

SolidPC for AMBA 2 AHB

The definitive AMBA AHB Compliance Checker

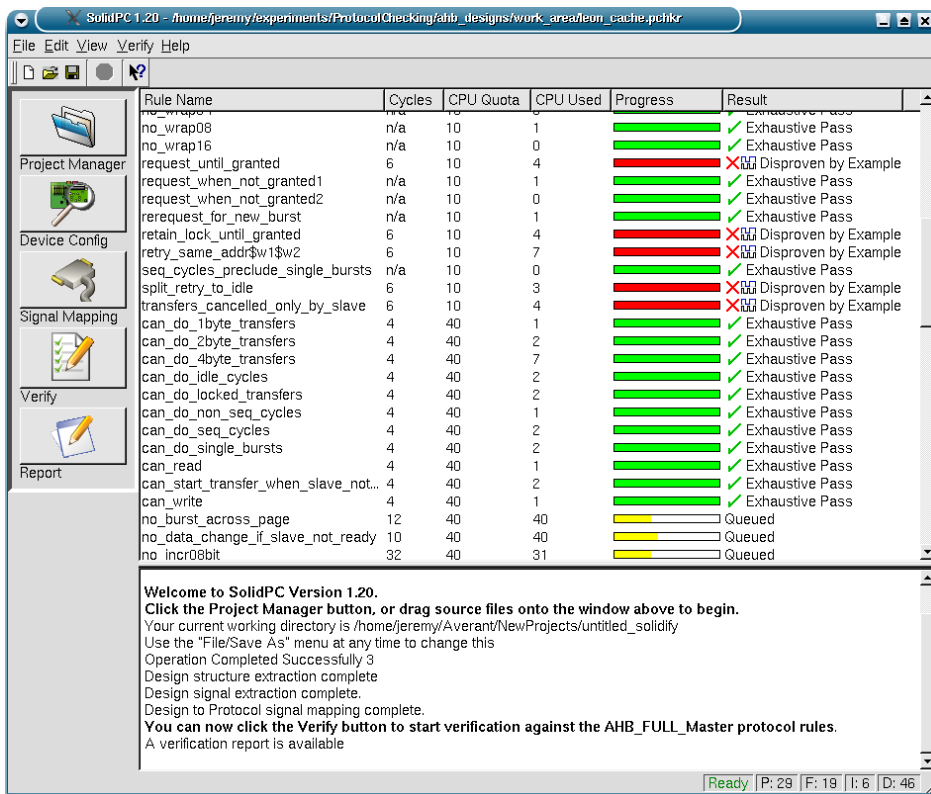


Figure 1

Product Features

- Complete functional verification environment for the following AMBA protocols:
 - AHB Lite
 - AHB full (includes splits & retries)
 - APB
- Rule sets developed and maintained by ARM
- Based on formal methods - exhaustive proofs
- No simulation vectors, testbenches or properties to write
- Easy to use
- Quick to run
- Checks VHDL and Verilog designs
- Linux, Solaris and Windows support

Introduction

Today's complex system-on-chip designs frequently incorporate embedded processors and rely upon a bus protocol to communicate between different blocks in the system.

The ARM range of 16/32 bit microprocessors leads the embedded microprocessor market and the associated AMBA bus protocols have become the

first choice of companies designing complex, state of the art system-on-chip.

One of the key verification tasks in the design of these systems is to ensure that each of the design units in the system obeys the interconnecting bus protocol. Failure to achieve this can result in poor product quality, ASIC re-spins, and delayed entry into the

market with corresponding loss of revenue.

Existing protocol checking methods are typically simulation based. They require the creation of large numbers of test vectors, have difficulty in exposing corner cases and will often take man months of effort and a large investment in EDA tools.

SolidPC Overview

To address the issues of protocol compliance, ARM & Averant Inc. have collaborated to develop **SolidPC**, a static property-checking tool. **SolidPC** implements a formal and exhaustive proof of correctness against a given set of properties. Each property (or rule) describes a specific aspect of the AMBA protocol and, taken as a whole, the complete property set will fully verify a design against the required AMBA protocol.

