trace (i.e., chip history execution). The goal is to obtain resulting SAT formula/instance, ϕ_C in such a manner that it becomes relatively smaller leading to reduced requirement of runtime and memory for SAT solving. The number of such partitions would vary depending on the trace length and the debug scenario.

B. Handling large traces for SAT-based error localization

The collected chip execution history, $ETrace$ of length T (consisting of traced signal states and/or PI and PO signal values) can be viewed as a composition of smaller error traces ($eTrace_j$). This can be represented by the following equation:

$$ETrace = \cup_{j=1}^{j=T/n} (eTrace_j) \quad (3)$$

where, n is the number of divisions required. However, while applying constraints on the SAT instance for each division of $ETrace$, we need to know the initial values of the circuit state for each smaller division. Considering reproducible behavior of the bugs, this can be obtained with the help of scan chains. For example- Consider T as 1000 and n as 10, to know the initial state for the second division ($eTrace_2$), the contents of the flip-flops at the end of 100^{th} clock cycle are simply the scan chain values (at 100^{th} cycle) which can be dumped off-line. The complete SAT-based localization analysis is then done for these smaller error traces ($eTrace_j$) leading to the unrolling of the circuit for only 100 cycles compared to the case when the netlist needs to be unrolled for 1000 cycles. Note that instead of having error localization with large trace ($ETrace$), if the smaller error trace(s) are collected, the trace collection needs to be done n times. Furthermore, typically the trace buffer depth is in the range of 1024 clock cycles [1] requiring the netlist unrolling up to that depth.

V. EXPERIMENTAL RESULTS AND OBSERVATIONS

The characteristics of the benchmark circuits are noted in Table II. We used *PicoSAT* [12] as SAT solver for extracting the UNSAT core from the Boolean formulas. We performed localization experiments for the bit-flip and stuck-at errors after analysis for the generated (extracted) UNSAT core

TABLE II
CHARACTERISTICS OF BENCHMARK CIRCUITS

Sl. No.	*Name*	*#PI*	*#PO*	*#FFs*	*#Gates*
$c1$	$softusb$ [13]	34	50	317	4718
$c2$	$s9234$	19	22	228	5597
$c3$	$s13207$	31	121	669	7951
$c4$	$s15850$	14	87	597	9772
$c5$	$s38417$	28	106	1636	22179
$c6$	$s38584$	12	278	1452	19253

whenever the SAT instances are unsatisfiable. All experiments were conducted with a machine having 4GB RAM and Intel i5 processor operating at 2.40 GHz. We observed in our experiments with bit-flip errors that enforcing only the *traced signal states* on the CNF expressions of the circuits turns the Boolean formula into SAT (i.e., the corresponding Boolean formulas are satisfiable) for most of the cases. Unless specified otherwise, the error trace length (T=Cy) is taken to be 100 cycles which means the netlist has to be unrolled 100 times for creating the Boolean formula ϕ_c for SAT solving.

A. Comparison with other trace signal selection methods

As stated before, restorability based selection techniques do not perform much better than any random signal selection. We observed that the proposed method of grouping requires lesser trace signals compared to the restorability based signal selection in [7]. This also leads to savings in SAT solving time as the formula size is smaller in case of the proposed method. Due to lack of space, the detailed results of comparison with [7] are not presented here. Instead, we present results of error localization (in the form of number of suspects) when the grouping methodology is varied and two different ways of applying chip execution history can to create the SAT instance.

