Speed Up and Improve Verification by using a Generic Scoreboard Library

Dr. Martin Ruhwandl
Senior Staff Engineer IP Design and Verification

Processor Platforms and IP-Reuse
Communication Solutions

CDNLive! 2007 in Munich – Session 3.4

May, 15th 2007

Never stop thinking

Overview

- **Motivation**
- **Testbench Structure**
- **Scoreboard Library**
- **Final Remarks**

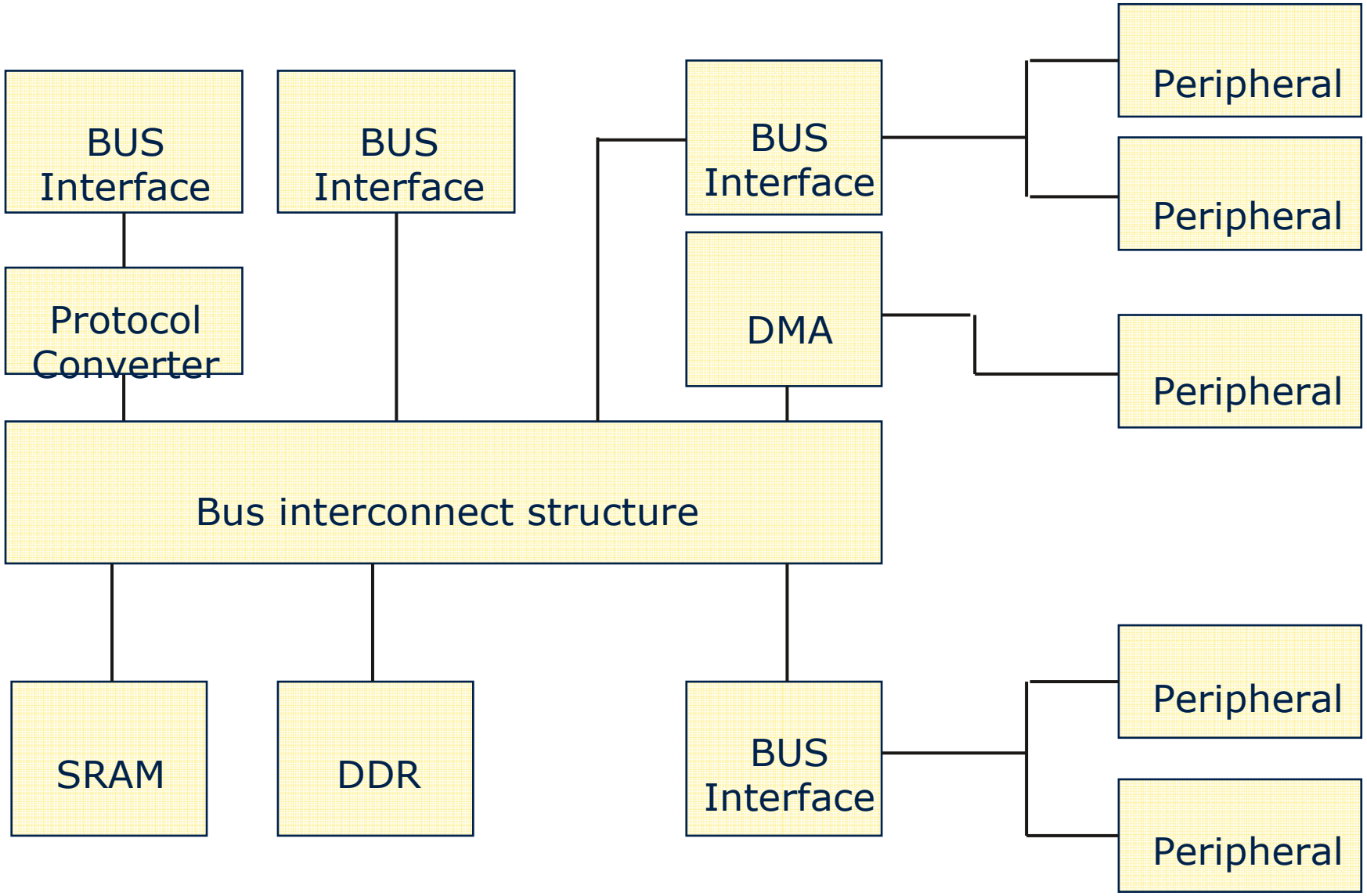
Overview

- **Motivation**
- Testbench Structure
- Scoreboard Library
- Final Remarks

Setting

- IFAG COM PS IPR: Processor platforms and IP-Reuse
- Development and verification of
  - processor subsystems (platform based)
  - standard IPs (e.g. line interfaces, timer, …)
- Varying effort
  - from complete development incl. implementation (RTL and Layout)
  - down to internal or external 3rd party IPs