Each protocol rule has been expressed in Averant's Property Checking language, thoroughly verified by Averant and ARM, and embedded within the **SolidPC** product.

SolidPC combines a new user interface, the AMBA protocol rule sets and the Averant's formal engine to offer push-button verification of designs against the AMBA protocols.

The combination of the new GUI, the pre-written protocol rules and the power of the formal engine make **SolidPC** easy to use and quick to run. The property sets are used internally by ARM and are endorsed by ARM for use by their customers.

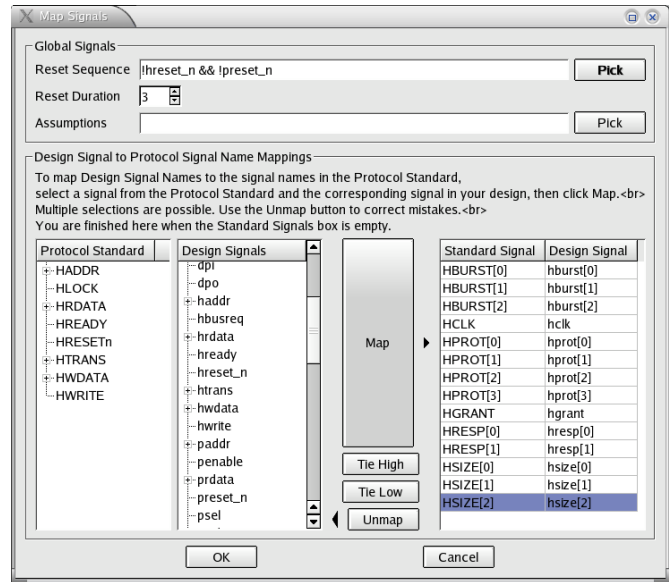


Figure 2

No test vectors, testbenches, coverage points or simulations are required.

Methodology Flow

SolidPC, with its intuitive user interface, guides the user through the necessary steps to establish that a given design under test is compliant with the AMBA protocols. The process starts with mapping the signal names in the design to the names defined in the AMBA standard. (Figure 2)

Next, the rules for the selected protocol are displayed and the user may select which of these are to be applied to the design under test. (Figure 3) Many AMBA interfaces only implement a subset of the protocol. If a rule is deselected a contrary rule is enabled stating that the unit under test can never perform this operation.

