Improving the Testability and Reliability of Sequential Circuits with Invariant Logic

Nuno Alves*, Kundan Nepal†, Jennifer Dwork*, Iris Bahar*
*Brown University, Providence, RI †Bucknell University, Lewisburg, PA

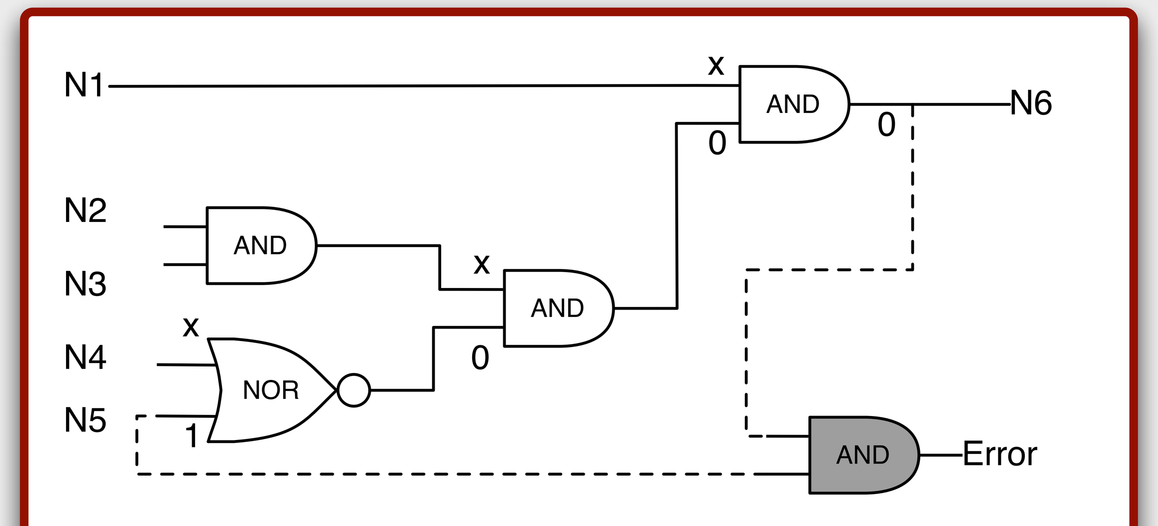
Abstract

- In previous work we showed that logic implications are effective in providing online error detection capabilities to a circuit.
- As an added value, logic implications have also been shown to improve the testing efficiency of an integrated circuit.
- We describe an algorithm that selects an implication set optimized for both online error detection and for test.

Background

Logic Implications

A logic implication describes the invariant relationship between two nodes within a circuit.