B. Error localization results with fixed signal tracing

We considered TBw as 32 and 64 for the localization experiments and five different choices of these trace signals are analyzed. These choices are marked as S_1, S_2, S_3, S_4 and S_5 corresponding to one signal each from 32 groups, two signals each from 16 groups, four signals each from 8 groups, one signal each from 64 groups and two signals each from 32 groups. S_1, S_2, S_3 correspond to TBw as 32 and S_4, S_5 correspond to TBw as 64. The chip execution constraints were applied in two flavors- *a) traced signal states and PI, PO values*, *b) restored (and traced) signal states and PI, PO values*. The notations in Table III denote the constraints enforced and the considered signals chosen from the groups for a specific TBw. For example- S_3^a represents the list of 32 signals when 4 signals are collected from 8 groups and traced signal states and PI, PO values are enforced on the CNF of the unrolled circuit to create the SAT instance, ϕ_C. The values in Table III represent the average number of suspects (flip-flops) when a random bit-flip at a particular clock cycle is injected in the netlist. The reported values are obtained from 10 iterations of error injection and subsequent error localization experiment. We obtained exact localization of the flip-cycle along with some other suspect flip-cycles for all the circuits in the error injection experiments. Note that the flip-cycle varies in all the experiments unlike a fixed flip-cycle utilized in the experiments in [8]. We observed that enforcing PI,PO values and $restored\ signal\ states$ yields always lesser number of suspects compared to the case when PI,PO values and $traced\ signal\ states$ are utilized as constraints in satisfiability solving. Same trend is seen for stuck-at errors. For s9234 circuit, TBw is taken as 8 in addition to 16. In this case, 32 signals are not needed because of the small circuit size. So, S_1, S_2 are S_3 are adapted for 8 and 16 signals. Due to space limitations, we do not report the time spent and memory usage in UNSAT core extraction for all cases. We observed that without the proposed signal grouping methodology, larger trace buffer width (such as 128 or 256) are required to achieve error localization with SAT-based methodology.

TABLE III
AVERAGED NUMBER OF SUSPECTS DUE TO BIT-FLIP ERRORS

ckt.	S_1^a	S_1^b	S_2^a	S_2^b	S_3^a	S_3^b	S_4^a	S_4^b	S_5^a	S_5^b
$c1$	5.4	3.5	5.3	3.2	5.4	3.5	5.5	3.3	5.2	3.1
$c2$	7.3	2.5	7.6	2.2	7.7	2.3	-	-	-	-
$c3$	4.3	4.1	4.5	3.9	4.7	4.0	3.9	3.5	3.8	3.6
$c4$	3.4	3.5	3.3	3.2	3.3	3.3	3.3	3.1	3.0	2.9
$c5$	5.4	3.2	5.3	3.4	5.7	3.1	4.6	3.2	4.5	3.2
$c6$	6.7	4.0	6.1	3.9	5.7	4.8	4.5	4.1	4.0	3.8

We injected random *stuck-at* fault at one of the nets in the netlist. The flip-flops (signals) in the outward logic cone of this net are computed and we aim to localize these signals. We assume that once these signals can be identified, a subsequent analysis may be performed to localize the particular net. Although this analysis is required, the hints from the UNSAT core can definitely assist the debug engineer. With different grouping of signals, the selected signals for tracing differ which lead to variation in the number of suspects.

TABLE IV
AVERAGED NUMBER OF SUSPECTS DUE TO STUCK-AT ERRORS

ckt.	S_1^a	S_1^b	S_2^a	S_2^b	S_3^a	S_3^b	S_4^a	S_4^b	S_5^a	S_5^b
$c1$	7.6	3.4	7.7	3.8	7.2	3.9	6.7	3.5	6.4	3.1
$c2$	5.6	4.4	5.7	4.8	5.6	2.9	4.3	4.5	4.7	4.3
$c3$	4.1	3.7	4.3	3.9	4.7	4.1	3.9	3.5	3.8	3.3
$c4$	6.6	4.3	6.3	4.3	6.1	4.2	5.8	4.0	5.6	4.1
$c5$	9.6	5.5	9.4	5.4	9.3	5.0	8.7	4.3	8.6	4.4
$c6$	8.8	5.9	8.4	5.4	8.6	5.2	6.4	4.9	6.6	4.9