Motivation: Typical Design

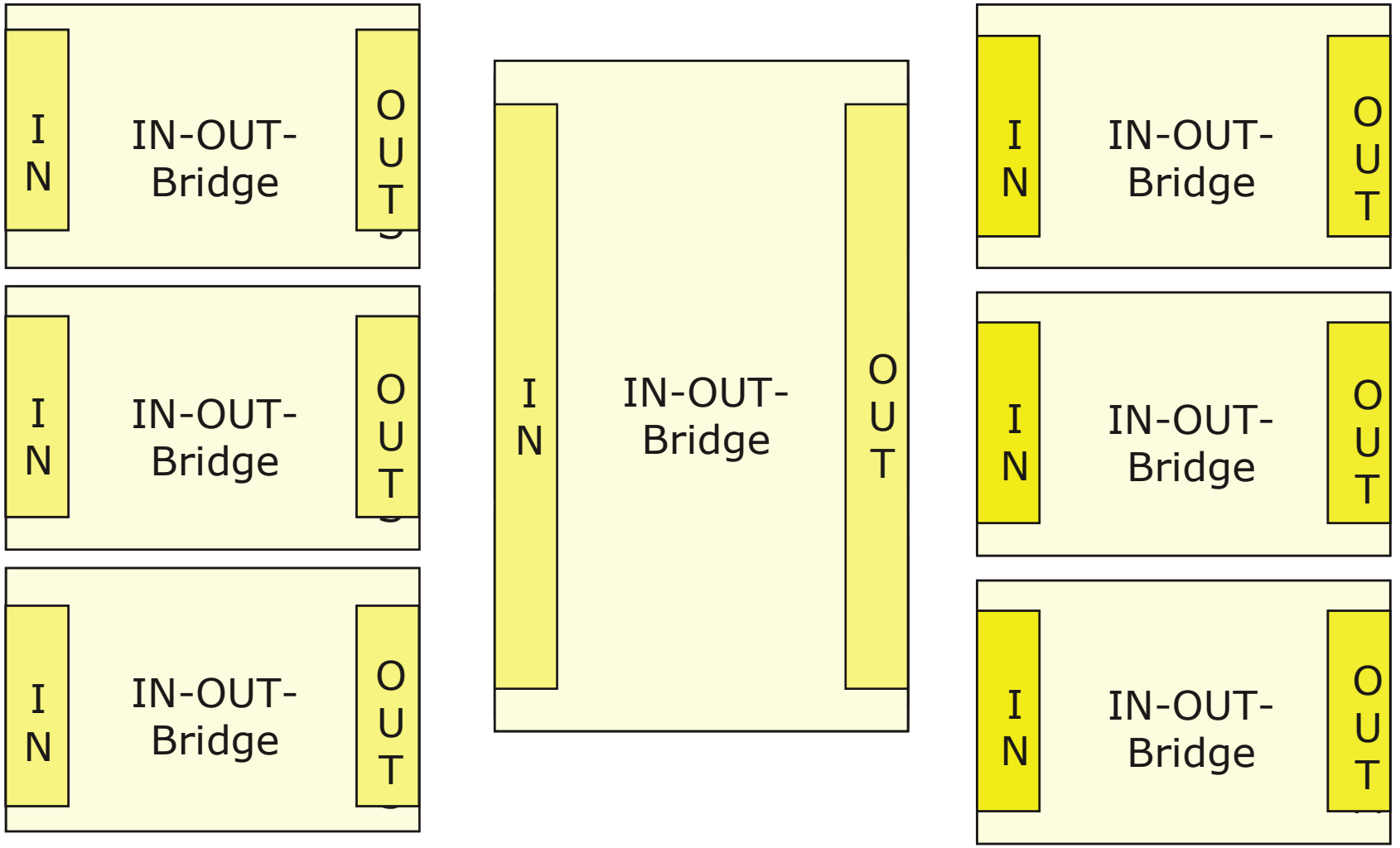


Motivation: Blocks to be verified

Forget the protocol

=> transaction level

and number of ports



Motivation

- Most of our designs are some kind of bridges
