ADAS IP Case Study: De-Coupled Generation and Simulation for Functional Verification

Rhys Hodson
Senior Verification Engineer
17 May 2022

Agenda

- Background and Motivation
- De-Coupled Generation and Simulation
  - Generation Test Bench
  - Simulation Test Bench
- Other Benefits
- Enhancements

Background and Motivation

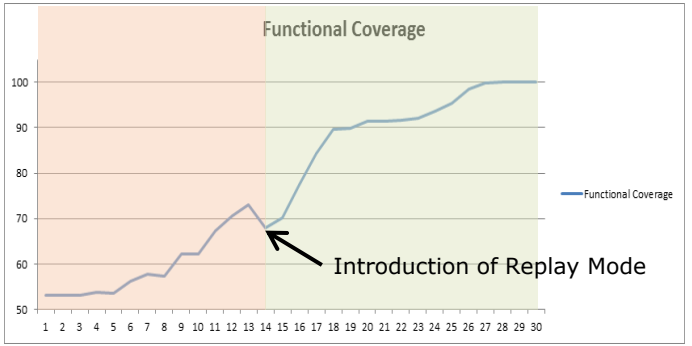
- First Gen ADAS IP Test Bench had a traditional coupled generation and simulation environment in Specman ‘e’

- ADAS IP is highly configurable – with complex interdependencies between configuration fields and other test input parameters
  - As test constraints evolved - Behaviour of tests in the Ranked Regression (test-seed pair) changed

- Enhancements for first and second gen Test Bench
  - Replay Mode with eRM
  - Followed by Decoupled Generation (e) and Simulation (SV)

Replay Mode

- The graph shows the functional coverage achieved by the ranked regression over the last 30 weeks of a project

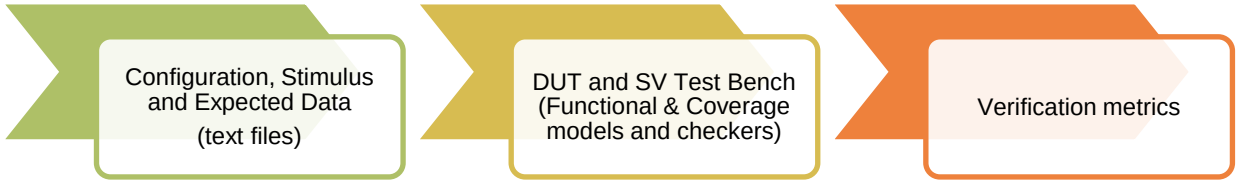


- The first 14 weeks of this 30 weeks period showed about a 20% rise in functional coverage

- After the introduction of the replay mode in week 14 we saw a sharp rise in coverage and coverage was closed off in the next 15 weeks

De-Coupled Generation and Simulation

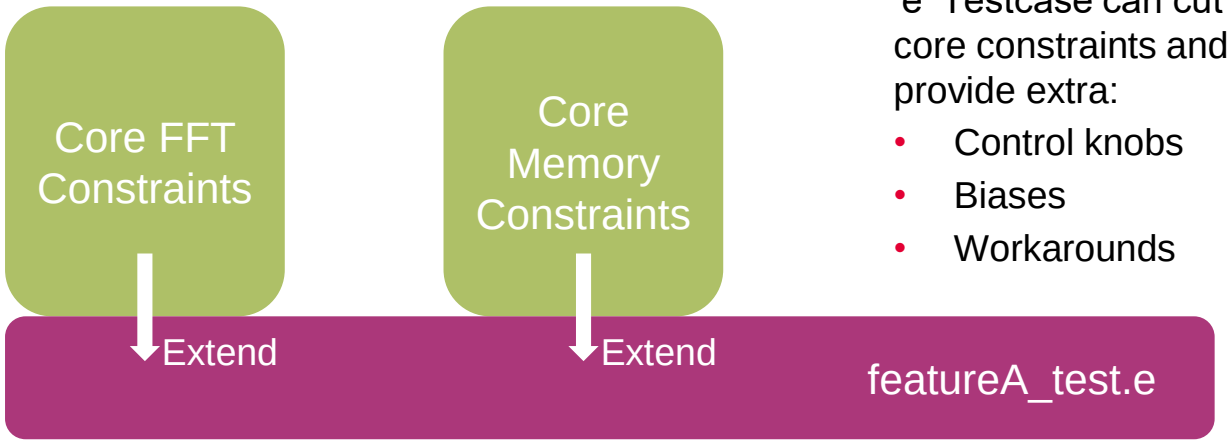
Generation Test Bench – Specman 'e'



Simulation Test Bench – System Verilog

Benefits of Specman ‘e’ for a Generation Testbench

- Write constraints involving multiple fields and resolve complex relationships among fields
  - leveraging the benefits of Aspect-Oriented Programming.



