Soft-Error-Analysis

Patrick Klampfl
patrick.klampfl@student.tugraz.at

Computer Science

March 1, 2016

Overview

- Previously … [recap]
  - Introduction: Soft-Errors & Soft-Error Analysis
  - Detect Soft-Errors
  - Verify Protection Logic (vulnerabilities)

Internship Summer '15

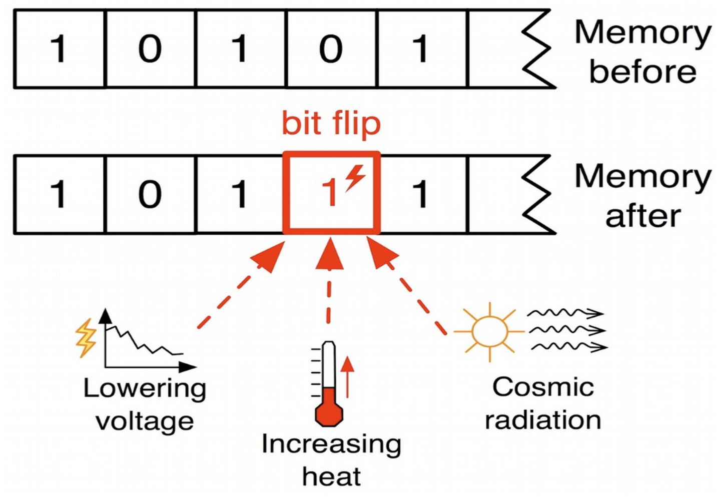
- Latest work:
  - Verify Protection Logic (false positives)
  - Environment Models
  - Benchmark Results

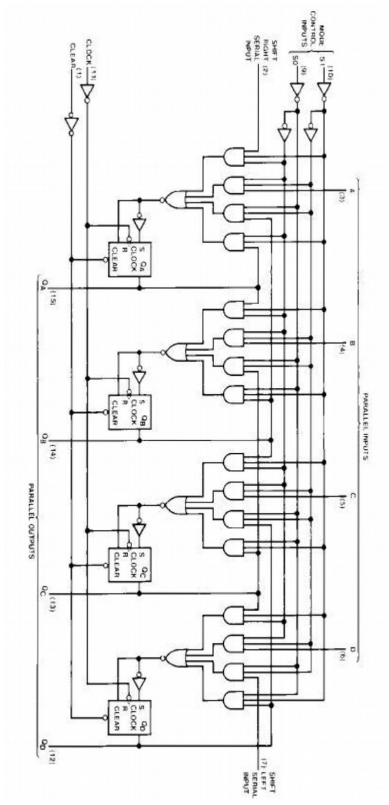
Master-Project

- Conclusion

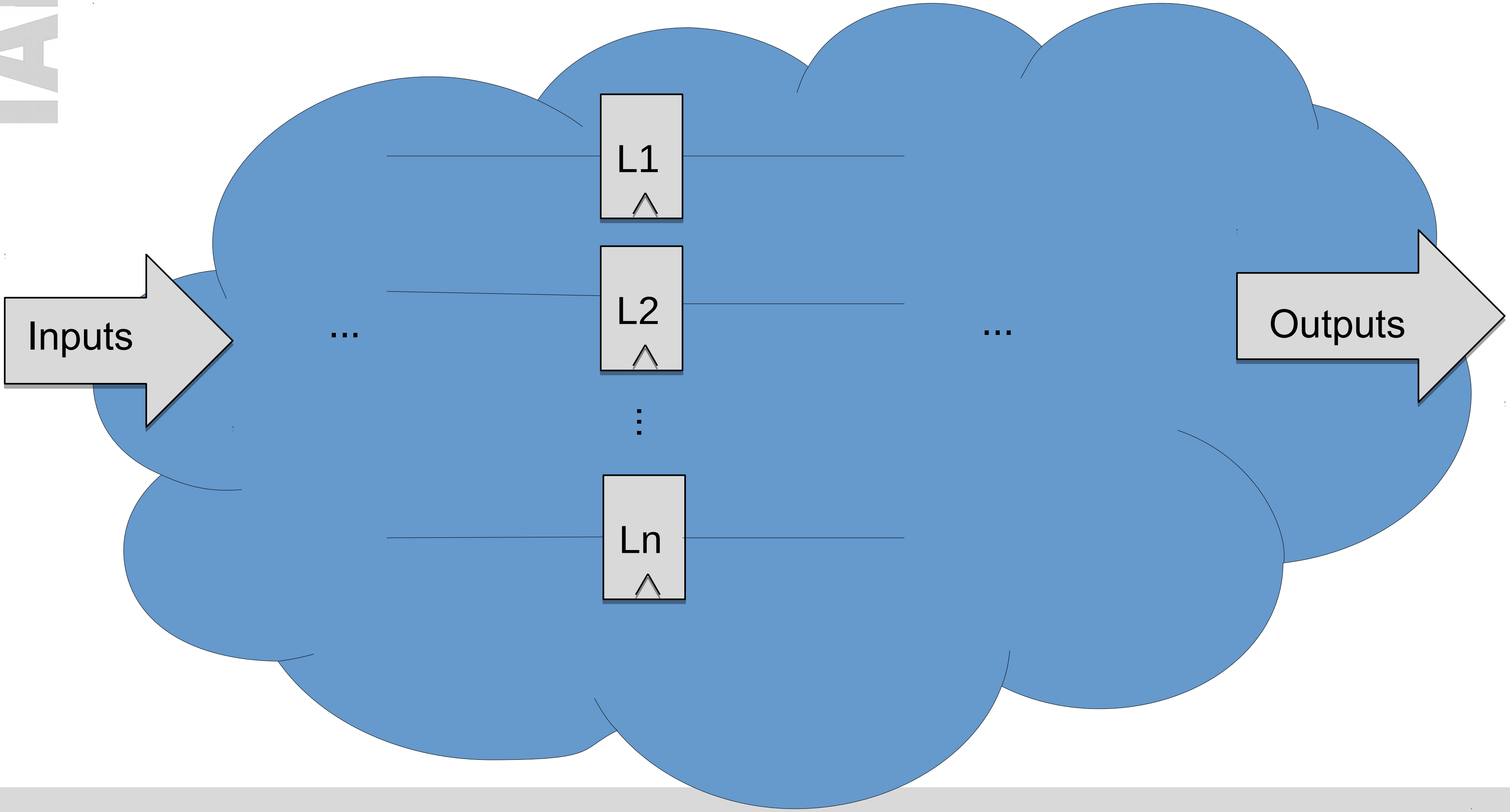
Soft Errors

- Boolean circuits: inputs, AND gates, latches, outputs
- Components (latches, AND gates) can have soft-errors
  - flip truth value