Logic Implication: N5=1 ⇒ N6=0

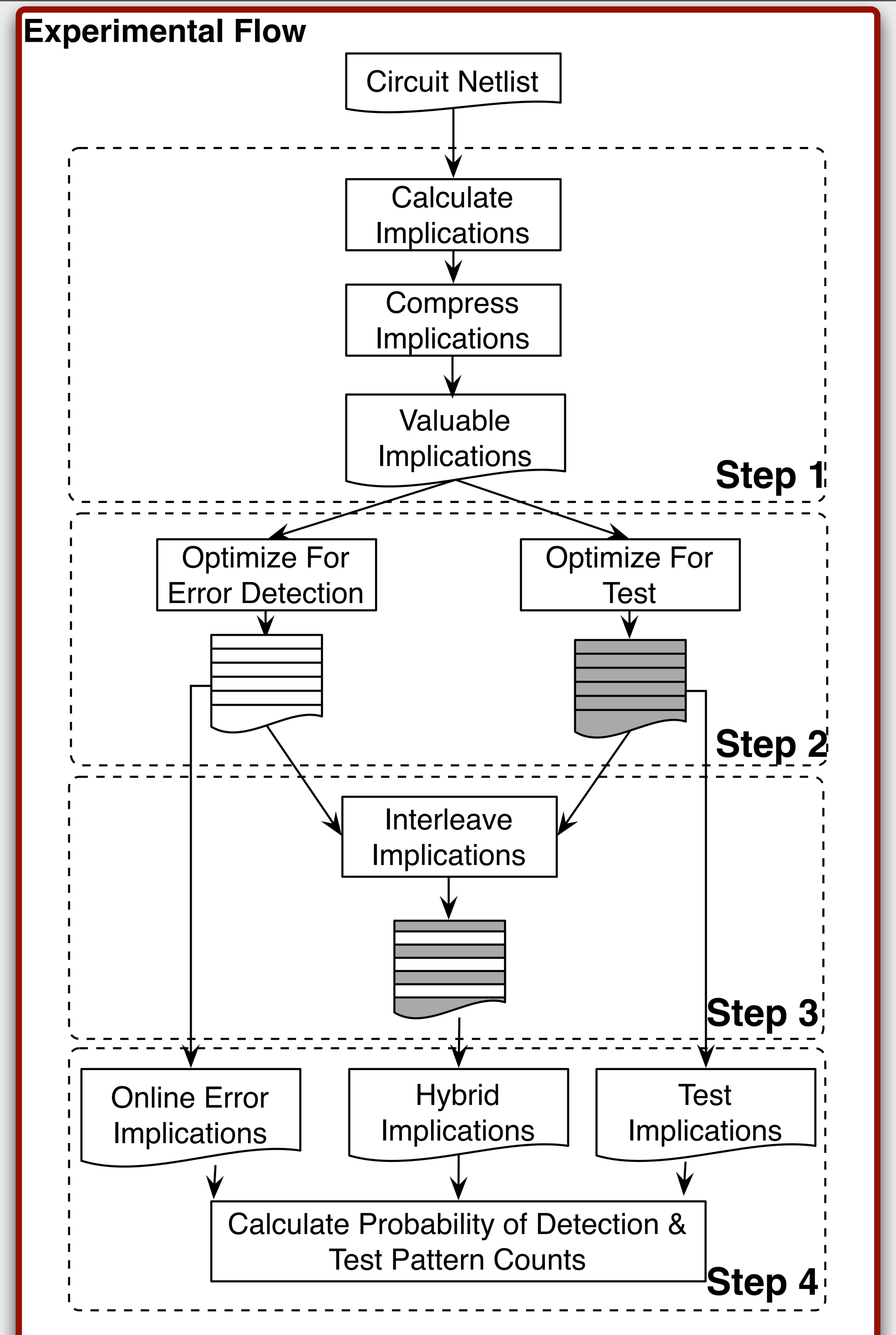
Logic Implications for Online Error Detection

- We have shown in our previous work [1] that logic implications have strong online error detection capabilities.
- In a circuit, due to logical constraints, there are several implications that exist between various nodes in the circuit .
- If this relationship does not hold, an error must have occurred in the intervening logic between the two sites or at the second site. The logic shown in gray can be easily added to the circuit to check for violation of this implication.

Logic Implications for Test

- In previous work [2], we have shown that the logic implication checking hardware also is helpful to mitigate the problem of difficult-to-detect defects.
- In addition to runtime detection of faults, each implication we insert can essentially serve as a new test point, which will increase the observability of the logic covered by that implication.