‘e’ Testcase can cut across core constraints and provide extra:

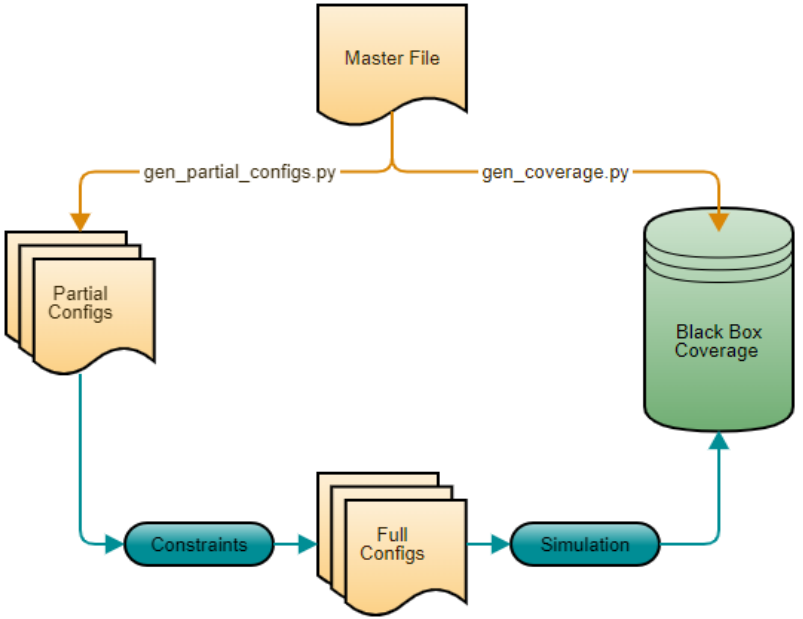
- Control knobs
- Biases
- Workarounds

Benefits of Specman ‘e’ for a Generation Testbench

- Debug tools (Generation Debugger) have stronger debugging capabilities compared to the ones offered for System Verilog.
- Specman Tables, reflection API and easy Python integration
  - Easy to read in constraints from a file with tables feature
  - Easy to output generated data in a variety of formats with scripting
- The configurations of ranking tests can be run through the generation constraints each time the test is run to check the configuration is still valid
- Easily switch between random generation of a new configuration and ‘re-generation’

Why use partial configurations alongside random tests?

- Partial configurations and black box coverage can be generated via scripting from a single source
  - Helps to close coverage quickly
  - Helps to qualify that the generation testbench is not over-constraining
  - Easy to soft constrain fields that are not of interest to their default values to make nice bring-up testcases