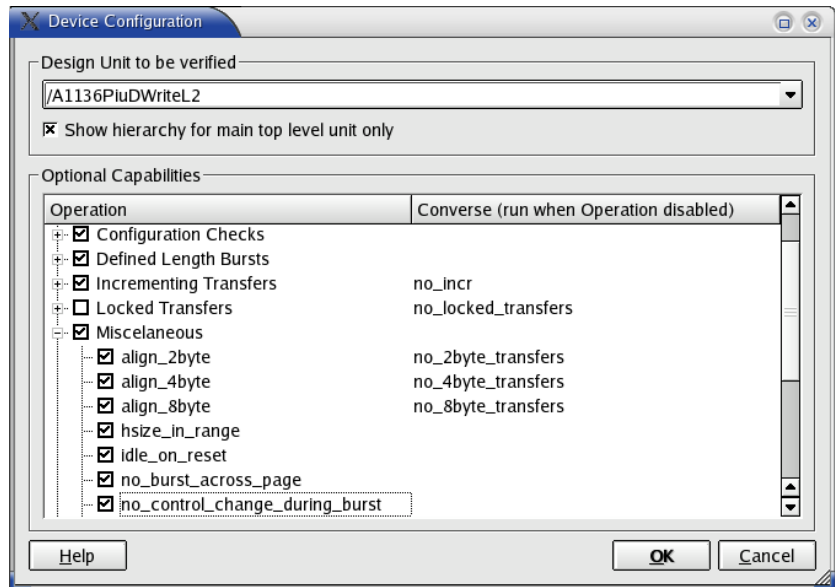


Figure 3

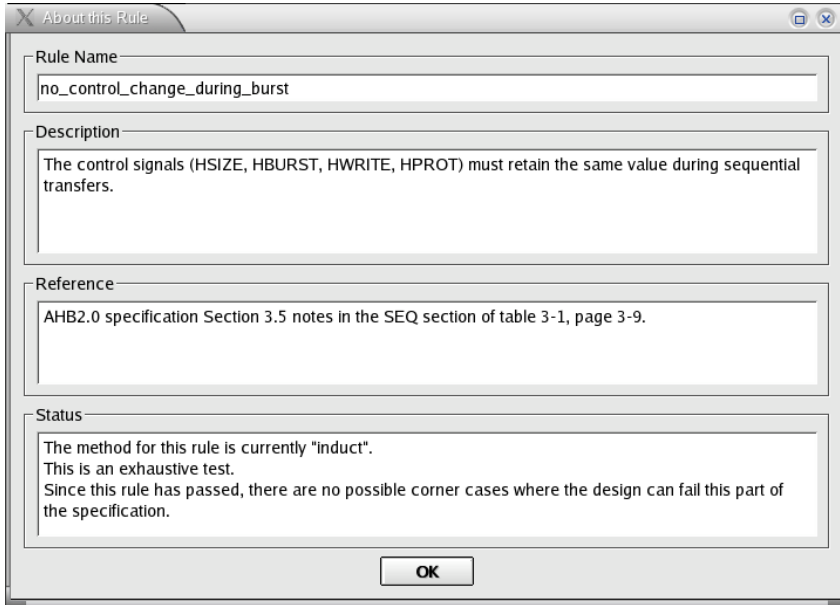


Figure 4

Verification now commences. Each protocol rule is taken in turn and submitted to the formal engine for proving against the design under test. **SolidPC** explores every possible legal combination of inputs the design may experience to determine if the rule holds true in all cases. If it does, the rule is passed *exhaustively*.

For rules that pass exhaustively there are no input conditions under which the design could fail that aspect of the protocol, something very difficult to establish with simulation. The running of these rules is also fast, with properties often passing exhaustively in seconds.

For each rule **SolidPC** applies to the design there are three possible outcomes:

- ◆ The rule passes exhaustively: The design passes for all possible legal input values.
- ◆ The rule fails: **SolidPC** will produce a VHDL or Verilog testbench that instantiates the design and creates the precise input vectors that, when applied to a simulator, will cause the design to fail that protocol rule.
- ◆ The rule cannot be proved exhaustively, but passes for a given number of cycles after reset. **SolidPC** will produce a simulation monitor for further analysis of the rule within a simulation environment.

Failing rules are listed along with an explanation of exactly why they have failed, and a reference to the section in the AMBA specification where the rule is defined. (Figure 4)

At any point in the verification process a report is available showing the verification status of each rule. Proved rules are listed, along with any user given assumptions required to make them pass. (Figure 5)

Verification report for Design: /AHB_IF

This block has been verified using some of the protocol ruleset: AHB_Master. Failures have been found during this verification run.

Category	Number	Percentage
Passed Exhaustively	37	48%
Failed	11	14%
Inconclusive and exported for Simulation	2	3%
Disabled	27	35%

1.1 Breakdown and Analysis

Exhaustively Verified

The following rules were exhaustively proven to hold, given the restrictions you have specified. Restrictions: !HUNALIGNW && HCLKRWEN
If you are confident these restrictions are valid, then no further verification work is needed for these rules.

burst_length_correct	no_busy_after_burst_end2	idle
no_seq_at_start	no_busy_when_idle	no_
no_control_change_when_slave_not_ready	no_data_change_if_slave_not_ready	no_
no_burst_across_page	addr_incr08	adc
no_wrap08	no_wrap16	no_
can_do_1byte_transfers	can_do_2byte_transfers	car
can_do_busy_cycles	can_do_idle_cycles	car
can_do_seq_cycles	can_read	car
can_do_incrementing_bursts	can_do_wrapping_bursts	trar
cannot_start_transfer_when_slave_not_ready	can_stay_idle_when_slave_not_ready	bus
busy_to_idle_only_if_slave_error	can_stay_busy_when_dlb_snr	car
can_stay_busy_when_udlb_snr	no_write_data_change_if_slave_not_ready1	no_

Figure 5

Summary

In order to reduce risk and meet time-to-market pressures, verification of compliance with bus protocols in a rigorous, efficient and timely manner for system-on-chip designs is essential. **SolidPC**, designed for AMBA bus protocol compliance checking and verification, represents a major breakthrough in terms of the quality of the verification results achieved, ease of use and time taken to complete the task.

