

Runtime Fault-Injection Tool for Executable SystemC Models

B.-A. Tabacaru,

M. Chaari, W. Ecker, T. Kruse

Infineon Technologies AG



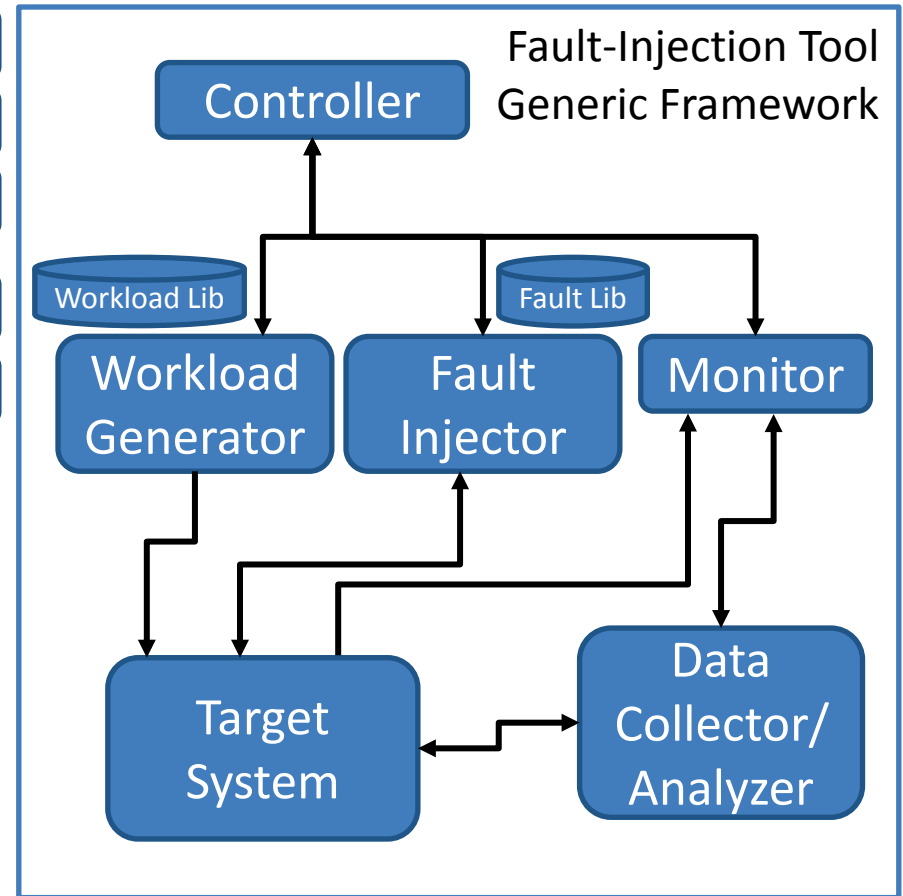
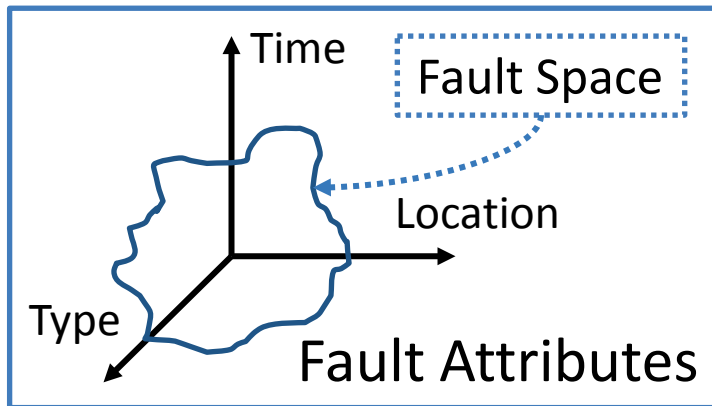
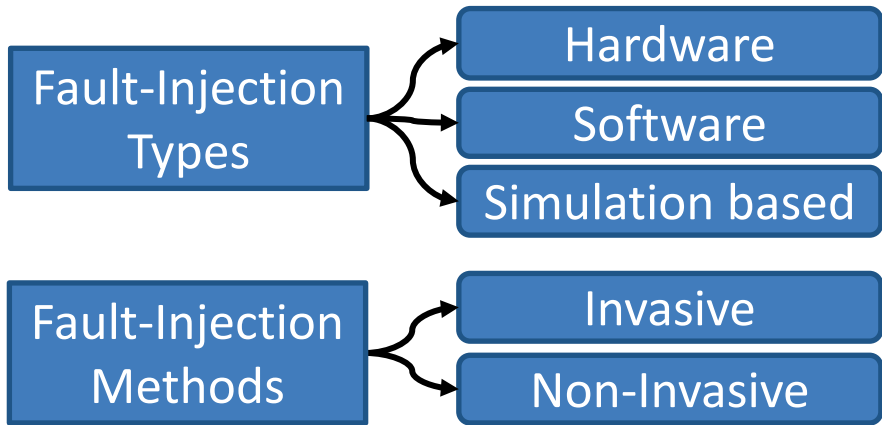
Outline