C. Localization with temporally variable signal tracing

We consider small chains (each of length two which means that two flip-flops are stitched together) with the scheme outlined in [1], [14]. For TBw as 16, 32 signals can be dumped as explained in Section III-E. Table V shows the averaged number of suspects for such an observability mechanism for bit-flip errors. TBw of 16 is sufficient to obtain lesser suspects and not much benefits are obtained for TBw to 32.[1] Since the circuits in Table II can be considered as blocks within a complex

TABLE V
AVERAGED SUSPECTS DUE TO BIT-FLIP (TEMPORAL TRACING)

ckt.	S_1^a	S_1^b	S_2^a	S_2^b	S_3^a	S_3^b	S_4^a	S_4^b	S_5^a	S_5^b
$c1$	5.6	2.9	5.5	2.8	5.9	2.7	4.9	2.7	4.8	2.7
$c3$	4.9	2.7	4.7	2.6	4.9	2.6	4.6	2.6	4.4	2.5
$c4$	5.4	3.1	5.0	2.9	5.1	2.8	4.9	2.9	5.0	2.8
$c5$	6.7	3.3	6.7	3.4	6.3	3.1	6.0	3.1	6.0	3.0
$c6$	6.9	3.6	6.7	3.8	6.6	3.5	6.3	3.5	6.4	3.5

[1]Here, S_1, S_2, S_3 are for TBw as 16 and S_4, S_5 are for TBw as 32. Results are not reported for $s9234$ circuit as fixed signal tracing is sufficient.

SoC, with a fixed trace buffer width of 64, more such blocks can be traced. We observed the similar trend in the averaged number of suspects for stuck-at errors when a temporally variable signal tracing (with a tracing frequency of half) is employed. Thus, in addition to an appreciable increase in signal restoration with this observability enhancement scheme [1], SAT-based error localization also improves.

D. SAT-based localization in large post-silicon error traces

Error traces of length (T) equal to 500 and 1000 clock cycles are dividend into 5 divisions resulting into smaller error traces of 100 and 200 clock cycles respectively. The averaged number of suspects is respectively represented by $n(S)$ and $n(U)$ in the former and the latter case. We injected a bit-flip randomly somewhere between the first and T^{th} cycle. As explained in Section IV-B, the SAT instances are created for these smaller divisions and provided to the solver. It is obvious that these instances (formulas) corresponding to some divisions ($eTrace_j$) would be satisfiable. In the division which contain the flip-cycle, the SAT instance would be surely unsatisfiable from which UNSAT core is to be extracted. We consider here TBw as 32 with one trace signal selected from each of 32 groups. In addition to the average (computed over 10 iterations, separately for error traces of length 500, denoted by S and 1000 cycles, denoted by U) number of suspects, we report the average memory usage (in MB) and time spent in UNSAT core extraction (in seconds) in Table VI for the division ($eTrace_j$) which is found to be unsatisfiable. As

TABLE VI
SUSPECTS DUE TO BIT-FLIP ERRORS FOR LARGE $ETrace$

$ckt.$	$n(S)$	$Mem(S)$	$time(S)$	$n(U)$	$Mem(U)$	$time(U)$
$c1$	2.1	158.9	1.2	3.4	307.8	2.4
$c3$	1.8	238.3	1.6	1.9	452.8	2.9
$c4$	2.3	275.2	1.6	2.7	532.1	2.7
$c5$	4.5	632.9	3.4	4.3	1202.5	5.7
$c6$	5.7	589.2	3.8	5.3	1120.1	5.6

expected with the increase in the length of error trace (from 100 cycles to 200 cycles in case of U), there is a considerable increase in the runtime and memory usage. To put this scheme in perspective, a qualitative comparison with major SAT-based post-silicon localization methods is outlined in Table VII.

TABLE VII
COMPARISON WITH OTHER SAT-BASED LOCALIZATION METHODS

Method	Type of Error	Spatial localization	Temporal localization	Trace width
[8]	bit-flip	within 20 to 40 FFs[2]	5 to 10 cycles of flip-cycle[2]	128, 256
[10]	RTL errors	max. 3% of modules	10-15% of total trace	groups of 16 registers
[3]	stuck-at[3]	actual net/wire	-	5% of all signals
Proposed	bit-flip, stuck-at	mostly <10 flip-flops	exact cycle[4]	32, 64

[2]Computed approximately by averaging the reported results of s35932 & s38417 circuits for a large number of experimental runs. In these iterations, exact flip-cycle can be localized, alongwith other probable candidates.

