TITLE OF PAPER	**PSS beyond reuse: Streamlining the DV effort for a low power multi-core SOC**
AUTHOR 1	Name: Vivek Gopalkrishna Organization: Analog Devices Job Title: Senior Engineer Email ID: Vivek.Gopalkrishna@analog.com Mobile no: 8792324950
AUTHOR 2	Name: Nitish Swamy Organization: Analog Devices Job Title: Senior Engineer Email ID: Nitish.Swamy@analog.com Mobile no: 7411393620
AUTHOR 3	Name: Ponnambalam Lakshmanan Organization: Analog Devices Job Title: Senior Manager Email ID: Ponnambalam.Lakshmanan@analog.com Mobile no: 9945089481
AUTHOR 4	Name: Santoshkumar Pathak Organization: Analog Devices Job Title: Senior Manager Email ID: Santoshkumar.Pathak@analog.com Mobile no: 8792324950
AUTHOR 5	Name: Sonal Patil Organization: Cadence Design Systems Job Title: Senior Engineer Email ID: sonala@cadence.com Mobile no: 7996742047

ABSTRACT

Increasing SOC complexities further aggravate the verification challenges in a multi-processor system. This highlights the need for streamlining the stimulus generation efforts across all platforms and verification teams, to reduce the time-to-market. The natural answer to this is portable stimulus, which has evolved as a standard over the years and offers a robust reuse methodology that has been embraced in this paper. Handling of a multi-processor verification setup, exhaustive testing of datapath scenarios, facilitating true reuse of test intent from IP to system level and across platforms, verifying power transition scenarios, accelerated coverage closure and efficient regression management were the focus areas for PSS. Starting from out-of-the-box memory access tests enabling verification from Day0, to complex stress testing scenarios exercising multiple cores at system level, the various wins achieved through PSS are described in this paper.

BACKGROUND

Creating sufficient tests to verify today's complex designs is a key verification challenge, and this challenge is present from IP block-level verification all the way to SoC validation. In cases of multi core designs, there is an added complexity. The tests must be multi-threaded to handle the various core

operations as well as drive stimulus to the UVM based testbenches. This is traditionally done by manually creating tests which have the right synchronization handshakes between the multiple threads. In certain scenarios it is needed to switch between having core and coreless architectures. For instance, in block level tests, it may be sufficient to use a BFM model with a coreless design. The tests created for such an environment cannot be ported directly to a subsystem or system level verification with the actual cores in place. Also, coverage closure being the key metric driving the verification effort, it is required to create meaningful tests that add to the overall coverage and make sure that these are not redundant tests.

This paper outlines the PSS efforts built on top of the previous work presented at DVCON 2023 titled "Tackling the verification complexities of a processor subsystem through Portable stimulus". This includes how PSS can be used for a power-aware multi-core SOC to establish test intent reused across multiple platforms and also how PSS can be used for streamlining the overall DV effort of coverage closure through test-gen coverage, efficient regression management and showcasing the cornercase bugs found only through stress testing concurrent scenarios.

DESCRIPTION

The SOC architecture is shown below. It includes 4 processor cores of which 2 are from the ARM family and 2 are from the Cadence Tensilica family. There are multiple flavours of DMA engines – Descriptor DMA and the iDMA. The processor cores and DMAs form the major chunk of data movers in the system. Traditional approaches of verification make the test generation process extremely complicated as one needs to manually add multiple synchronization points across the various C tests to maintain the sequential nature of the intended test scenario.