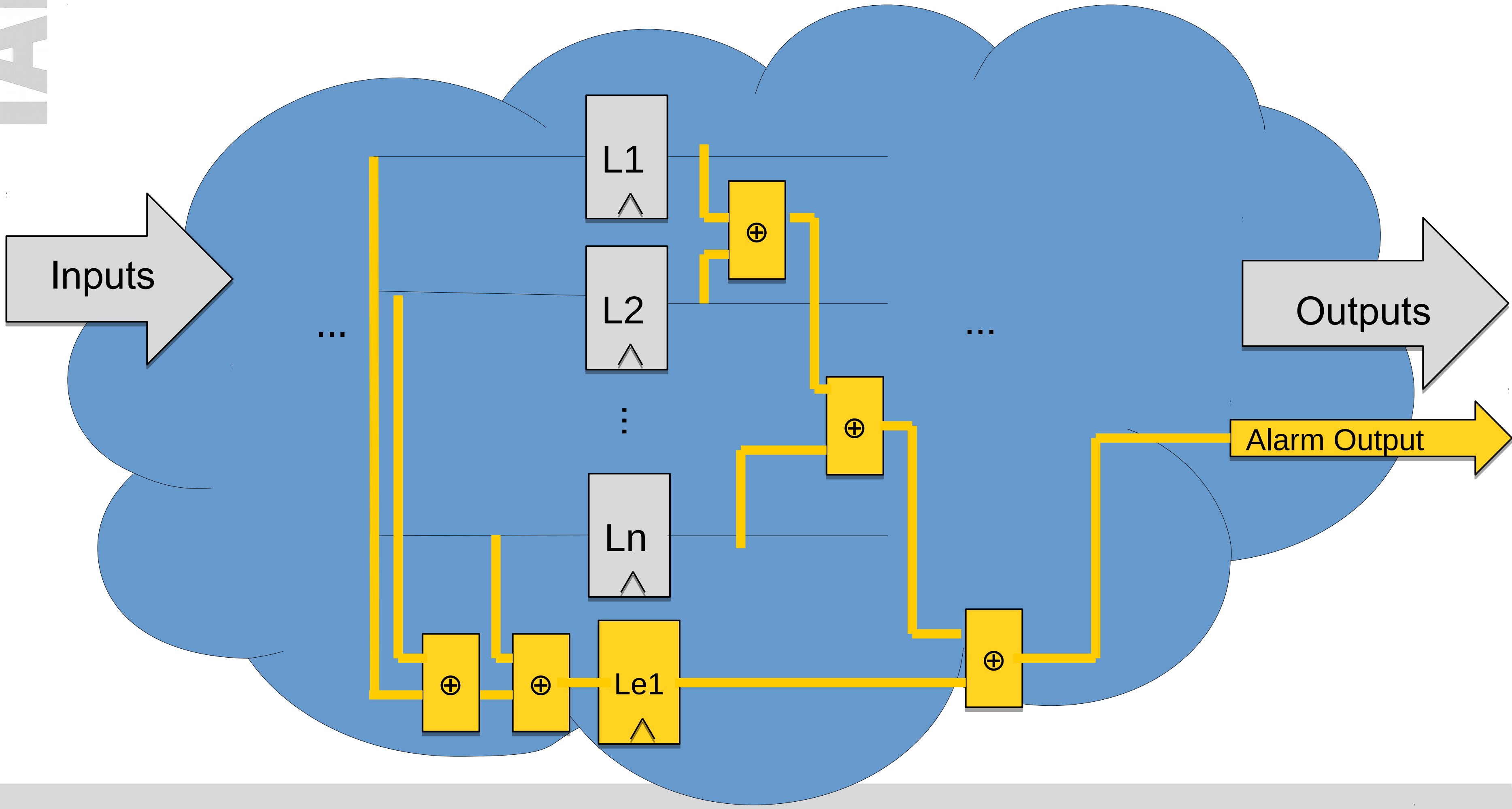


How to detect Soft-Errors?



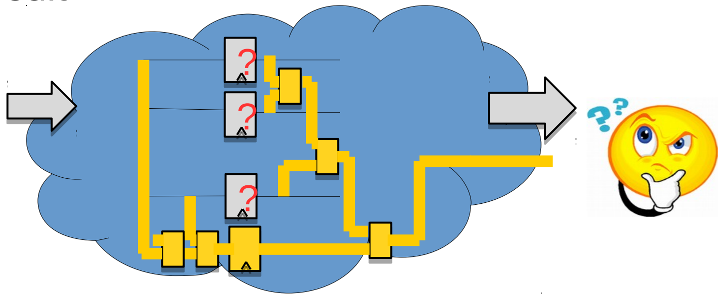
How to detect Soft-Errors:

- → add **redundancy.** Tool: **AddParityTool**

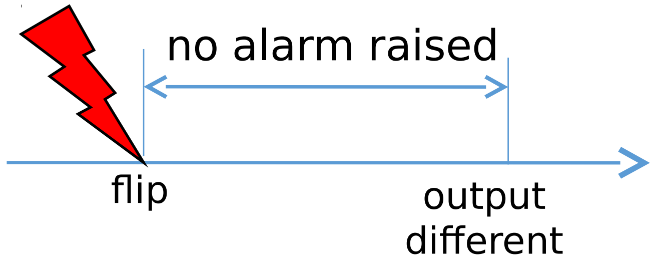


Q: Is the protection-circuit correct ... ?

