A Scalable and Robust Approach to STL Certification including Safe Fault Analysis

Chandra Has Dondapati, Mohammed Arif, Saya Goud Langadi, Vishal Bhadauria, Subham Mohapatra, Prasanth Viswanathan P, Sanjay Singh, Krishna Allam, Karthik Rajakumar

60 DESIGN AUTOMATION CONFERENCE
FROM CHIPS TO SYSTEMS — LEARN TODAY, CREATE TOMORROW

1. Motivation & Problem Statement (1/2)

- One work product that we deliver to the end customer is a Software test library (STL).
- A self test library(STL) is a software work product that is used to identify hardware faults in a processor core. The percentage of faults detectable is called the Diagnostic Coverage(DC)
- A STL is used to identify hardware faults in a processor core in the field. The percentage of faults detectable is called the Diagnostic Coverage(DC)

$$DC = \frac{DETECTABLE\ FAULTS}{TOTAL\ FAULTS\ -SAFE\ FAULTS}$$

Functional safety:
Refers to the correct *function* of a safety related product

Traditional product safety
Refers to electrical shock, fire, mechanical etc

2. Motivation & Problem Statement (2/2)

- We have observed a significant rework with re-using the STL for a spinoff design. The challenge is to improve the time taken to get certification for the STL for a spinoff SOC

Challenges faced

1. Safe faults need to be identified for every design
2. The pass/fail criteria for the STL changes for every spin

3. Proposed Solution (1/3)

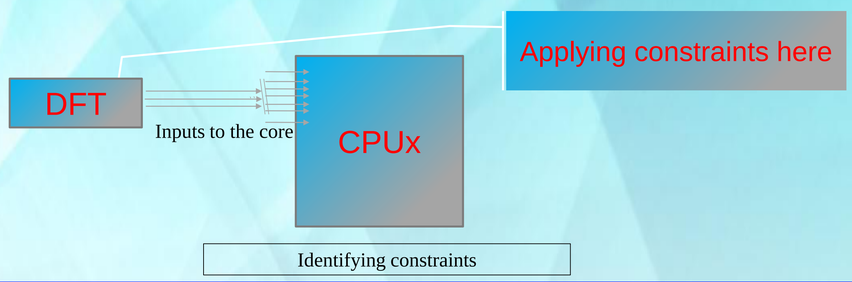
C1: Identifying safe faults	• Using a formal approach using JG -FSV • Constraints applied at source block (DFT)
C2: The pass/fail criteria for the STL is altered	• Event dependent CRC checks • Cycle dependent CRC checks • Lockstep comparator module

4. Proposed Solution (2/3)

The JasperGold Functional Safety Verification (FSV) tool outputs a comprehensive list of safe faults from the design.

Problem : Identifying the formal constraints in every SOC in a reliable and quick manner

Solution : Apply constraints at DFT boundary



5. Proposed Solution (3/3)

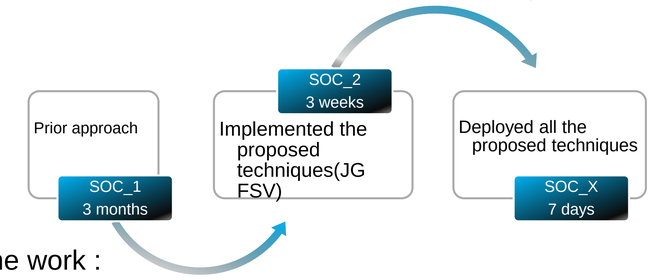
Problem : Volatile CRC signature between spinoff devices

Solution :

Use event-based CRC calculation instead of cycle-based CRC	No cycle wise matching - less conservative fault discovery
Have N golden signatures for N arbitration states	Need to add PSA(s) considering all arbitration cases
Add a INIT segment to bring the pointer to a known arbitration state	This works under the assumption that the STL run is uninterrupted.
Using Lockstep comparator	If lockstep is available on the device, it can be used for fault detection in place of a defined pass/fail criteria

6. Evidence and Results

- Significantly reduced the time taken for analysis – comparative data shown below



- Impact of the work :

Better STL quality :
- The robustness and the formal approach to identifying these safe faults ensures that the STL(customer deliverable software product) is error free and is standardized across different SOC families.

Easier Certification process :
- The standardized and scalable approach also helps us smoothen the process of obtaining certification for derivative designs without any significant rework.

Can be used for Hardware FI
- This approach can be used by any team with an end goal of delivering a STL to the customer or any team trying to demonstrate DC for the purposes of obtaining FuSa certification

7. Conclusions

The application of these techniques leads to :

Better STL quality → *Quicker Certification process* → Reusable by any team → Can be used for Hardware FI

Acknowledgements

- **Name** of Texas Instrument for …

TEXAS INSTRUMENTS