

# AMBA Compliance Checking with SolidPC



# SolidPC Overview

- ❖ AMBA™ compliance based on Static Functional Verification
- ❖ Co-developed with ARM
- ❖ The goal:

To *exhaustively* prove that a design obeys the AHB protocol rules



# Current Verification Methods

- ❖ Simulation based
  - ❖ Never have enough cycles, only check subset of functionality
  - ❖ Emulation, h/w accelerators very expensive
- ❖ **You** must think of the interesting cases to check
- ❖ Simulation **proves** your design works for the cases you have checked

## A New Approach - Property Checking

- ❖ State properties that blocks in your design should, or should not, possess
- ❖ Solidify then formally proves there is no combination of inputs for which that property does not hold true
- ❖ No testbenches to write – just properties!



# Writing Properties

- ❖ Written in a Verilog like language:
- ❖ `mode == add => result = a + b;`
- ❖ `Req => T(5,Ack);`
- ❖ `Ack => T(-5,Req);`
  - ❖ How do you do that in simulation?
- ❖ Solidify supports OVL, PSL, OVA...



**averant**

# Levels of Confidence

- ❖ For a property that passes exhaustively there is no combination of legal inputs that can fail
- ❖ Very hard (often impossible) to achieve that level of confidence with simulation

# SolidPC

- ❖ SolidPC *exhaustively* proves that a design obeys the AMBA protocol rules
- ❖ Protocol rules are written and approved by ARM
- ❖ **No test-vectors are required**
- ❖ **No simulations are run!**



# How Does SolidPC Work?

- ❖ SolidPC converts each AHB protocol rule into a mathematical property
- ❖ Averant's Solidify™ verification engine is then used to prove the properties
- ❖ For each rule, SolidPC will return one of three possible results:
  - ❖ Rule *exhaustively* verified
  - ❖ Rule fails
  - ❖ Rule is inconclusive



# ***Rule Exhaustively Verified***

- ❖ Proves that design will work under all legal input sequences
  - ❖ All corner cases are automatically considered
- ❖ Guarantees that design satisfies that part of the protocol standard
- ❖ Often very difficult or impossible to do in simulation!



## Rule Fails

- ❖ A counter example found which shows the design fails to comply with the protocol rule
- ❖ SolidPC generates a set of vectors, starting from reset to demonstrate the non-compliance
- ❖ Automatically creates a VHDL or Verilog testbench for debugging in a simulator

## Rule Inconclusive

- ❖ Tool cannot prove the design correct, nor find a fail that starts with the reset condition
- ❖ SolidPC exports a simulation monitor that can be included inside customers test bench.
  - ❖ Monitors check the correctness of the protocol rules during simulation
- ❖ Experience at **ARM** indicates that this result only occurs in a small percentage of cases.



# The Simulation Alternative

- ❖ Currently simulation based products are used to verify AMBA compliance
- ❖ They contain mixture of assertions to check compliance and coverage points to measure completeness
- ❖ We typically hear of around 10 simulation licenses being used concurrently for 3 months when using this approach
- ❖ The results are still not exhaustive



## Further Info

❖ For further information please contact:

Carey Sayer

[carey@saros.co.uk](mailto:carey@saros.co.uk)

Jeremy Sonander

[jeremy@saros.co.uk](mailto:jeremy@saros.co.uk)

Saros Technology Ltd

[www.saros.co.uk](http://www.saros.co.uk)



# Summary

- ❖ SolidPC is a break through technology for AMBA compliance checking
- ❖ ARM has developed and endorsed the protocol rules
- ❖ Exhaustive proofs
- ❖ Easy to use
- ❖ Quick to run
- ❖ SolidPC for AMBA 3.0 AXI development underway

[carey@saros.co.uk](mailto:carey@saros.co.uk); [Jeremy@saros.co.uk](mailto:Jeremy@saros.co.uk)

+44 1491 837787