Methodology



Step 1 - Implication Discovery

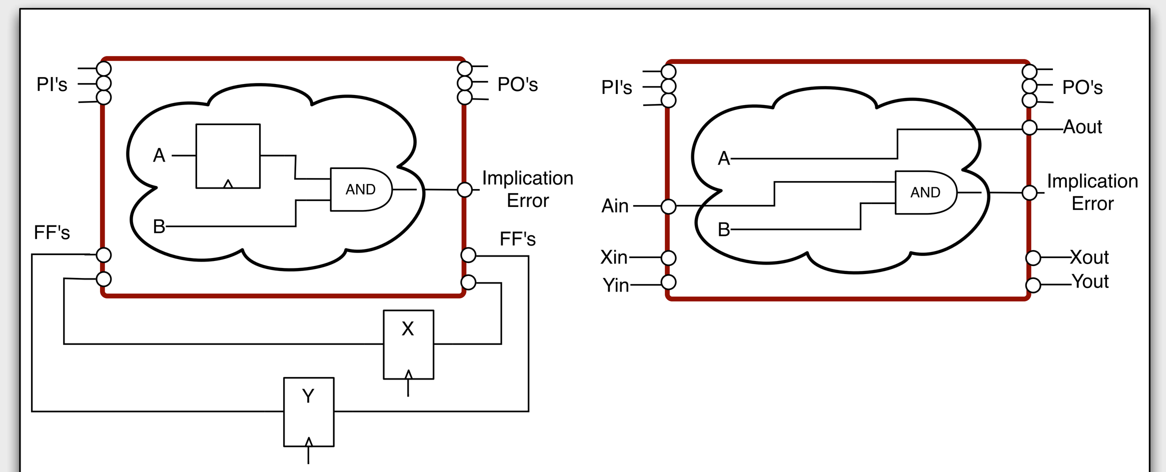
- Implications are found by performing pairwise comparisons between circuit sites [3].
- Even a relatively small circuit will yield thousands of implications. Not all of them are equally effective.
- Ineffective implications are filtered out, creating a compressed list of "valuable" implications.

Step 2.1 - Determine Implications For Error Detection

- The procedure begins with the insertion of all valuable implications into the circuit and subjecting them to fault simulation.
- The result of this fault simulation will tell us what implications are able to detect which faults, and how many times each fault is detected.
- Once all faults are processed, we sort the implications in descending order according to number of fault detections with which they are associated.

Step 2.2 - Determine Implications For Test Pattern Reduction

- Implications can ease the observation requirements for certain hard-to-detect faults.
- If previously hard-to-detect faults become easier to detect, it implies that fewer test vectors may be required to fully test the circuit.
- The flip-flops in cross-cycle implications may be placed on the scan chain, resulting in additional observation points. Therefore even fewer vectors should be required to fully test the circuit.



(a) normal mode (b) scan mode

- Starting with the set of valuable implications, we proceed to add one implication at a time to the circuit under test, and calculate the required number of 15-detect vectors using commercial ATPG tools.
- When the 15-detect vector count has been computed for each implication, we add the implication with the largest vector count reduction to the circuit, and repeat the process.

