DATE '05

Assertions for SystemC

Holger Keding – Synopsys

Agenda

- Motivation
- Assertions for TL Models
  - Using variables and events instead of signals
  - Re-use assertions across levels of abstraction
- Case Study: Simple Bus
- Summary

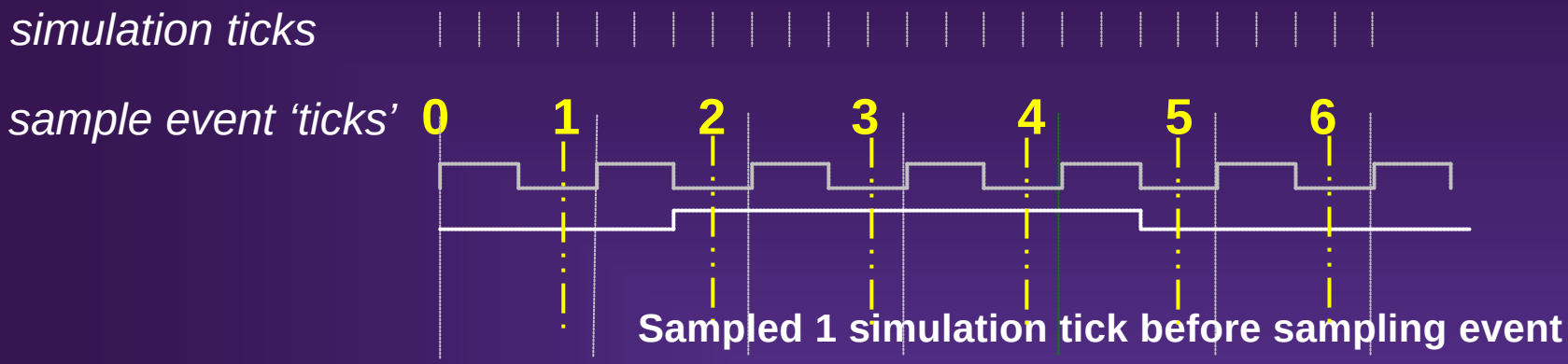
Motivation

- Assertions have become an important component in today's RTL verification environments
- Declarative language
  - Efficient and concise
  - Different to DUT language
    - Bad thing – need to think differently (consumes engineering time)
    - Good thing – need to think differently (avoids common mode errors)
- Assertions for SystemC?
  Easy solution – using same technology and methodology as for HDLs - is is no solution.
  - SystemC is used for abstract modeling
  - RTLish assertions are of no value

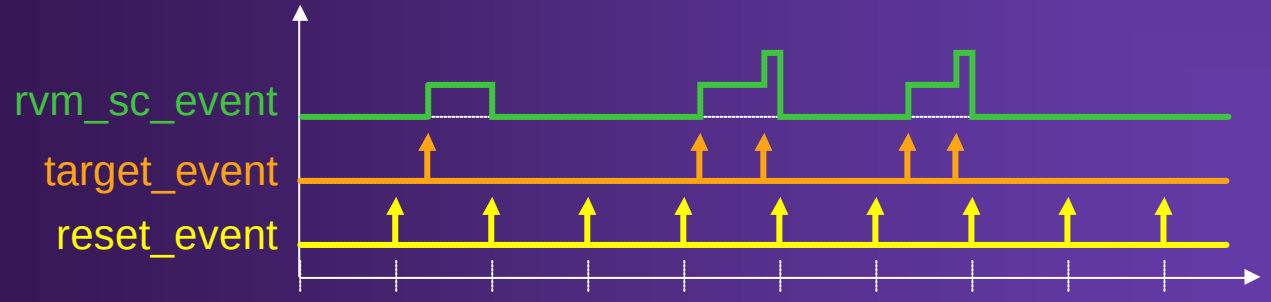
Assertions for TL Models
Technical Aspects

- TL Models:
  - Different timing accuracies – not necessarily using a clock
  - Signals are hardly used
  - Events are used for synchronization
  - Variables and Objects contain represent data and states
- Technical requirements for Assertions
  - Allow assertions to be triggered by events, not necessarily a clock
  - Bind Assertions to variables and events rather than to signals and registers

Value Sampling of Assertions



- **Assertions bound to variables: Treated as signals**
- **But how to sample a non-persistent sc_event?**
  - **rvm_sc_event class 'integrate' the events:**



