Detecting Errors Online Using Invariance Information
Motivation

- Errors in ICs are increasing
  - Particle strikes, temperature, power, noise, process variations, test escapes, etc.
- Can we use *logic implications* for online error detection during a single clock cycle?
- *What happens if we consider implications across time cycles?*
Our Approach

Find natural expected relationships and check for their violation.

Water should be blue….

Not brown…

In circuits, expected relationships at the gate level consist of logic implications.
Implications Naturally Occur in Circuits

\[ n_5 = 1 \rightarrow n_8 = 0 \]
Implication Violations Can Be Used to Detect Errors

Appropriate checker logic can detect multiple errors with a single implication.

\[ n5 = 1 \quad \rightarrow \quad n8 = 0 \]
Implication Violations Can Be Used to Detect Errors

Appropriate checker logic can detect multiple errors with a single implication.

\[ n5 = 1 \quad \text{and} \quad n8 = 0 \]
Multi-Cycle Implications

Sequential Circuit Containing No Non-Trivial Implications in Combinational Logic

Time Frame Expansion

Logic Value in First Clock Cycle Implies a Value at a Different Site in the Second Clock Cycle

$B_1 = 0 \rightarrow F_2 = 0$
Multi-Cycle Checker Hardware

Checker hardware requires state to be held between first and second cycle....
Contribution of Different Implication Classes to Error Detection

- 1st cycle
- 1st and 2nd cycle
- Cross cycle
- All
The grant:

- Enabled us to discuss our work with a leading researcher in the electronic testing field through several face-to-face meetings.
- Gave us the opportunity to make several presentations (nationally and internationally) on our work during its early stages.
- Provided opportunities to receive useful feedback to help improve our approach.
- Helped provide international recognition for our project.
What’s Next?

- We just received a $500K NSF award to further this research. 😊
- Implications also have potential for use in manufacture testing.
  - Reduce size of test set by adding implication checkers.
- Incorporate fault criticality into implication selection
  - Only worry about faults that have an especially detrimental impact on user applications.