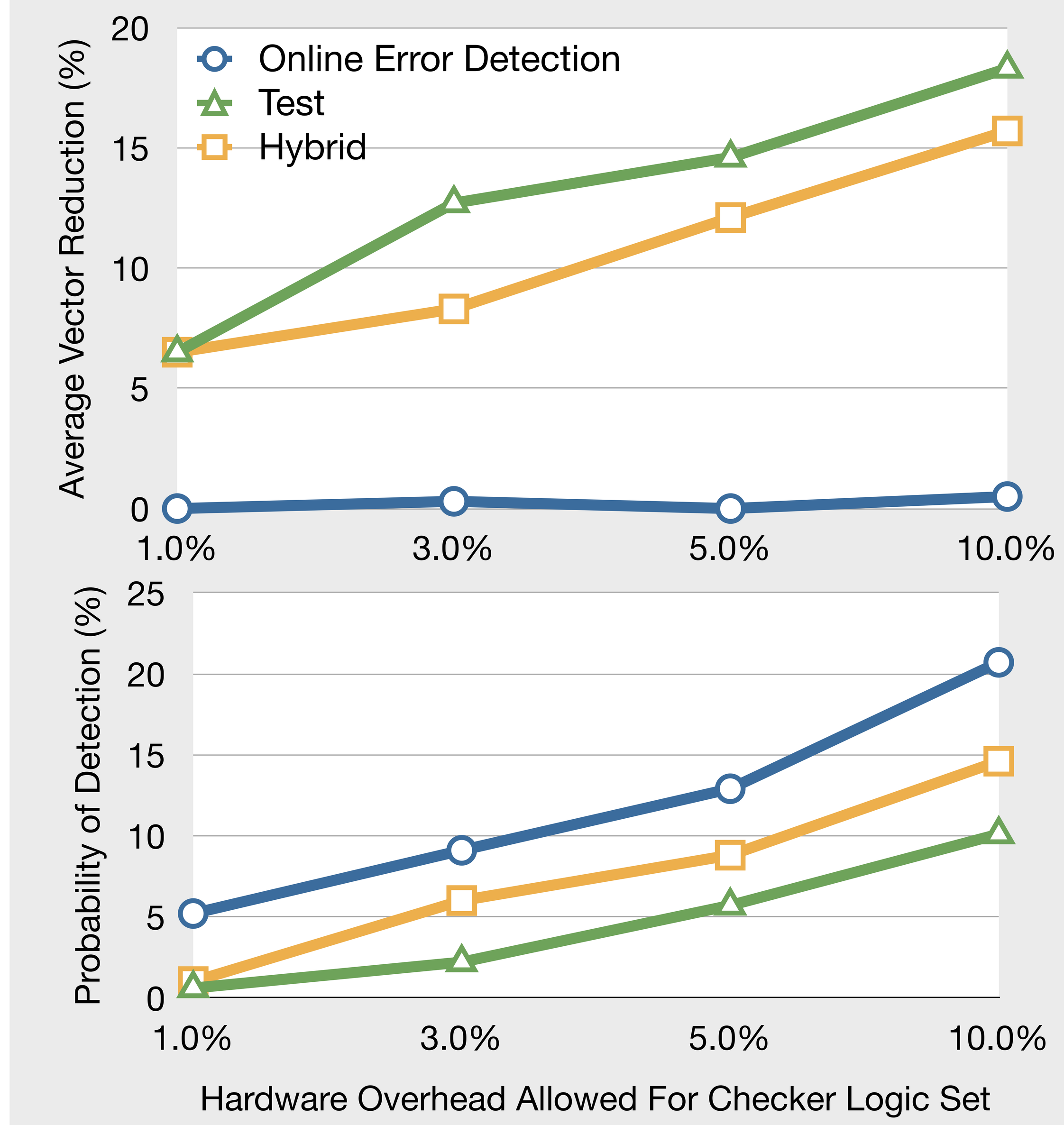
Step 3 - Determine Hybrid Implication Set

- Our goal is to find a set of implications that is equally useful for both online error detection and test.
- The hybrid implication set is the interleave of the sets of implications optimized for test and implications optimized for online error detection.

Step 4 - Calculating Performance For each Implication Set

- To evaluate the performance of an implication set we utilize two metrics: probability of detection and 15-detect vector count.
- The probability of detection is calculated by initially performing stuck-at-fault simulation, with 32k random input vectors, and then counting how often an internal fault that is observable at one of the circuit's outputs is detected and flagged by the implication checker hardware.
- The 15-detect vector count metric is accomplished by calculating the number of ATPG vectors required to detect each fault in the circuit at least 15 different times.

Results



Probability of Detection

- Implications optimized for test perform poorly compared to the other optimized implication sets in terms of error detection. However, they still had some noteworthy online error detection.
- An implication set optimized for online error detection, provides significant error coverage with a very low area overhead.
- The hybrid set lies roughly between the other two sets of optimized implications for all hardware overheads.

Test Pattern Count

- The set of implications optimized for test perform much better than those optimized for online error detection in terms of test vector reduction.
- The hybrid implication set, which previously demonstrated promising online error detection capabilities, now shows consistent vector count reduction as well.

References

(1) K. Nepal, N. Alves, R. Iris Bahar, J. Dworak, *Using Implications for Online Error Detection*, ITC, October 28-30, 2008
(2) N. Alves, K. Nepal, R. Iris Bahar, J. Dworak, *Compacting Test Vector Sets via Strategic Use of Implications*, ICCAD, November 2009
(3) N. Alves, A. Buben, K. Nepal, R. Iris Bahar, J. Dworak, *A Cost Effective Approach For Online Error Detection Using Invariant Relationships*, IEEE TCAD, Vol. 29, No. 5, May 2010