- Given: Circuit

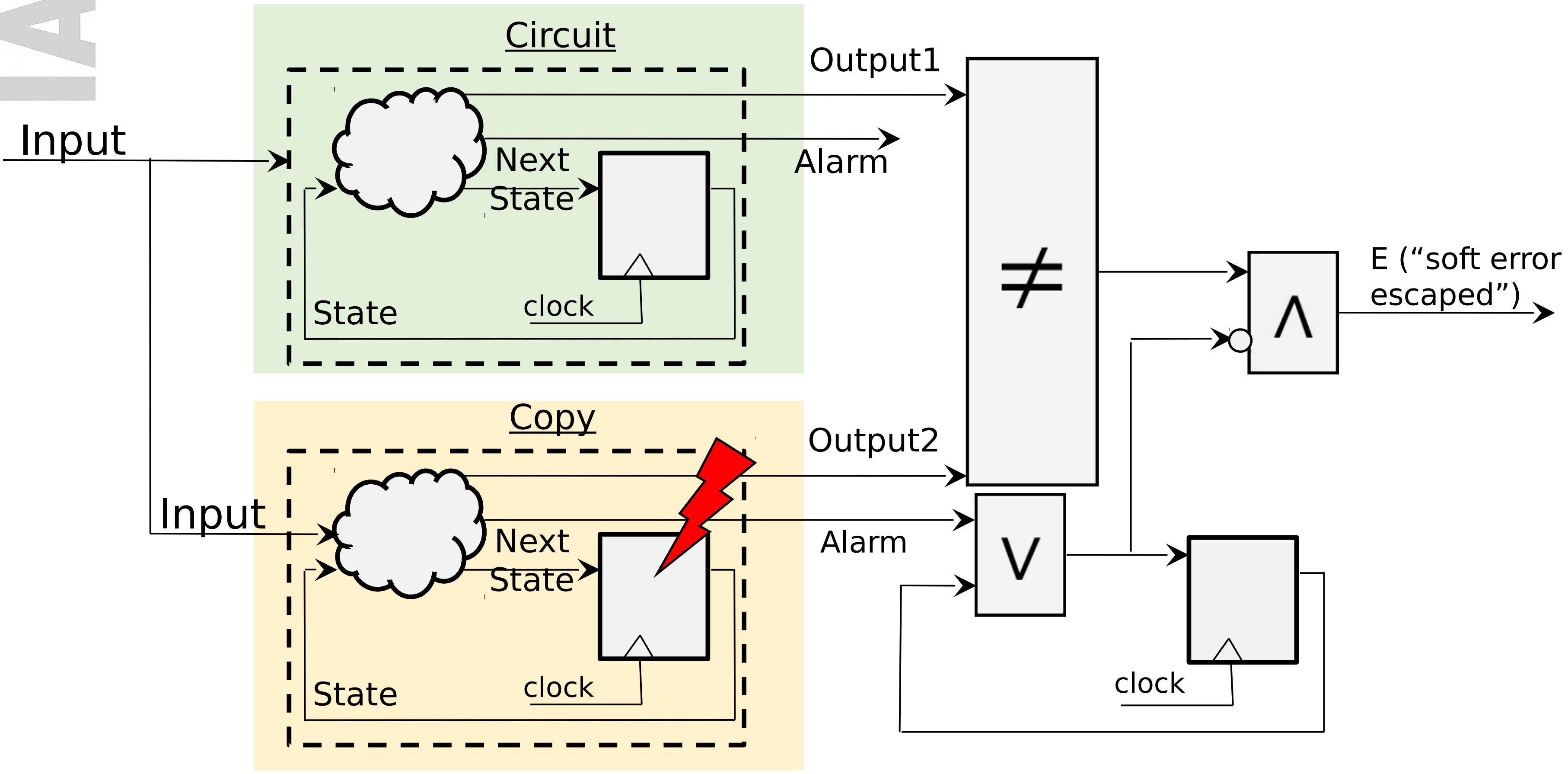


- Find Vulnerabilities:
  - Latches that can be flipped (once)
  - such that output changes (at some point in the future)
  - but no alarm raised (up to that point)

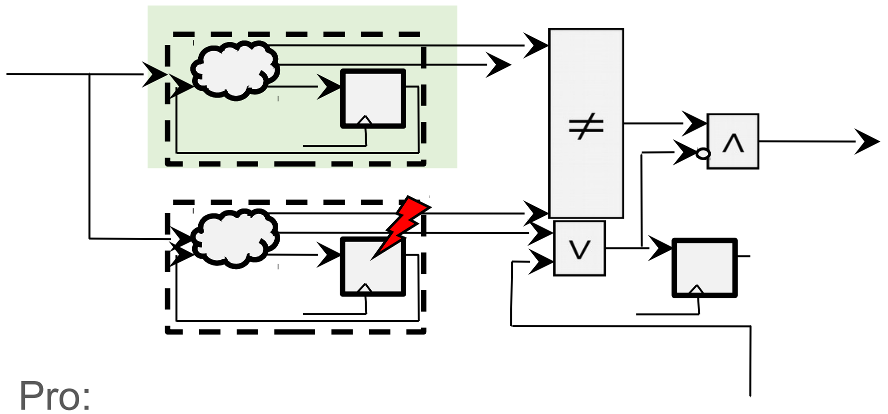


Model Checking Approach

- Tool: AlarmToMC

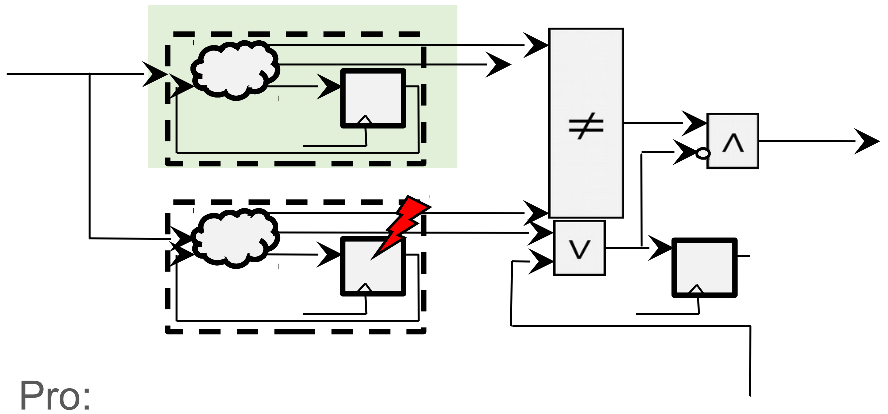


Model Checking Approach



- Pro:
  - Exact: Valid for all possible input combinations
- Contra:
  - Bad scalability