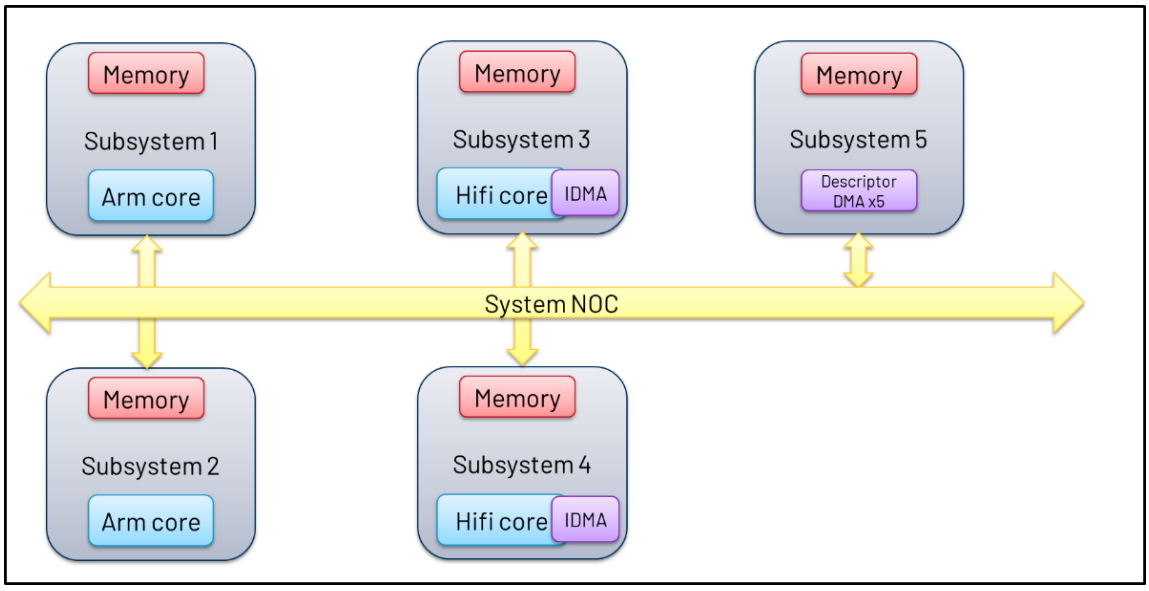


Figure 1:SOC architecture

Day-0 Testing through a library-based scenarios

System Methodology Library (SML) enables us to use CSV files to capture the processor and memory information of the system. This helps quick generation of memory access test scenarios exercising all processor cores and memories in the system through the system NOC. By accelerating the testing of multi-core memory access scenarios, we were able to catch few interconnect bugs in the system, early in the design cycle.

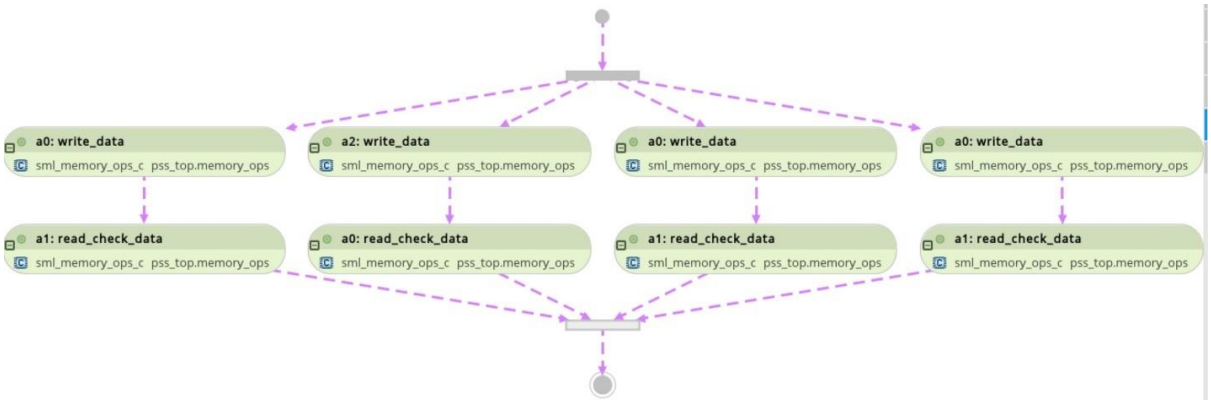


Figure 2: SML based multi-core memory access test

Descriptor based DMA engine

The DMA IP supports multiple modes of transfer and sizes. Creating a test suite for all possible combinations to verify at IP/subsystem level while also having the control to constraint randomize usecase based scenarios at system level was targeted through PSS. The scenario space of the descriptor based DMA IP is briefly shown below:

- Modes of transfer – Single mode, descriptor array, descriptor list, auto mode, trigger mode, descriptor on demand mode
- Transfer size – 1 to 32 bytes
- Data array dimensions – 1D and 2D, with configurable array indices and increments on source and destination memories.
- Interrupts support in PSS model

Self checkers

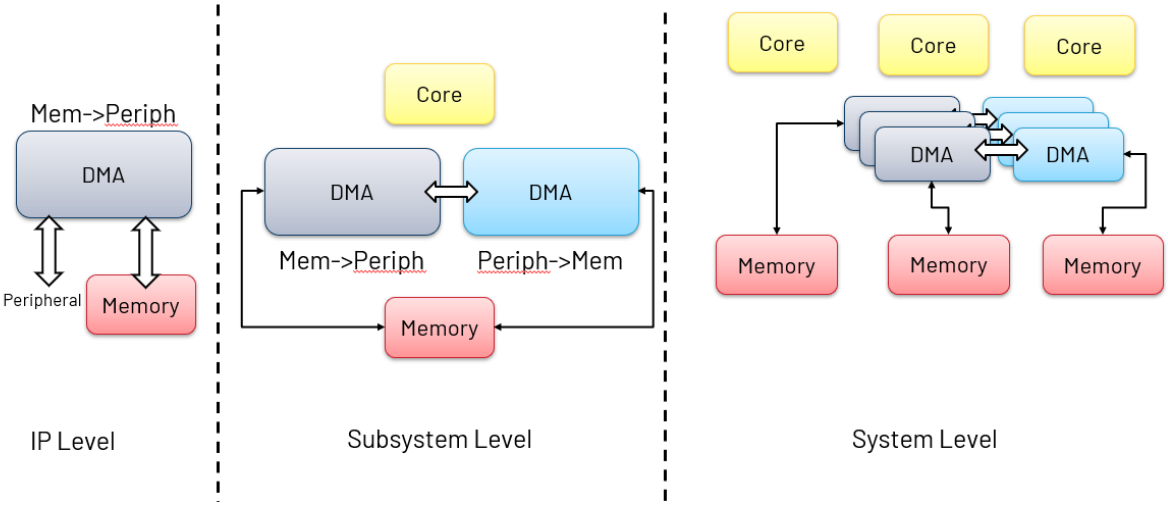
Since PSS is the “single source of truth” containing all the stimulus information, this was exploited to add self-checkers embedded in the PSS model to ensure the correctness of the test run. At the end of all DMA/memory transfers we add a self checker action which compares the expected data with the actual data in the memories, by reading the destination memories and provides a pass/fail result.