The combination of ARM developed and endorsed AMBA rule sets, the formal nature of the Solidify engine and ease of use, makes **SolidPC**, the tool of choice for AMBA protocol verification.

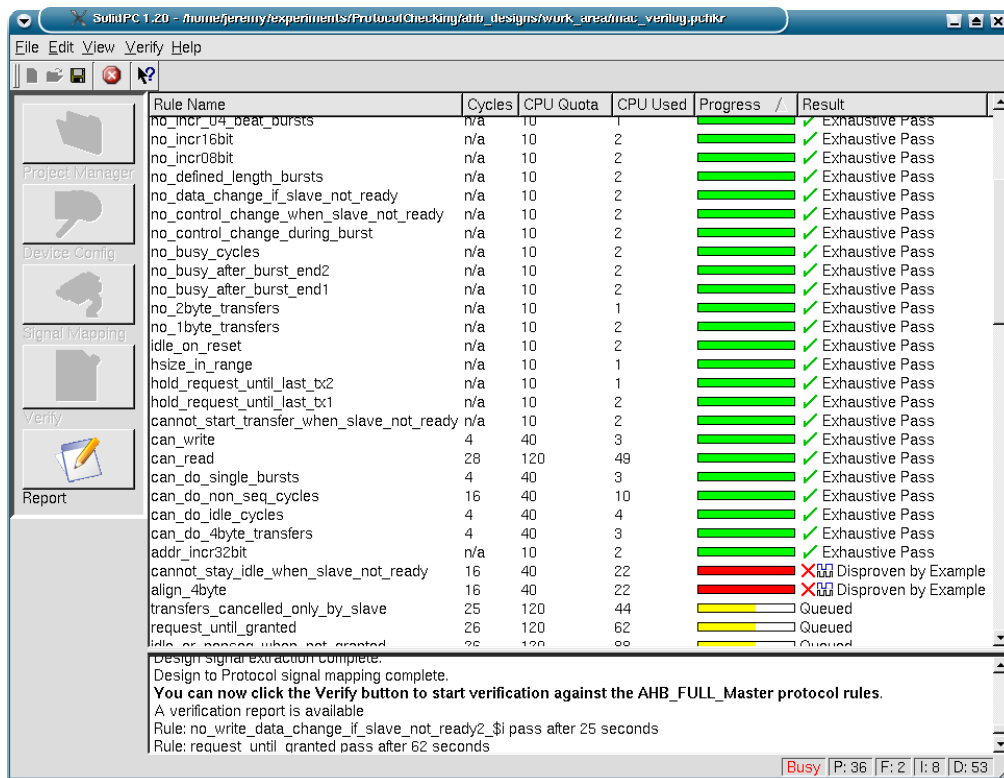
“Formal methods offer an innovative new approach for the verification of AMBA bus protocols, as the static nature means that verification is much faster than through simulation, and corner cases are more readily exposed.

The definition of correct AMBA technology behaviour, through the development of AMBA protocol rule sets, has done much to enhance the value of AMBA technology within the industry.

*By integrating our AMBA protocol rule sets into the **SolidPC** tool, developers will now have a high-performance verification tool that can rigorously check designs against the AMBA protocol rules.”*

Tim Mace
AMBA Product Manager
ARM

Figure 6



Contact us today for your free evaluation licence

Tel: +44(0)1491 837787
Fax: +44(0)1491 837477
Email: info@saros.co.uk

Saros Technology Ltd
20/21 Market Place
Wallingford
Oxon
OX10 0AD