- One or more input and output ports
- Varying protocols
- Similar behaviour on Transaction level

⇒ Many parts of the scoreboards are very similar.

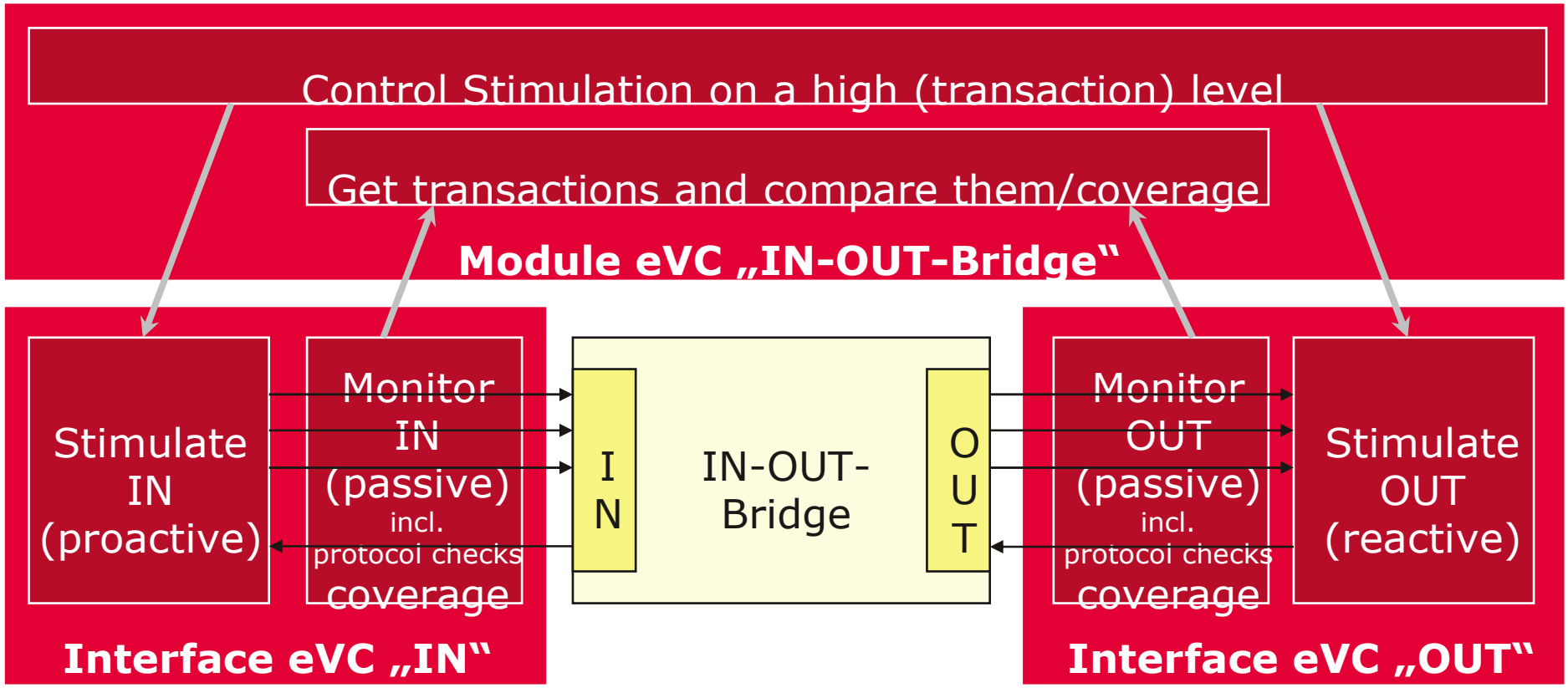
Reduction of development effort for scoreboard implementation possible?