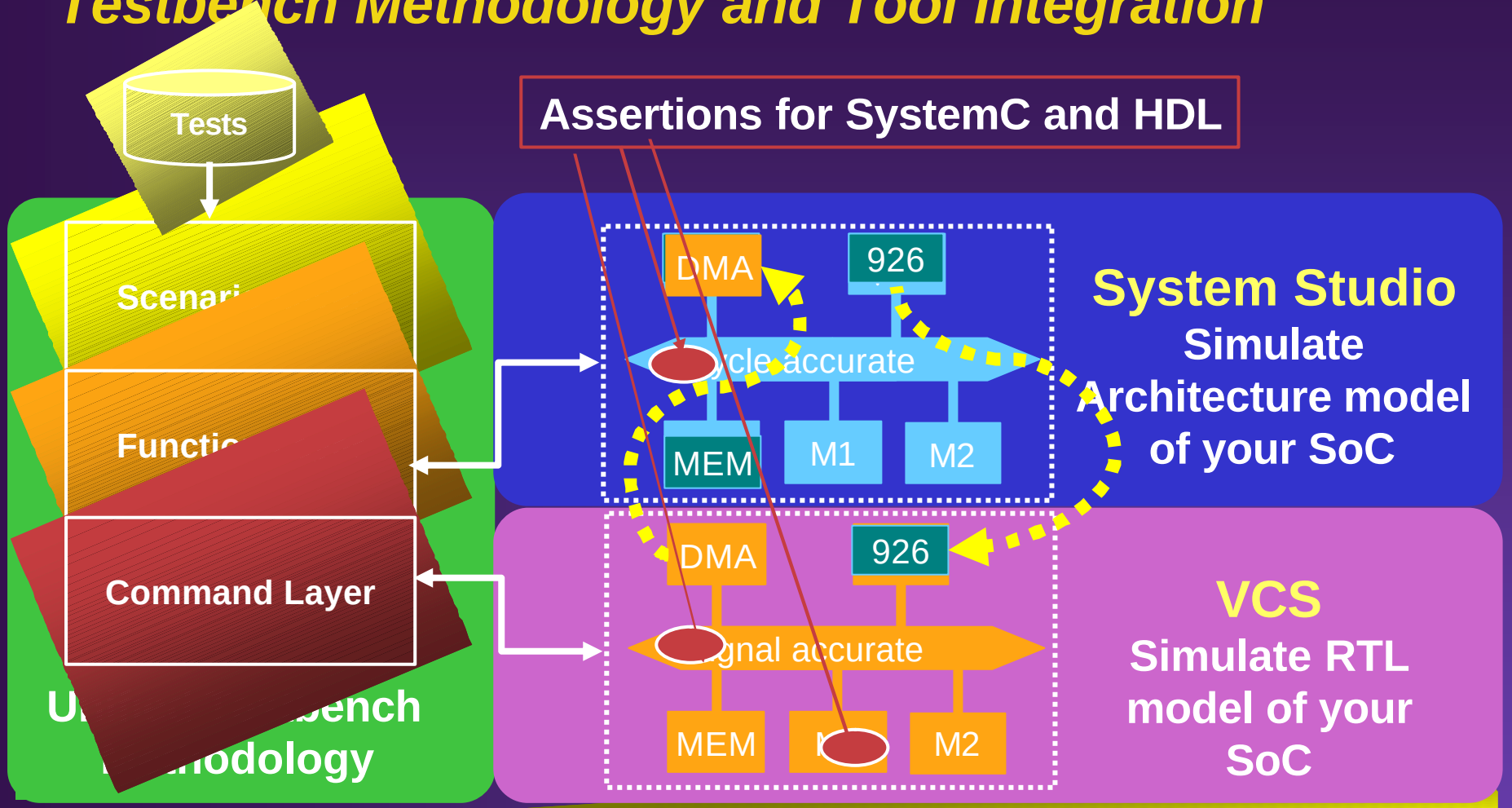
SYNOPSYS

Assertions for TL Models
Methodology Aspects

- TL Models:
  - Abstract from RTL models for
    - Architectural exploration
    - Early testbench development
    - Performance analysis
  - Re-used as reference models for RTL verification
- Methodology requirements for Assertions
  - Re-use Assertions as well
    - Ease re-use by a common Language for TL and RTL Assertions
    - Maintain System Level assertions in Assertion libraries