Model Checking Approach

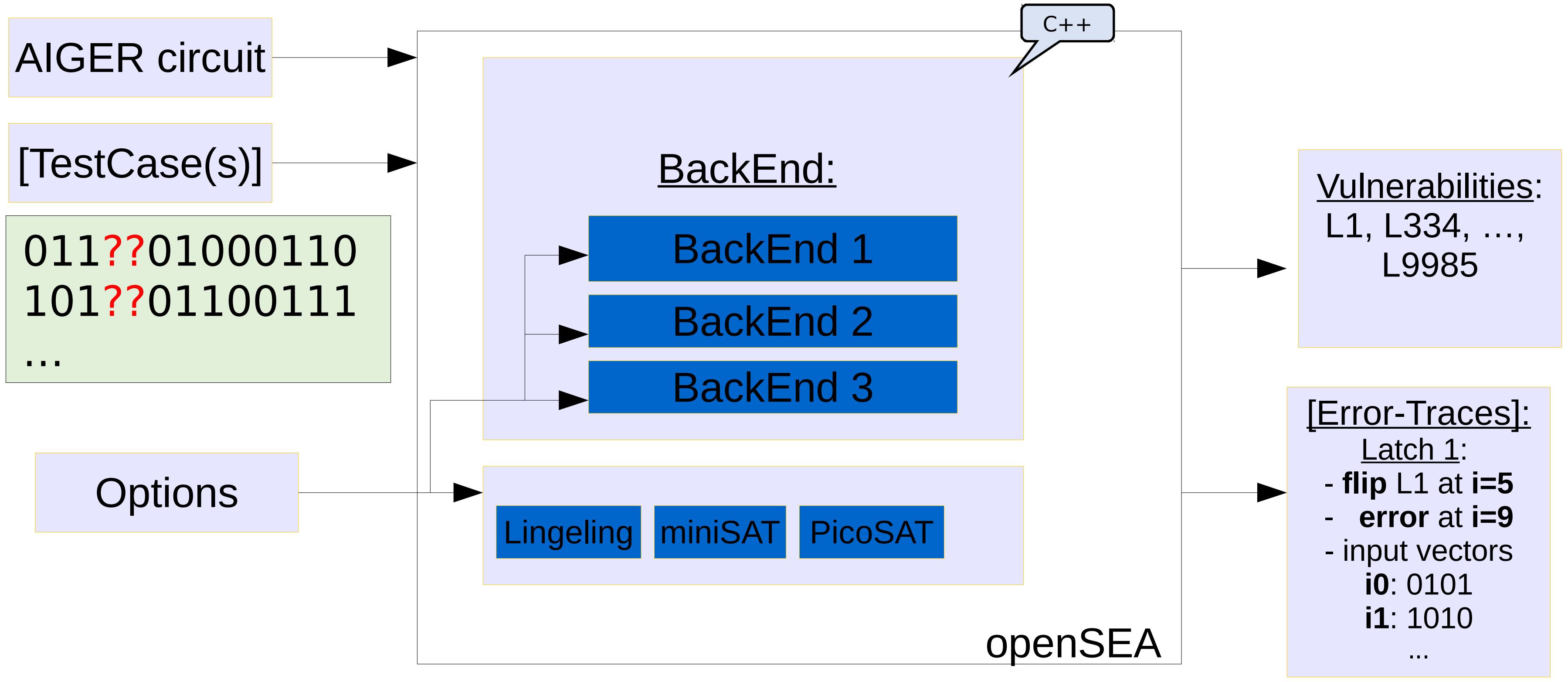


- Pro:
  - Exact: Valid for all possible input combinations
- Contra:
  - **Bad scalability**

Idea: Instead of all possible Input combinations, use concrete input vectors

openSEA

- Input: arbitrary circuit with protection logic (alarm output)
- Output: List of definitely vulnerable latches



BackEnds

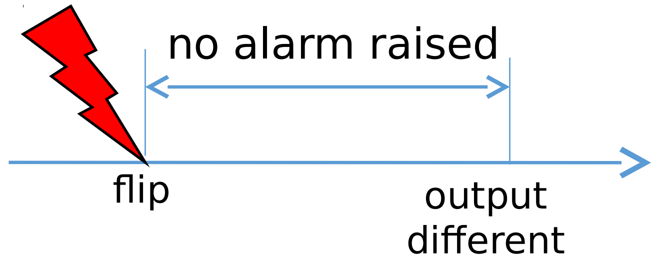
- Simulation – based: (SIM)
  - Execute **correct simulation** with the provided TestCase
  - Compare with **all possible faulty simulations**
- **Symbolic Time Analysis: (STA)**
  - Point in time when to flip a latch is symbolic
- **Symbolic Time + Symbolic Location (STLA)**
  - Point in Time + Latch to flip is component

Latest Work

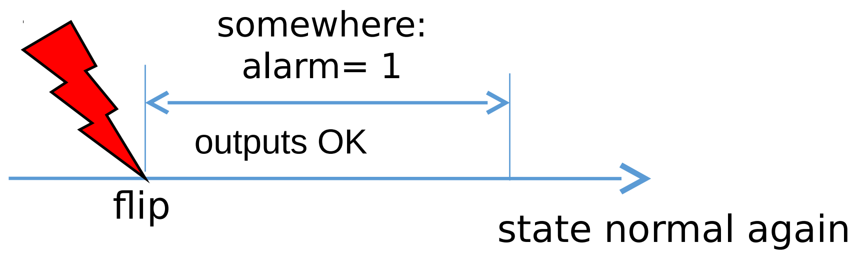
- False Positives
- Environment Models
- Benchmark Results

False Positives

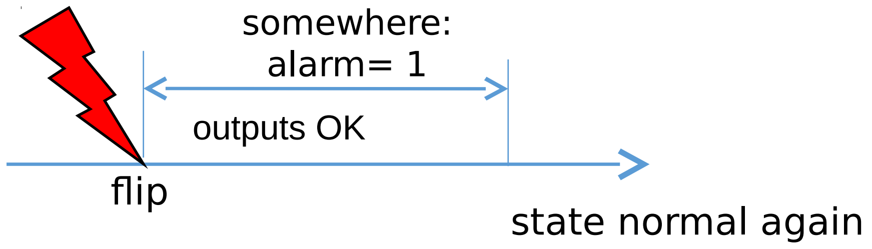
- Vulnerabilities are false negatives: a soft error happens, but is not detected
  - Alarm should have been raised



- False Positive: Alarm is true, but the soft-error has no effect
  - Alarm raised gratuitously