- Motivation
- Fault-Injection Theory
- Fault-Injection Requirements
- SCFIT (SystemC Fault-Injection Tool)
- Results
- Discussion
- Next Steps

Motivation

- Safety-critical applications
 - Dependability analysis
 - ISO-26262 based safety analysis
- System reaction to fault injection
 - Virtual prototyping for early fault analysis
 - Get results before hardware is ready
 - Faster simulation time
- Architecture exploration of fault-tolerant designs
 - Chip area
 - Design complexity/cost

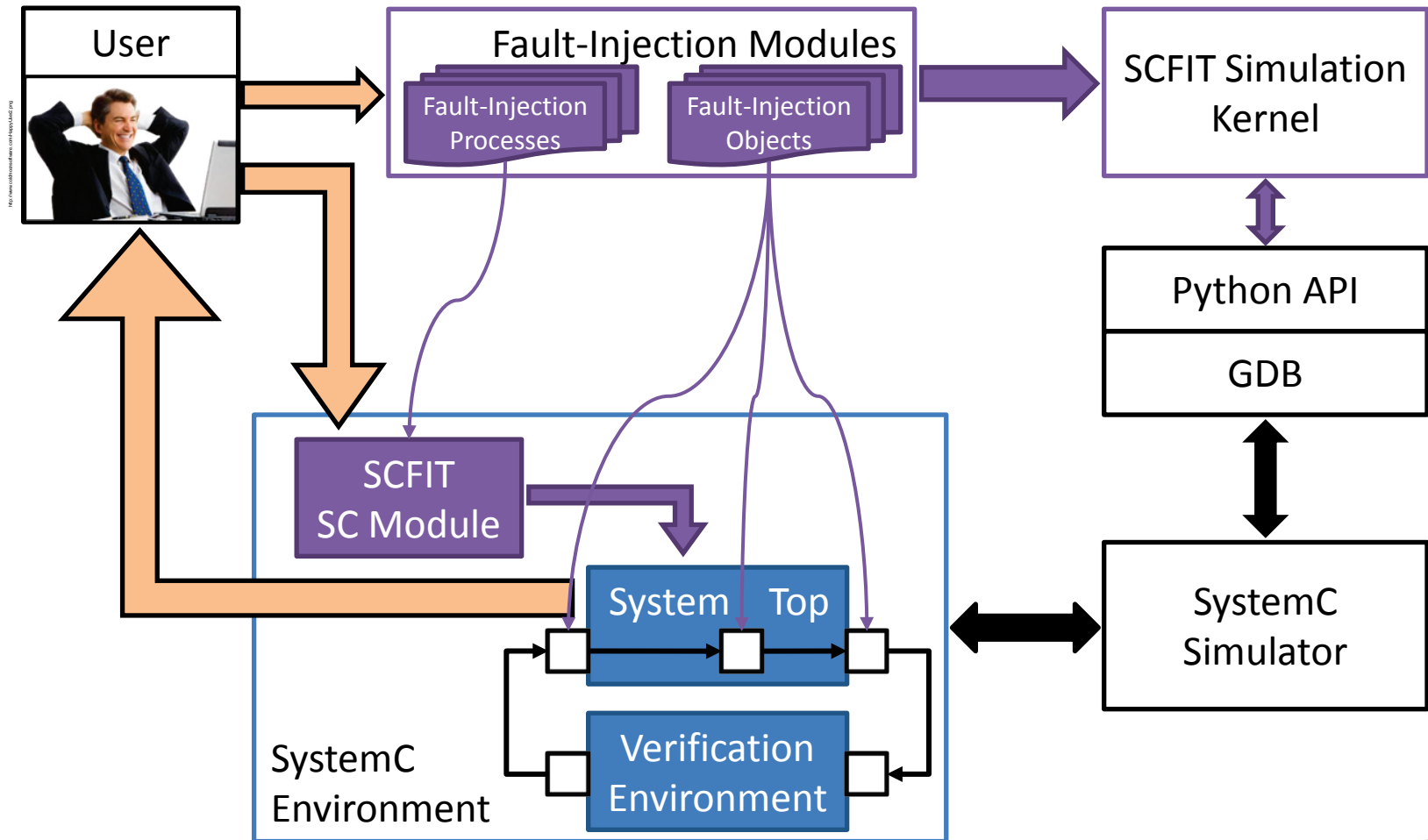
Fault-Injection Theory



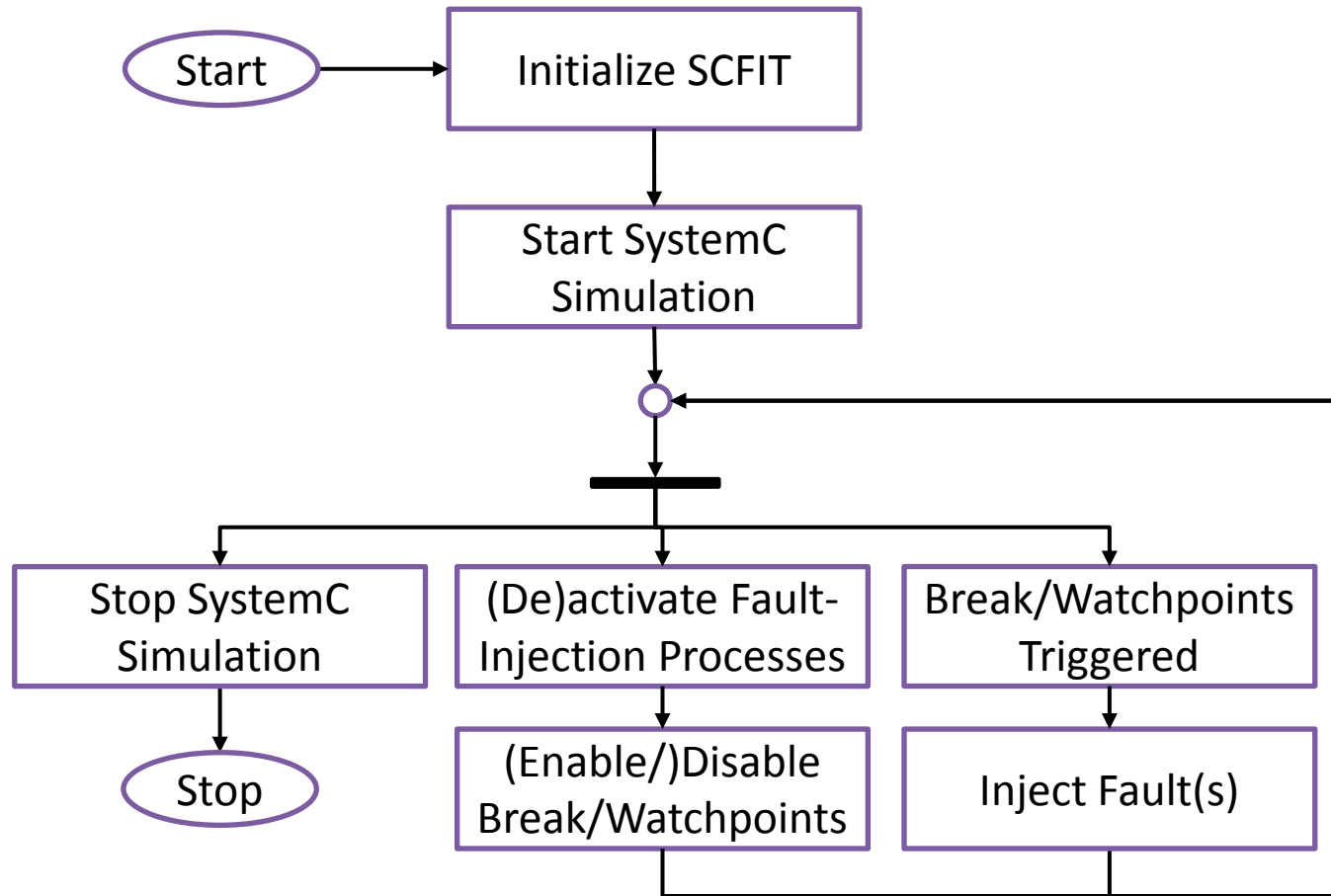
Fault-Injection Requirements

- SystemC models (e.g., TLM)
- Non-intrusive simulation based fault injection
- Access private/protected data members
- Separate fault behavior description from original behavior in models

SCFIT Architecture



SCFIT Flow Diagram



Results

- Simplified CPU architecture
- 12 bit wide registers
- 32 instruction set
- SystemC model
- TLM reference model

| SCFIT | # of Faults | Simulation Time (s) | Slowdown (%) |
|-------|-------------|---------------------|--------------|
| no | - | 3.71 | - |
| yes | 0 | 3.77 | 1.61 |
| yes | 1 | 4.10 | 10.51 |
| yes | 2 | 4.50 | 21.29 |

Discussion

Advantages

- Injection of any number of faults per simulation
- Fast simulation speed
 - For one fault per simulation
- Usable with O3 optimization
- User-friendly fault-behavior description
- Usable in regression runs

Disadvantages

- Dramatic slowdown for 2+ faults per simulation
 - Max 4 hardware-watchpoints
- Compiler debug-mode enabled in SystemC models
 - Not portable to customers

Next Steps

- GUI based generator
 - Fault-behavior description process
 - Documentation about injectable faults
- Automatic fault-detection and propagation analysis
 - Fault → Error → Failure
- Windows portability
 - GDB only available under Unix machines
- Simulation Speed Optimization
 - Migrate interpreted Python code to compiled C++ code

Questions

Thank you for your attention!