Unified ESL to RTL verification

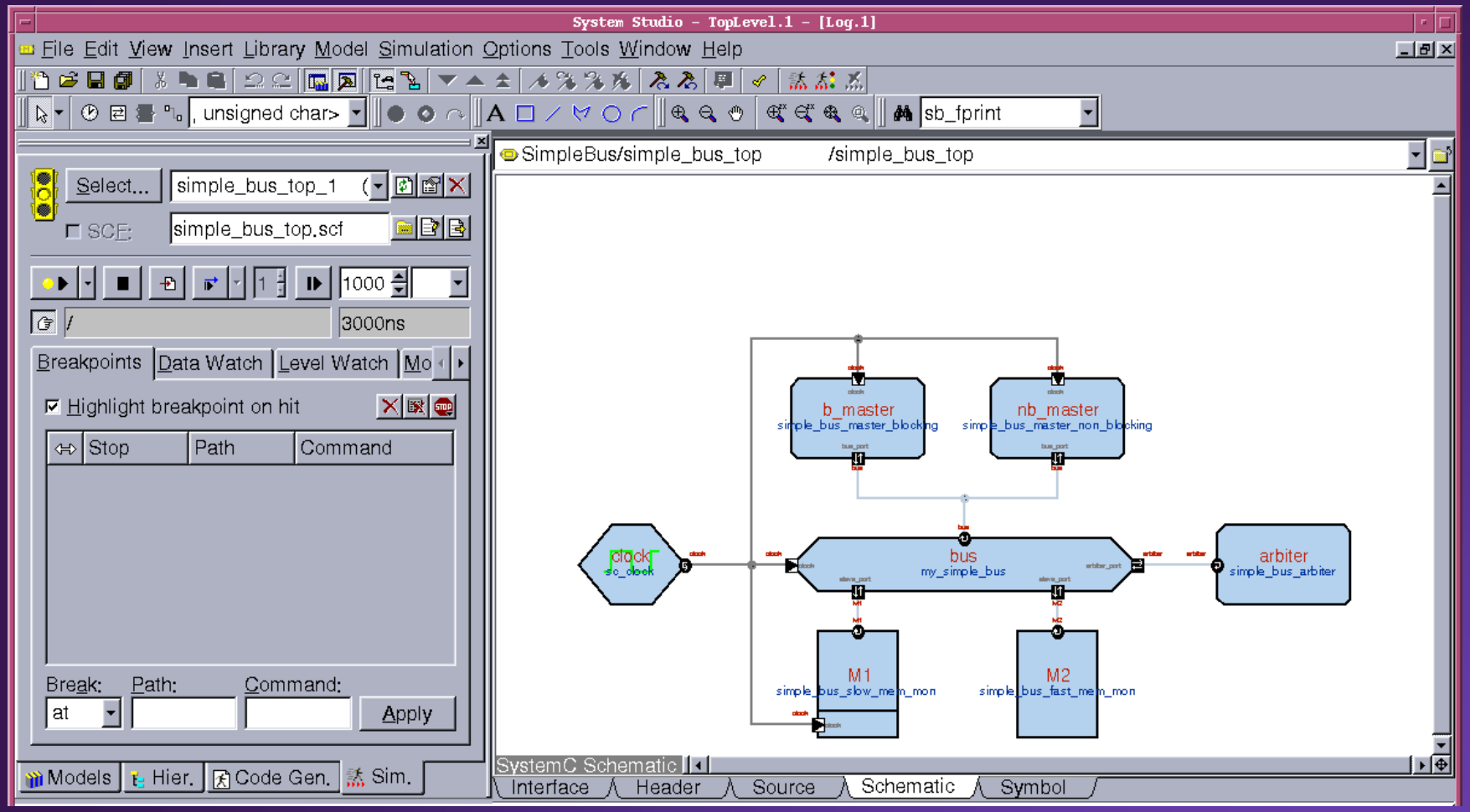
Testbench Methodology and Tool Integration