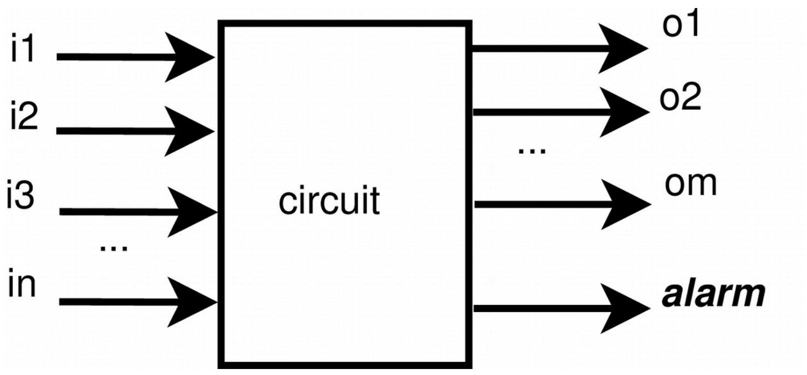
False Positives

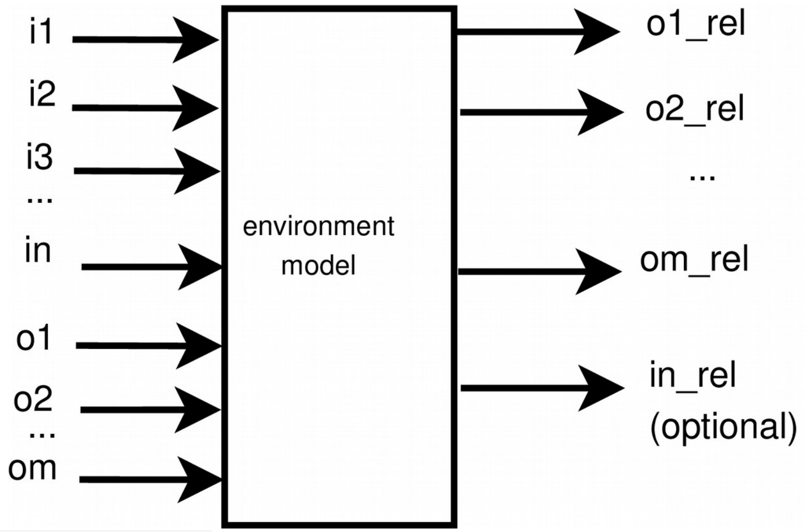


- Implemented similar to Algorithms for false-negatives:
  - Symbolic Time Analysis (STA)
  - Symbolic Time Symbolic Location Analysis (STLA)

Environment Models

- Output values might be irrelevant
  - e.g. if data on bus is not ready
- Some input combinations might not be allowed
  - SAT-solver choices for input values can be restricted

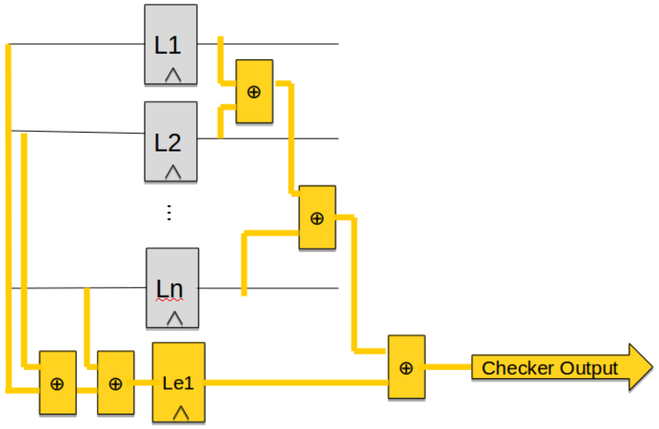




Benchmark Results – Setup for Experiments

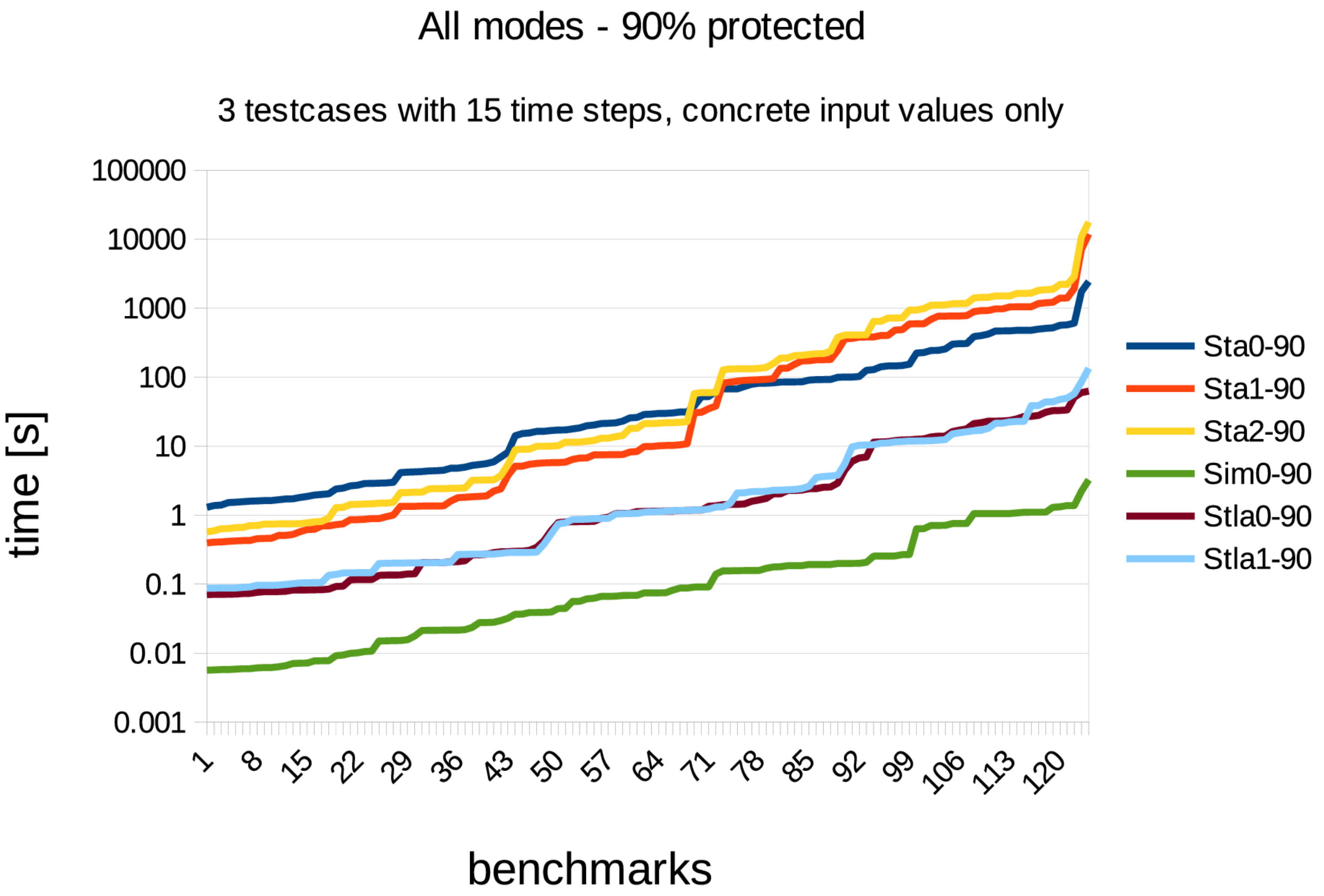
- IWLS 2002 and IWL 2005 [1] circuits converted to AIGER format
- Add protection (AddParityTool)
  - only parity
  - Parameters:
    - Percentage of latches to protect
    - Number of latches to protect with 1 new latch
- Test Inputs: created randomly

```
011??01000110
101??01100111
...
```