Overview

- Motivation
- **Testbench Structure**
- Scoreboard Library
- Final Remarks

Testbench Structure

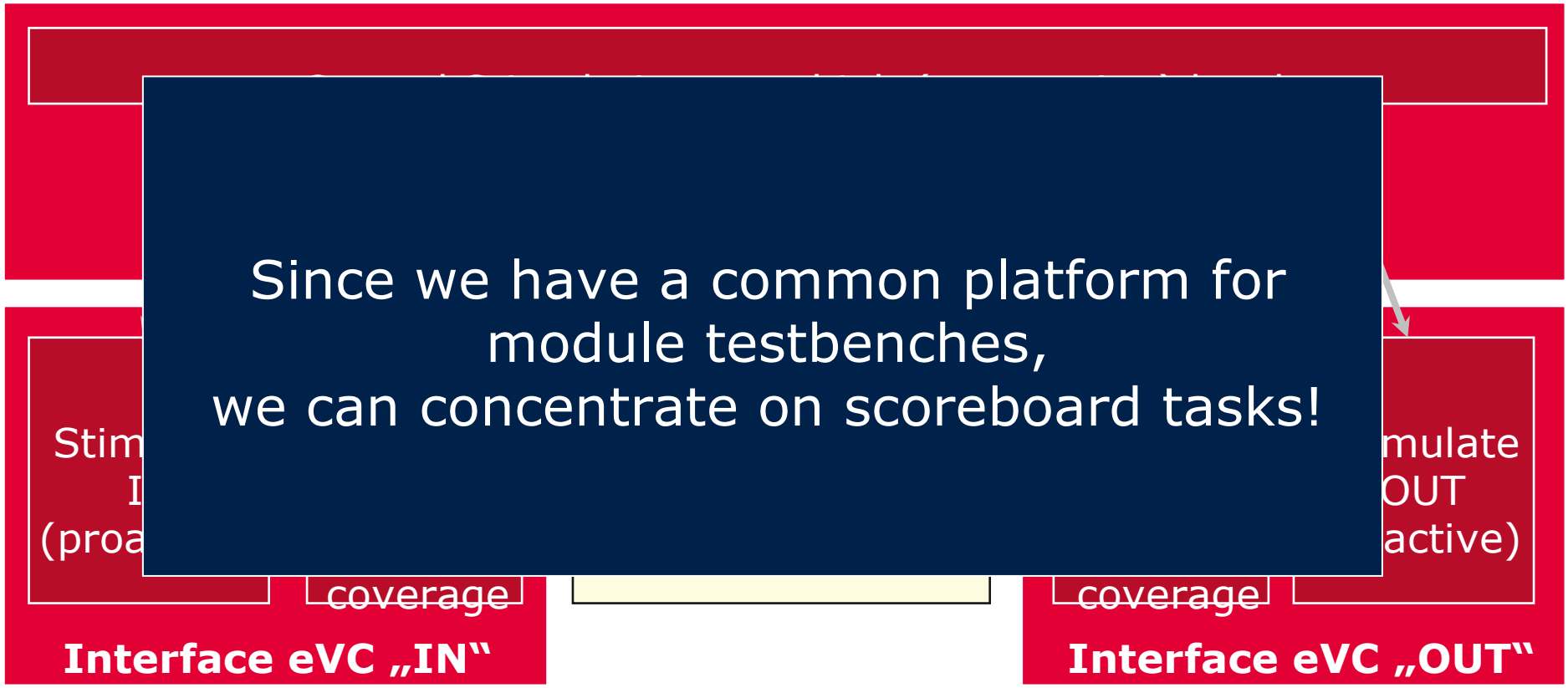
Common testbench structure



Important: strict split between interface and module related tasks;
Clear definition of interface between Interface- and Module eVC;