[3]It is claimed in [3] that arbitrary fault models are also possible.

[4]Other suspect flip-cycles also appear in some experiments.

VI. CONCLUSION

This paper proposed a signal tracing methodology for satisfiability based error localization for two kind of error scenarios in post-silicon environment. A grouping based methodology is presented for signal selection and with different observability enhancement schemes, chip execution history is obtained. The grouping methodology is intended to minimize the trace buffer width. The error trace obtained from chip execution is forced on time-frame unrolled circuit for the purpose of SAT solving as constraints. For larger error traces ranging in the range of thousands of clock cycles, SAT-based debugging can run into memory and solving time issues. To alleviate this, the error trace was partitioned into smaller traces. The proposed signal tracing methodology can be extended to address scalability issues such as applying this methodology to larger circuits. We are working on the applicability of the proposed methodology on different benchmark circuits from opencore and ITC'99 suites. Additionally, the proposed methodology needs further investigation to obtain the optimal value of $MinSize$ parameter. This has implications on the trace signals selected and the subsequent error localization.

REFERENCES

[1] K. Rahmani, S. Proch, and P. Mishra, "Efficient selection of trace and scan signals for post-silicon debug," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 1, pp. 313–323, Jan 2016.

[2] J. S. Yang and N. A. Touba, "Enhancing silicon debug via periodic monitoring," in *2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, Oct 2008, pp. 125–133.

[3] C. S. Zhu, G. Weissenbacher, and S. Malik, "Post-silicon fault localisation using maximum satisfiability and backbones," in *International Conference on Formal Methods in Computer-Aided Design, FMCAD '11, Austin, USA, October 30 - November 02, 2011*, 2011, pp. 63–66.

[4] X. Liu and Q. Xu, "On multiplexed signal tracing for post-silicon debug," in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.

[5] S. Prabhakar and M. S. Hsiao, "Multiplexed trace signal selection using non-trivial implication-based correlation," in *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, 2010, pp. 697–704.

[6] S. BeigMohammadi and B. Alizadeh, "Combinational trace signal selection with improved state restoration for post-silicon debug," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 1369–1374.

[7] M. Li and A. Davoodi, "A hybrid approach for fast and accurate trace signal selection for post-silicon debug," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, March 2013, pp. 485–490.

[8] A. Vali and N. Nicolici, "Bit-flip detection-driven selection of trace signals," in *21th IEEE European Test Symposium (ETS)*, May 2016, pp. 1–6.

[9] A. Suelflow, G. Fey, R. Bloem, and R. Drechsler, "Using unsatisfiable cores to debug multiple design errors," in *Proceedings of the 18th ACM Great Lakes Symposium on VLSI*, ser. GLSVLSI '08, 2008, pp. 77–82.

[10] Y. S. Yang, A. Veneris, N. Nicolici, and M. Fujita, "Automated data analysis techniques for a modern silicon debug environment," in *17th Asia and South Pacific Design Automation Conference*, Jan 2012, pp. 298–303.

[11] C. S. Zhu, G. Weissenbacher, and S. Malik, "Silicon fault diagnosis using sequence interpolation with backbones," in *The IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2014, San Jose, CA, USA, November 3-6, 2014*, 2014, pp. 348–355.

[12] A. Biere, "Picosat essentials," *Journal on Satisfiability, Boolean Modeling and Computation (JSAT*, p. 2008.

[13] http://www.opencores.org/.

[14] B. Kumar, K. Basu, A. Jindal, M. Fujita, and V. Singh, "Improving post-silicon error detection with topological selection of trace signals," in *2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Oct 2017, pp. 1–6.