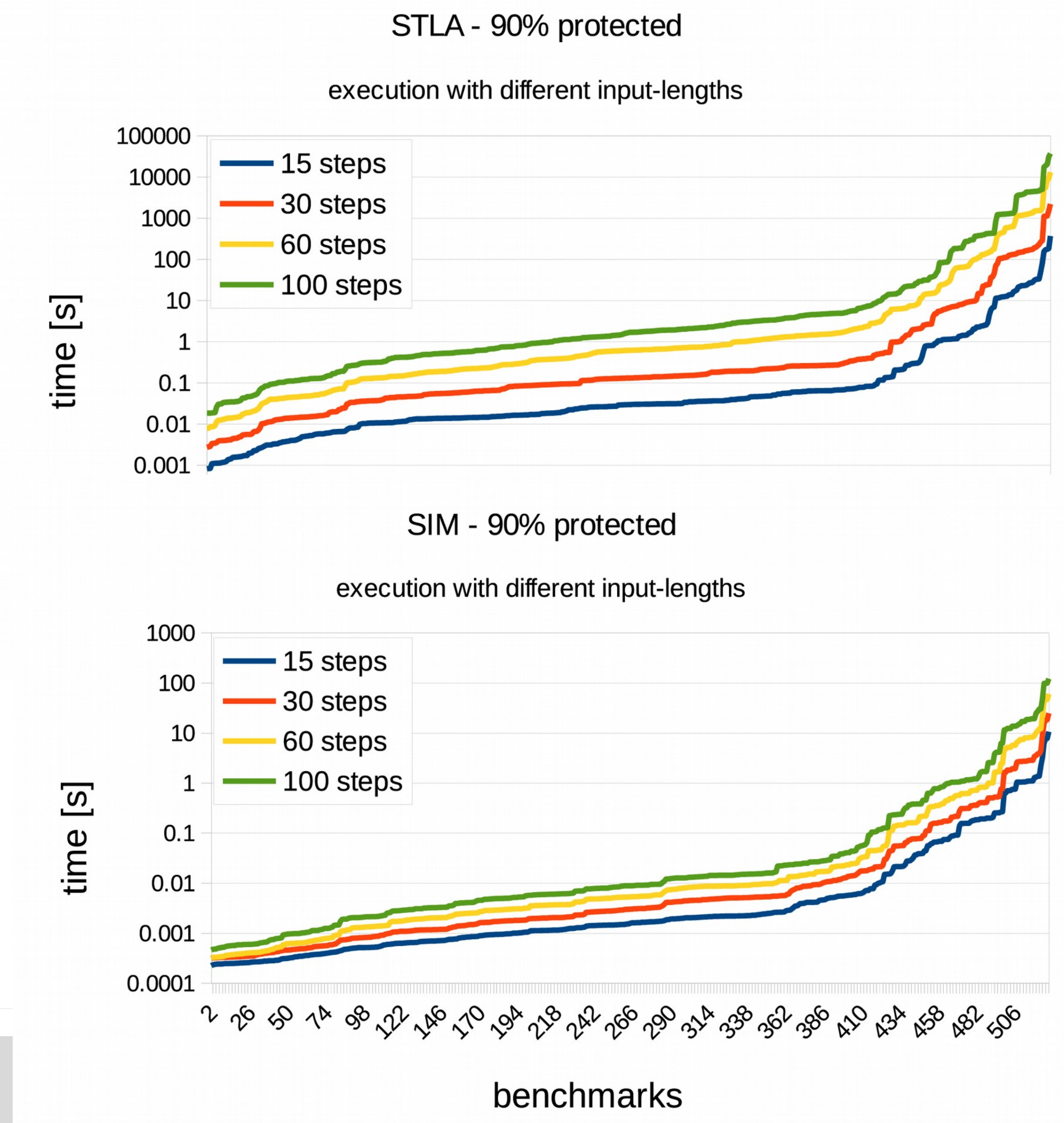


[1] http://www.eecs.berkeley.edu/~alanmi/benchmarks/

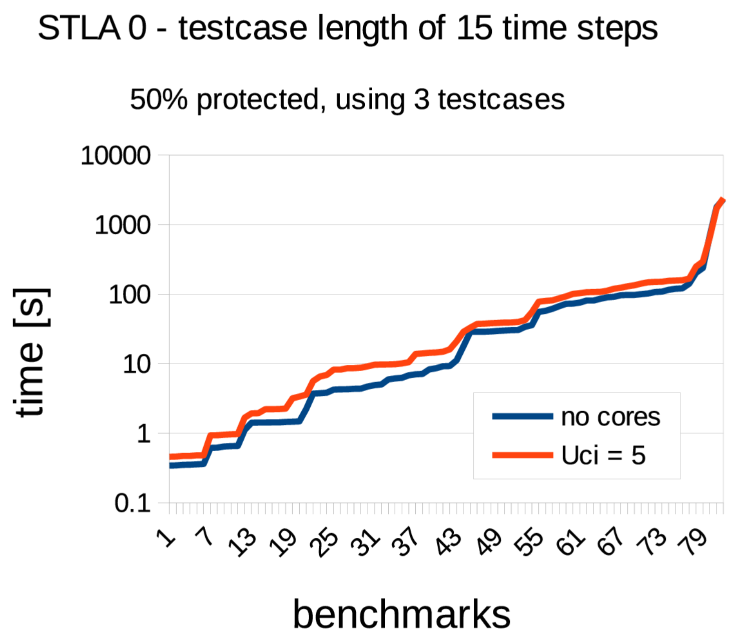
Results – All Algorithms

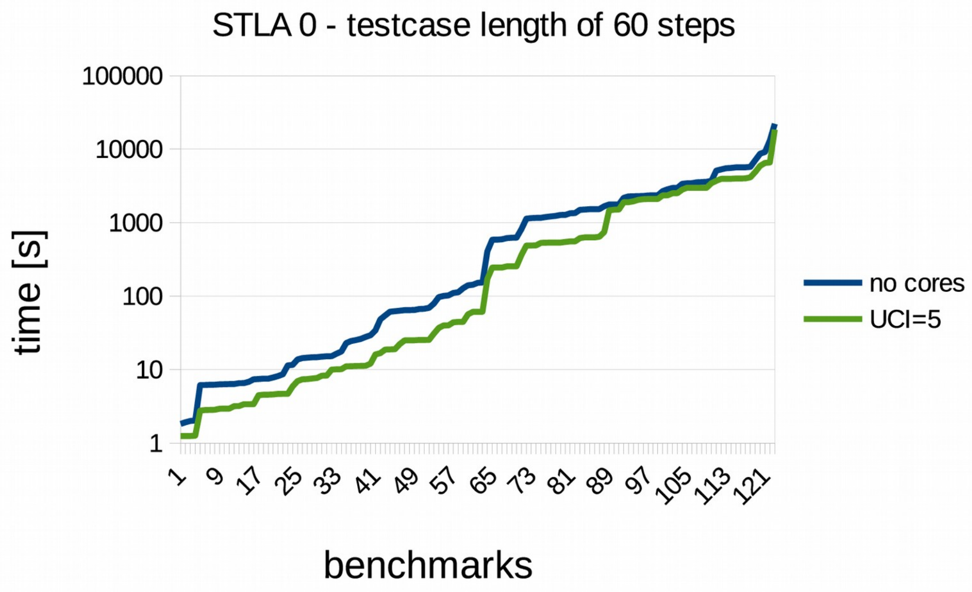


Results – Length of Test Cases

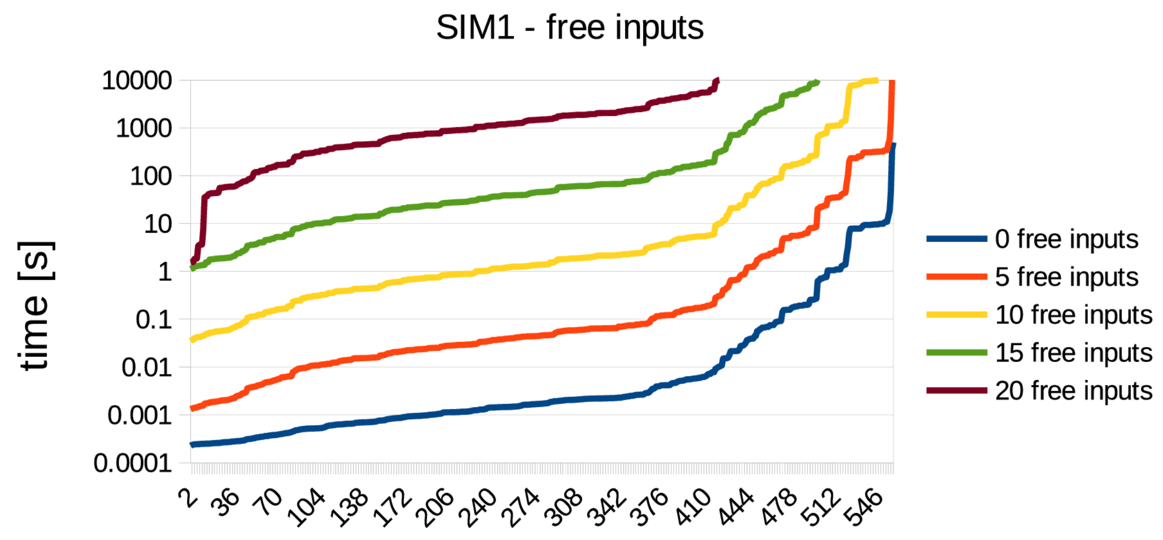


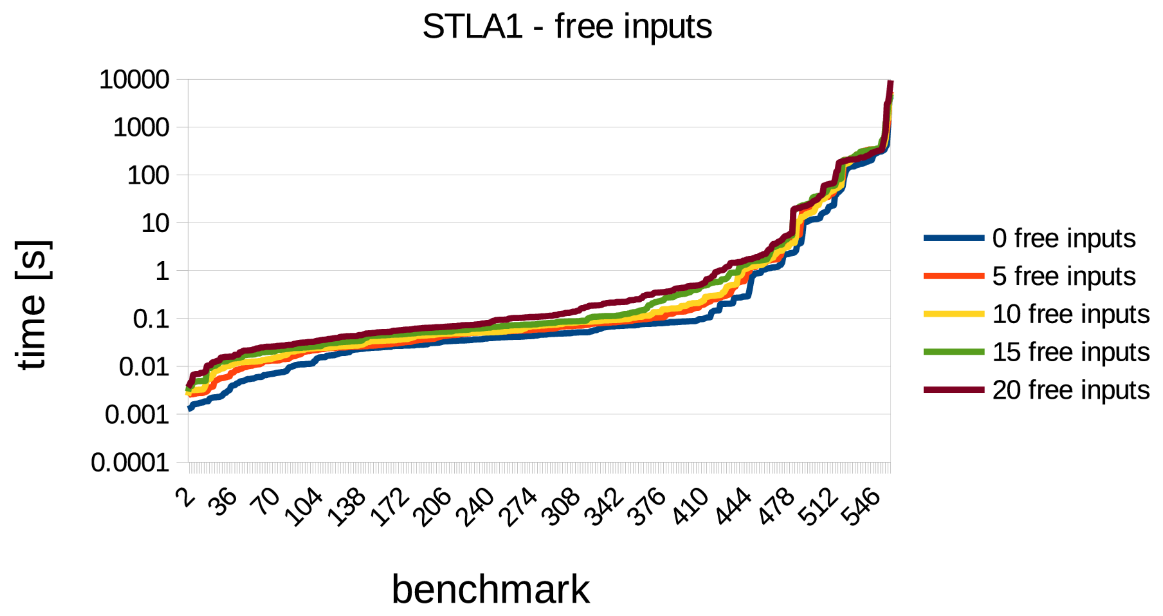