Testbench Structure

Common testbench structure



Important: strict split between interface and module related tasks;
Clear definition of interface between Interface and Module eVC;

Overview

- Motivation
- Testbench Structure
- **Scoreboard Library**
- Final Remarks

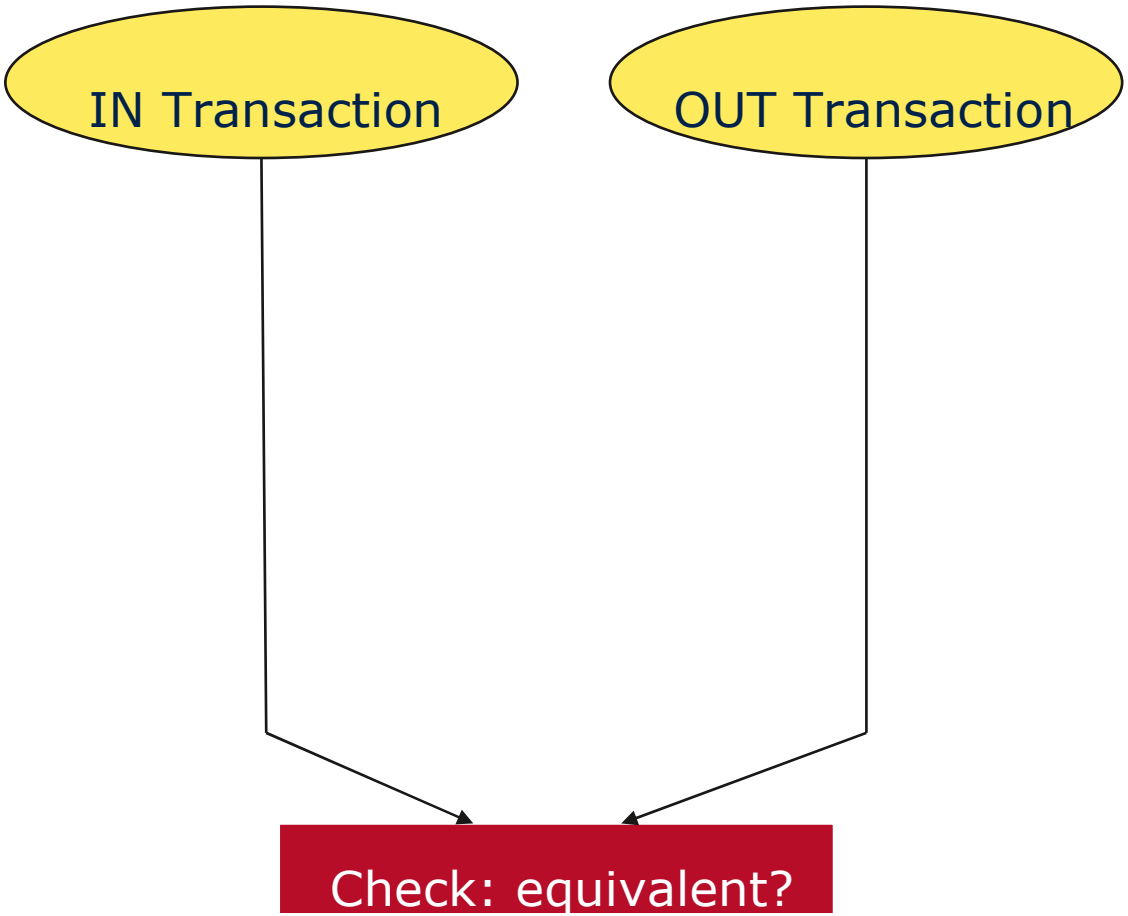
Scoreboard Library: Tasks

- Get transactions from input and output ports.
- DUT may
  - modify data from input to output (endianess, address map, …)
  - split one transaction into several transactions or vice versa
  - process incoming transactions internally only (e.g. for configuration or just delete transactions)
  - …

However, it's always a well defined and predictable behavior.

To obtain handsome modifications: find a good granularity for mapping DUT functionality to scoreboards (more than one scoreboard per module is ok).