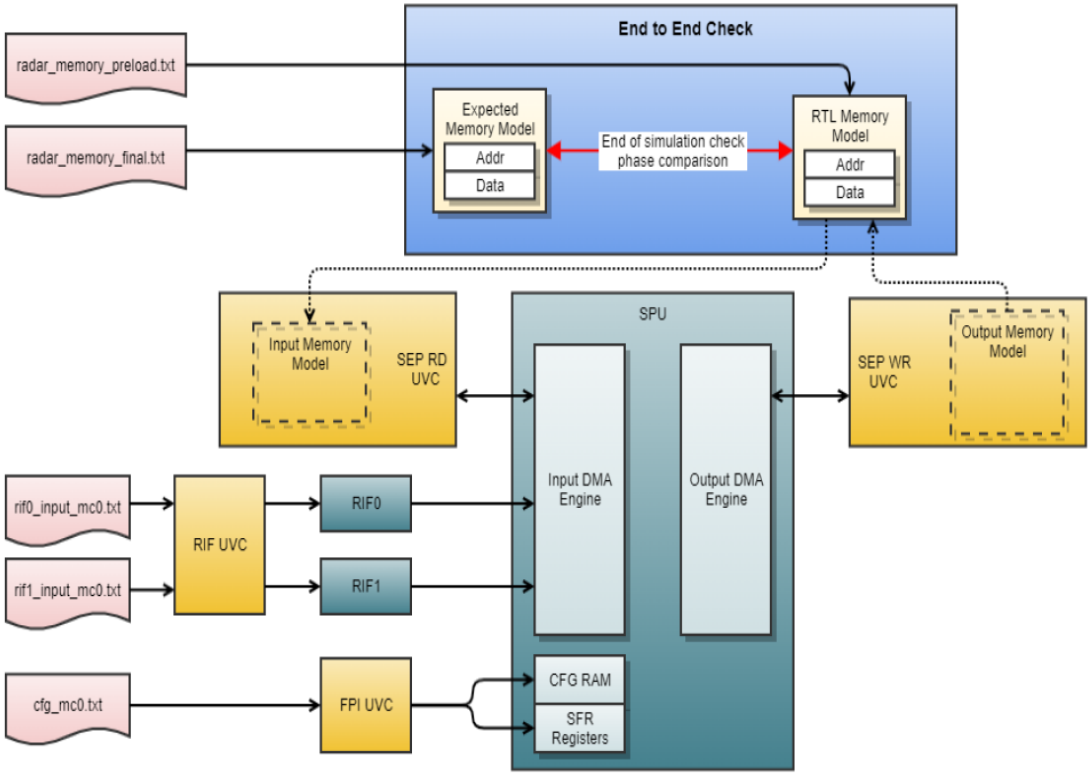


Benefits of a System Verilog Test Bench

- Helps Resourcing Verification Positions
  - Slightly easier to find SV resource
  - More Engineers are more open to work on SV rather than ‘e’
- No tie in to a particular simulator or vendor
- The option to use Emulator tools if you are willing to adapt testbench components

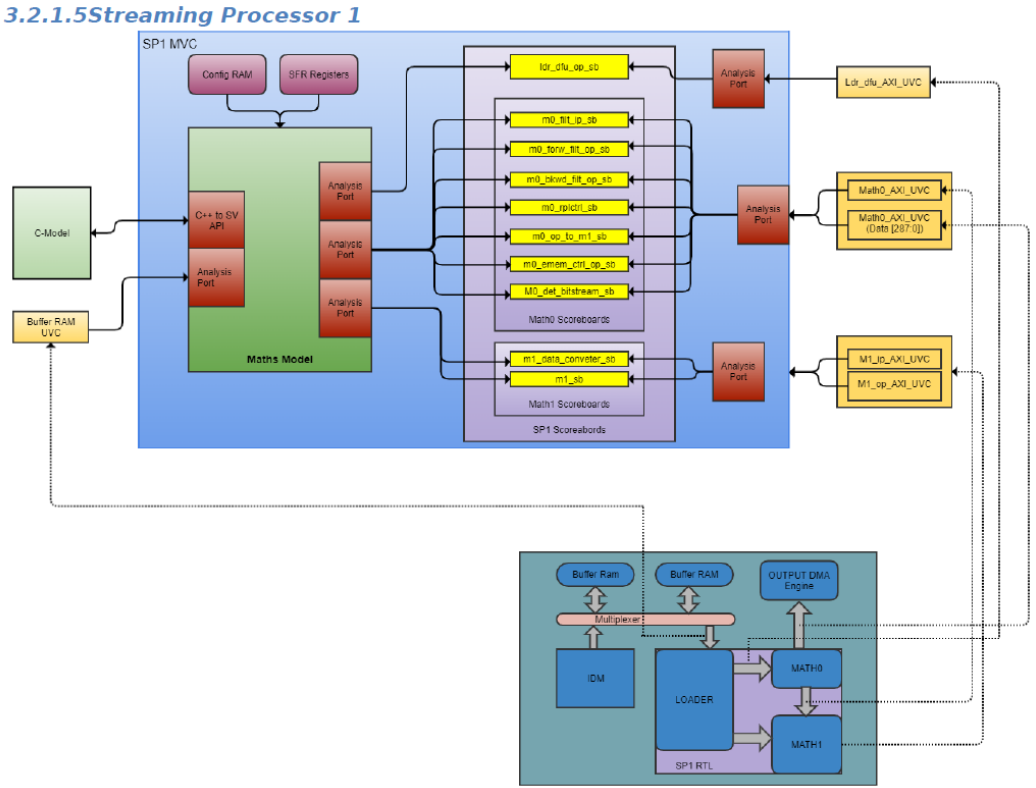
Simulation Test Bench

- Typical SV UVM Test Bench
- Subtle differences :
  - Register Configuration sequences use text files delivered by Generation Test Bench
  - Stimulus Driver and End to end check use data delivered from Generation Test Bench



Simulation Test Bench - continued

- Module level Checks :
  - Uses typical UVM approach along with C-functions to generated expected data for modules.



Other Benefits

- Generation is Cheap
  - Can afford to generate 1000s of configurations before deciding to simulate a handful of these configurations.
- Group tests based on Generated configurations and schedule simulations efficiently.
  - Longer tests do not run weekday nightly regressions.
  - Set timeout based on predicted cycles.
- Storing Tests as configurations in an SQL Database
  - Porting of tests between derivatives
  - Porting of tests between register changes
- Bugs found can be more reliably reproduced on different versions of a testbench

Enhancements

- Workflows that use multiple different tools and flows are… slow. But it can be argued that the time is certainly made up elsewhere

- Configurable Generation and Simulation Test Bench for Derivatives
  - Increase reuse of tests, verification models and checkers across different Families of Radar DUTs.

Q & A

Part of your life. Part of tomorrow.