Optimization: Unsatisfiable Cores



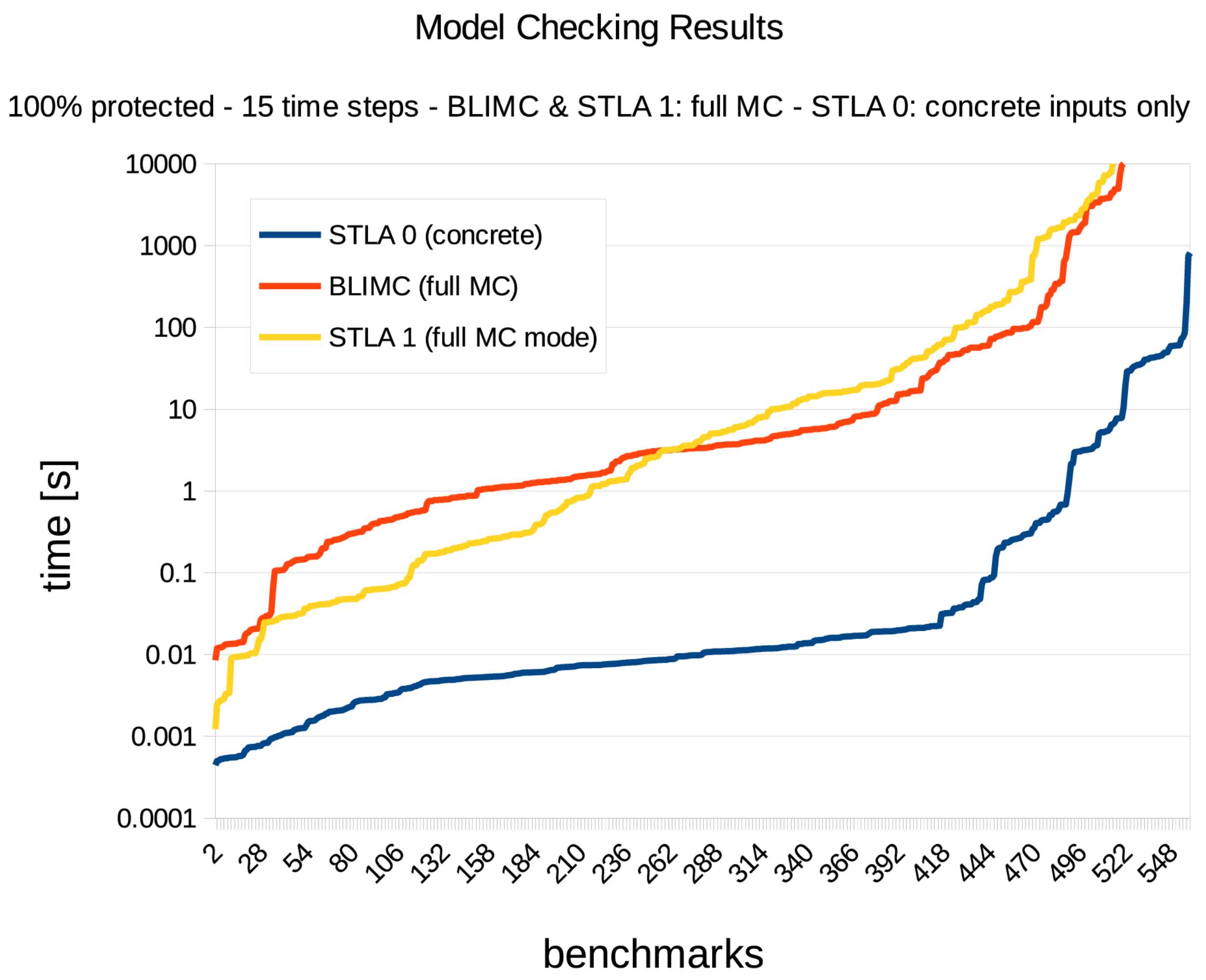


Number of unspecified input values



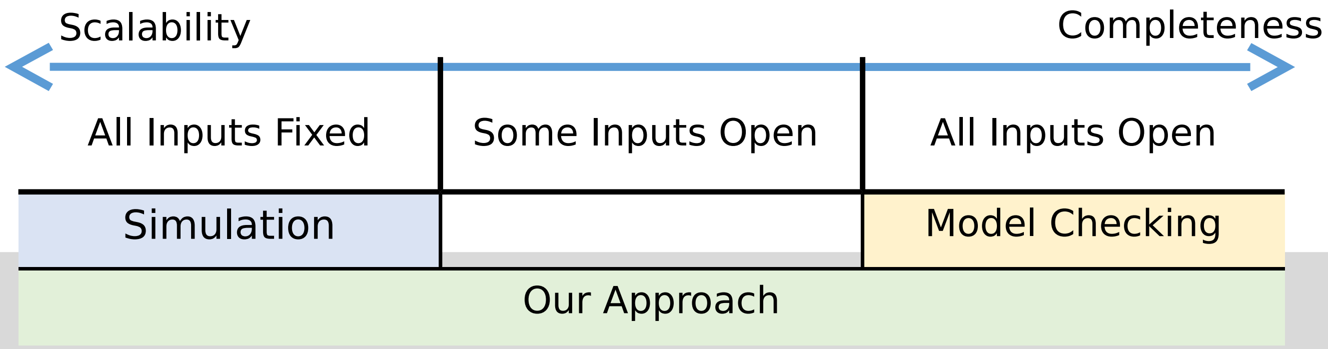


Model Checking Results



Conclusion

- Extended **openSEA**
  - False Positives Algorithms
    - Symbolic Time
    - Symbolic Time + symbolic Location
  - Environment Models
- Benchmarking results
  - Free inputs: sym. Algorithms (STLA) scale significantly better than simulation
  - Concrete Inputs: Simulation is fastest
  - Reducing input space: better than MC
  - UNSAT cores might speed up longer test cases


Scalability
Completeness
All Inputs Fixed
Some Inputs Open
All Inputs Open
Simulation
Model Checking
Our Approach