Scoreboard Library: Flow (1)



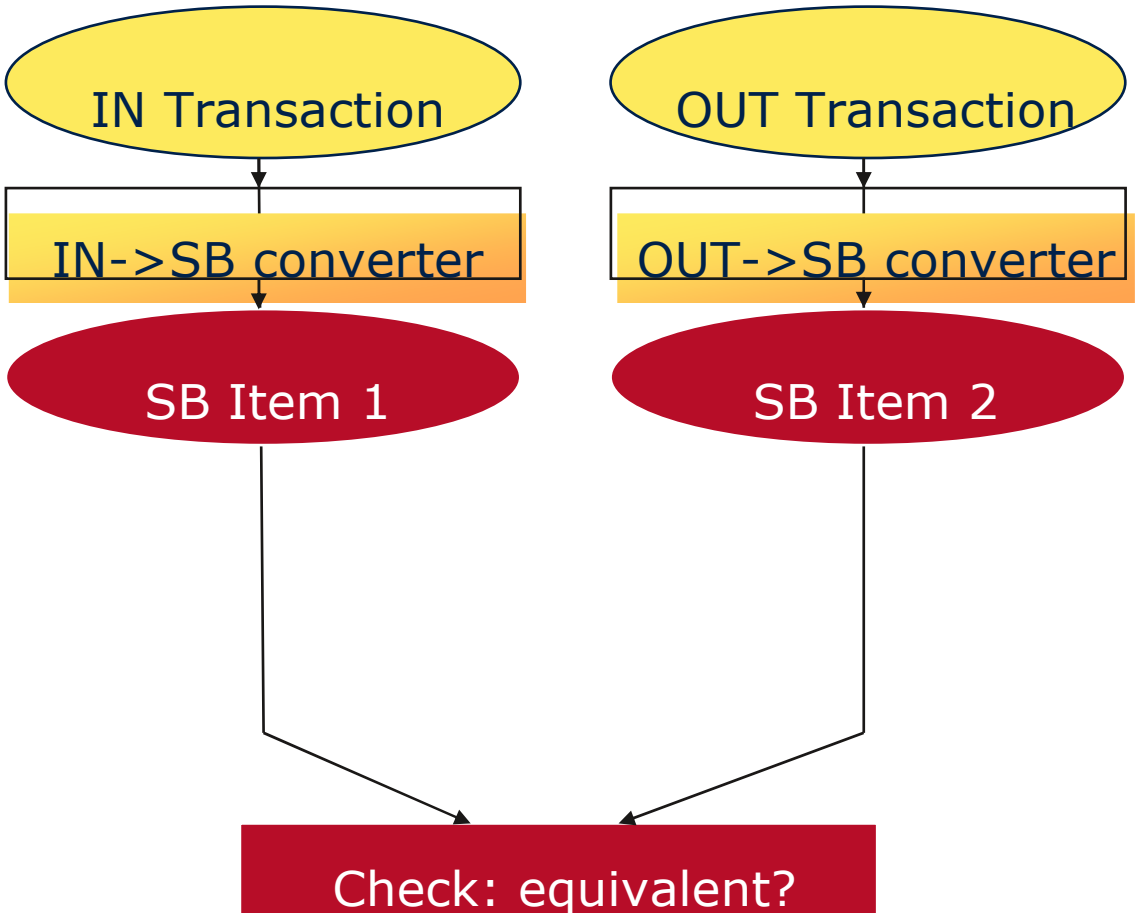
Scoreboard Library: Common Scoreboard Item

- Define transaction item w/o redundancy and create a common scoreboard item
- E.g., for bridge-like designs
  - command (opcode, read/write, byte enable, …)
  - address
  - data
- Now, comparison method can be implemented based on this item already.
- Conversion method from Interface-eVC monitor item to standard common scoreboard item can be part of the Interface-eVC.