This is particularly helpful in offloading the scoreboarding activities and with that, the entire verification effort from traditional DV methods to completely PSS driven model. The complexities of varying transfer size, varying descriptor counts and their dimensions(1D,2D) and the varying indices of row and columns of both source and destinations were effectively modeled in PSS and ensured a robust checking mechanism.

Figure 3:Descriptor based DMA scenario with self-checkers

Sub-system to system level reuse

The PSS generated tests at IP and subsystem levels are ported to system level, enabling complete reuse of the test intent without duplication of effort.



System level scenarios

Shown below is a system level scenario that is generated through PSS which exercises 3 cores in the system and multiple DMA engines. This truly captures the practical advantages of PSS in creating multi-threaded scenarios and handling complex sequencing of actions through these cores and DMAs.

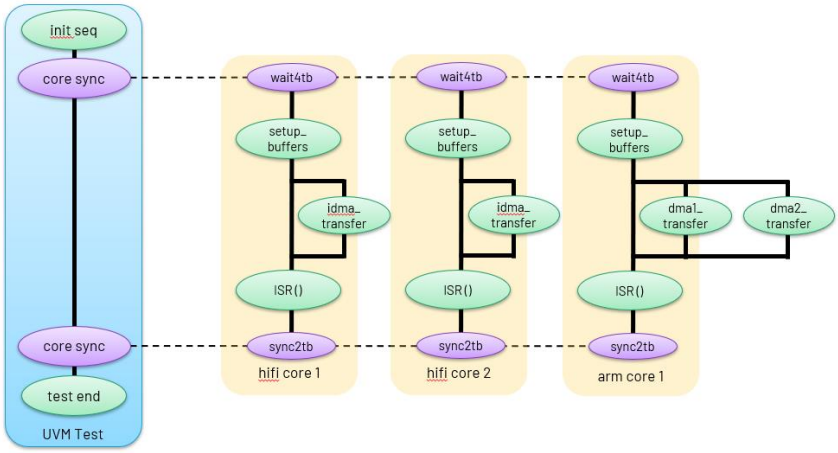


Figure 4: System level scenario

Coverage

PSS allows defining of covergroups and coverpoints similar to systemverilog and can be used to collect the pre-run coverage metrics, allowing the users the flexibility to either retain the tests or discard them based on the value they add to the overall coverage.
PSS enables coverage specification on use cases and on attributes of actions and flow objects.
The coverage metrics can be collected either in generation time or in runtime.

- Generation time coverage helps in regression planning and to prioritize the execution of tests on target platforms, even before running the tests.
- Run time coverage helps in validating the generation time coverage results, and to perform coverage analysis on scenarios and attributes that cannot be determined at generation time.

Test tables

For generating multiple configurations of a scenario, test tables offer significant help. Example would be to create multiple tests different of transfer size parameters, 1D and 2D array parameters, the PSS solver tool offers a robust CSV file input mechanism through which we can create multiple scenarios.

```
table:dma_reg_config
@package:dma_config_pkg, @size_const:NUM_DMA_DESC, @struct: dma_reg_config_s
#desc_inst,  #msize,  #psize, #xindex, #xincr, #yindex, #yincr
0,           1,       0,      5,       4,       0,       0
1,           2,       0,      8,       8,       2,       -3
2,           3,       0,      9,      16,       4,       -2
```

Figure 5: Configuration input CSV file

Results

- SOC PSS deployment yielded in identifying system level bugs faster.
  - System level scenarios with DMAs caught interconnect bugs early in the design cycle
  - Multiple RTL bugs(~15) caught with concurrent testing.

- Scenarios that are tough to create and randomize, done with ease.
    - Concurrent scenarios with all initiators(cores) and DMA channels active simultaneously.
    - Hybrid tests, combining PSS and non-PSS portions of stimulus being enabled.
    - Power aware scenarios being created. Results on this will be shared later.
- Complex memory architectures verified in PSS with highly configurable csv and control knobs.
- Vertical and horizontal(simulation, emulation) test intent reuse achieved.
- Streamlined DV with automated test-gen, coverage ranking, optimized regression list and run at realtime.
- 100% coverage achieved for PSS-focus areas.