Ensure Consistent Architecture and RTL Models

SYNOPSYS®

Case Study: OVA/SVAs for the Simple Bus



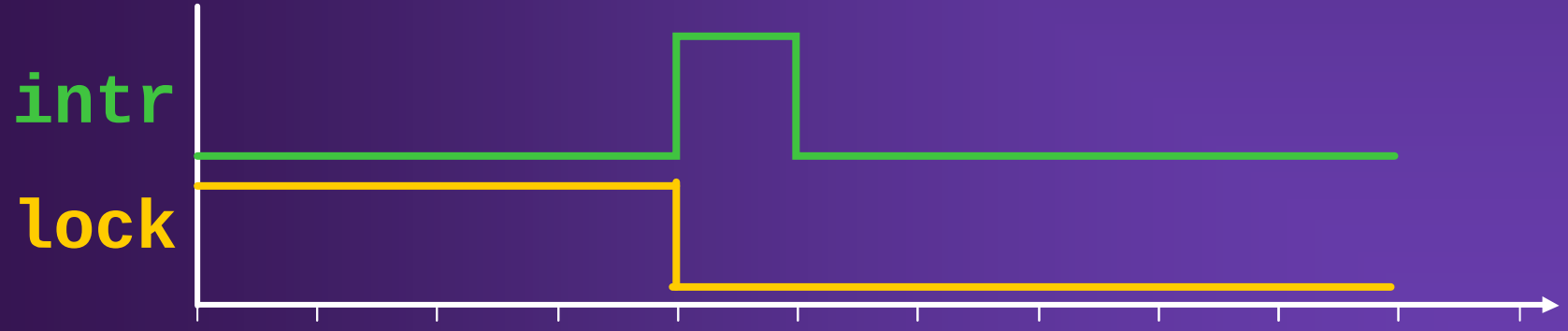
SYNOPSYS

Verify Correct Bus Locking

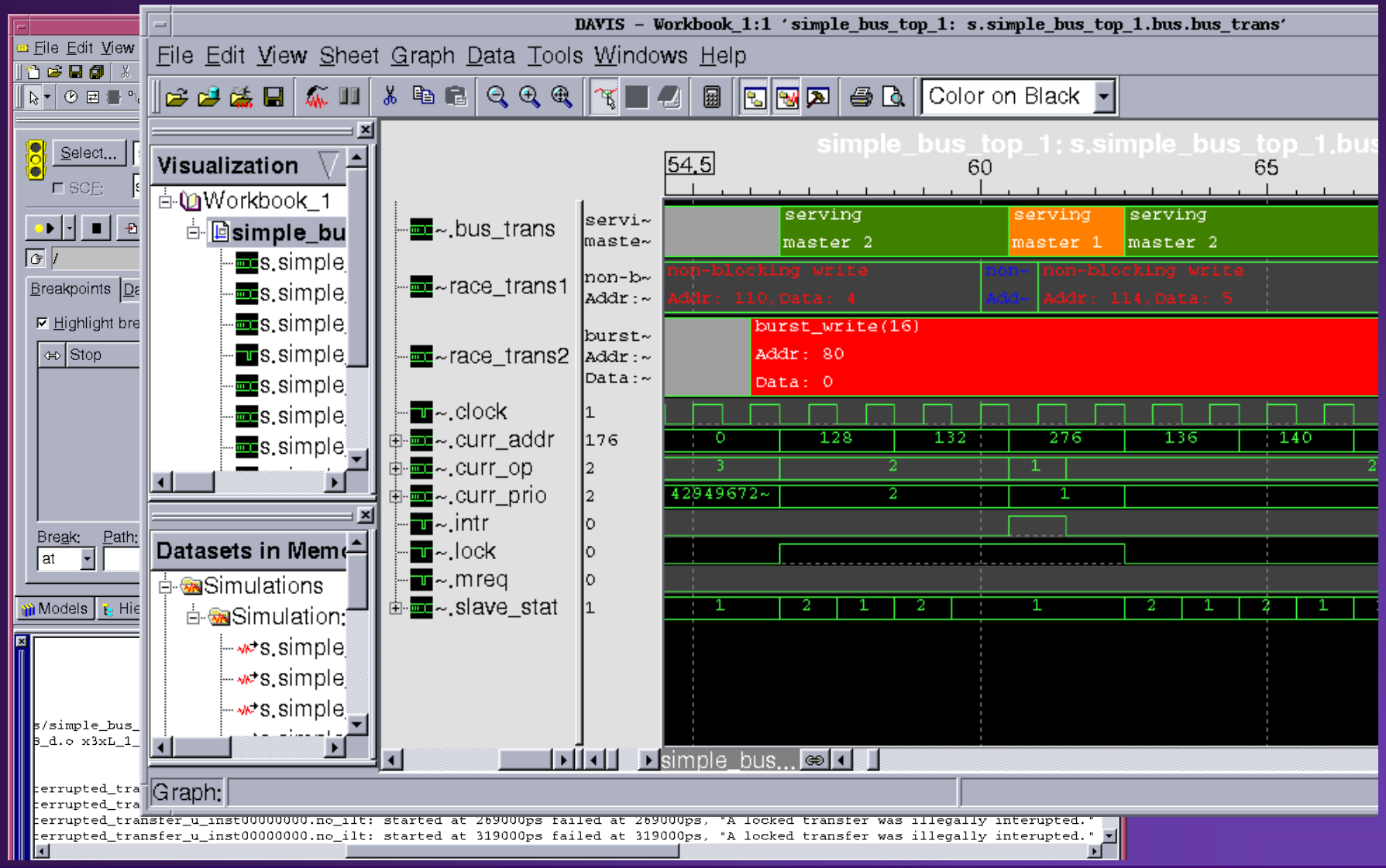
- A burst transfer can lock the bus to prevent being interrupted
- Check for bad transfer interrupt

```
event no_intr_lock: posedge intr && past(lock==1)
```

  - intr: interrupt a uncompleted transfer
  - lock: bus lock set for current transfer



Case Study: Simple Bus



SYNOPSYS

Summary

- Assertions can help to craft more efficient verification environments.
- Assertions for Transaction Level need to observe more than just signals.
- Assertions in a common language that are re-usable across TL and RTL increase verification productivity and quality.

SYNOPSYS®