Scoreboard Library: Flow (2)

Key is a standard scoreboard item:

Convert to standard scoreboard item



Scoreboard: Conversion Methods

- For easy comparison, add conversion methods to library, like
  - split burst transfer into singles (or other set of transactions)
  - convert endianess (little->big, big->little)
  - implementation width conversion (e.g. 32->64 bit width)
  - …

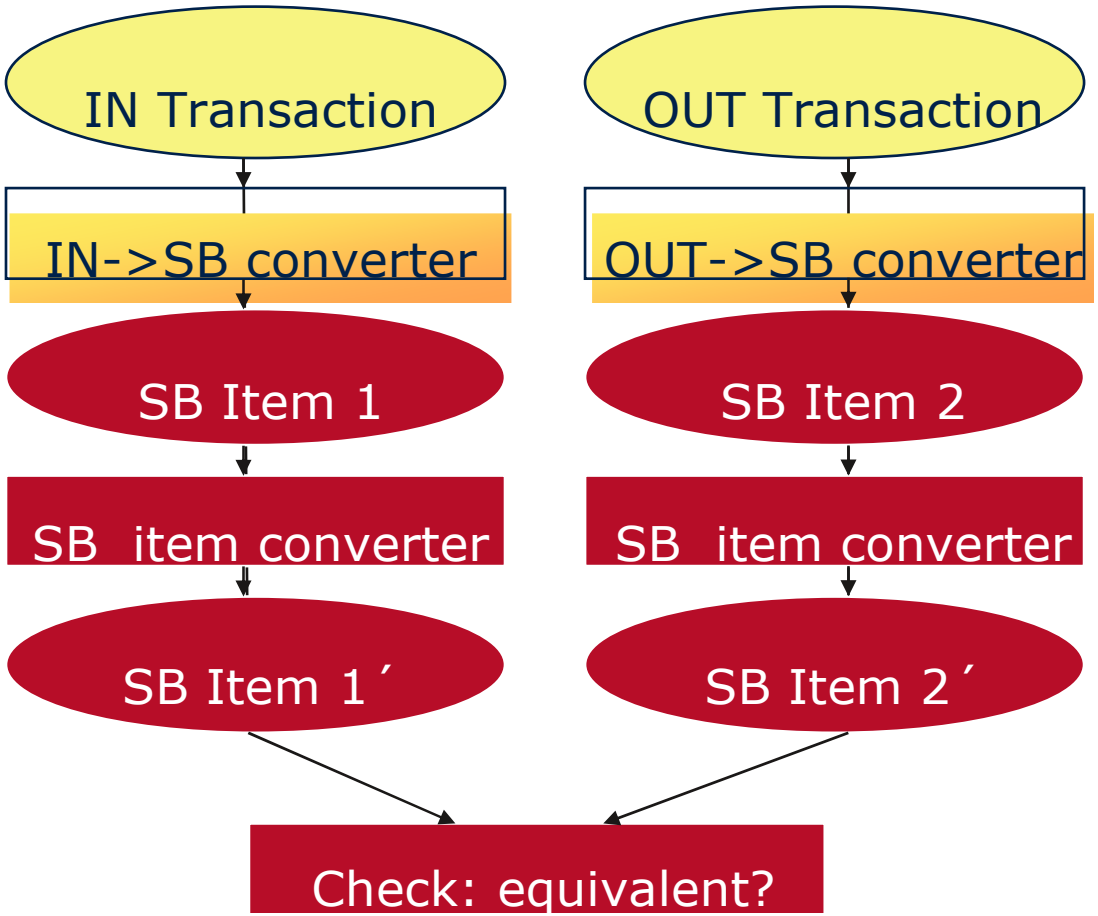
If necessary, modify in- and out-scoreboard items with this methods until they have an easy to check representation.

Scoreboard Library: Flow (3)

Key is a standard scoreboard item
+ modification and checking methods:

Convert to standard SB item

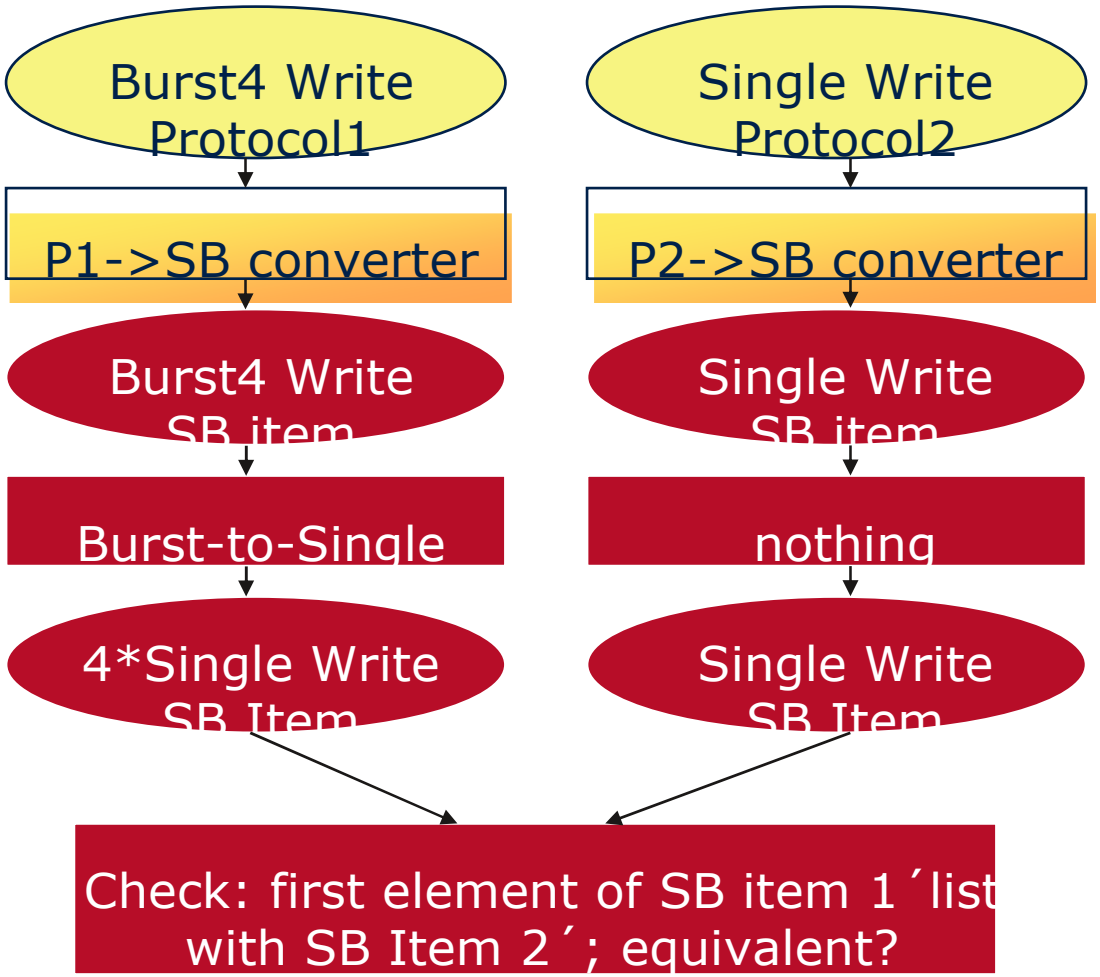
Modification for qualified and handsome checking



Scoreboard Library: Example

Convert to standard SB item

Modification for qualified and handsome checking



Scoreboard Library: Detailed Flow

- Always required for every module: Detailed scoreboard flow.
- Define flow of conversion methods and other DUT related stuff
  - which conversion methods must be applied,
  - item conversions: optional or configurable,
  - some transaction may occur only on one port,
  - possible connection of register model,
  - timing (reordering of transactions possible?),
  - additional DUT specific tasks.

- If standard scoreboard item doesn't meet the requirements of the redundance-free DUT transaction item
  - extend standard scoreboard item w.r.t. DUT,
  - extend conversion and comparison methods accordingly.

Overview

- Motivation
- Testbench Structure
- Scoreboard Library
- **Final Remarks**

Final Remarks

- We reduced our effort to create scoreboard for bridge-like designs dramatically (effort reduction: up to 80%-90%).
- Common approach eases verification engineers' teamwork.
- Comparison flow is established and meets most of our DUTs' requirements.

Infineon –
Never stop thinking

Never stop thinking