



The Test Bench Factory: Building Verification Environments Faster, Better, Smarter

Thorsten Dworzak, Dr. Johannes Grinschgl
Infineon Technologies AG, Neubiberg, Germany



Agenda

1

State-of-the-Art

2

Motivation

3

VIP Reuse

4

The TB Architecture

5

Stimulus, Checking, Coverage

State-of-the-Art

- Industry uses verification code generation for
 - Configuration and Status Register (CSR) Modelling
 - Structural test benches (TBs)
 - Formal property generation
 - Very specific tasks (see DVCon papers)
- OpenSource and commercial solutions available for SystemVerilog
 - E.g. UVMF: Universal Verification Methodology Framework

Motivation

- Several in-house solutions exist that are either
 - Too simplistic (e.g, DVT Eclipse templates)
 - Using Specman/e (being phased out)
 - Tied to a specific DUV architecture
- Introducing TBGen: a universal UVM-SV test bench generator that
 - Increases productivity
 - Is suitable for module and system-level
 - Facilitates Verification IP (VIP) and TB re-use
 - Provides a standard TB architecture based on SV libraries

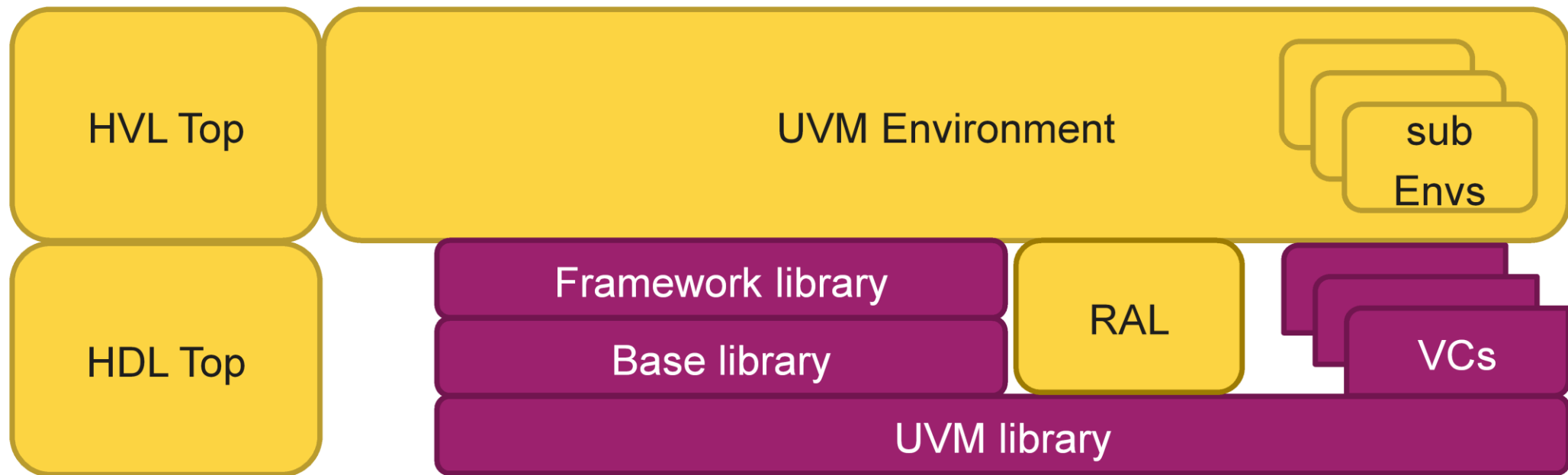
VIP Reuse

- Each verification component (VC) is associated with meta-data
- This meta-data contains all the information needed to instantiate the VC and for generating a VC (e.g. TB)
 - Interfaces
 - Sequencers
 - Configuration
 - Environments
 - VCs which should be instantiated
 - Register model instantiations
- All reusable VCs are shared company wide over a central repository.

Taxonomy of VIPs

Type of VC	Description
iVC	an interface VC drives and monitors connection to a DUV via SV interfaces; the iVC needs minimal information from the user to control the UVM-VIP-generator, because most information is implicitly inferred from the generator (standard UVM architecture)
iVCg	same as iVC but the "g" denotes a 3rd party iVC that does not follow the structure of generated VCs; it contains explicit information about its sequencers, analysis ports etc.
mVC	typically containing an agent with a scoreboard; does not connect to the DUV (no SV interfaces) but receives items from iVCs on TLM1 analysis ports
sysVC	the integration layer environment; typically containing instances of iVC(g)s, mVCs, other sysVCs, and register components
TB	this is similar to a sysVC but also contains tests, HDL testbench modules etc.

The TB Architecture



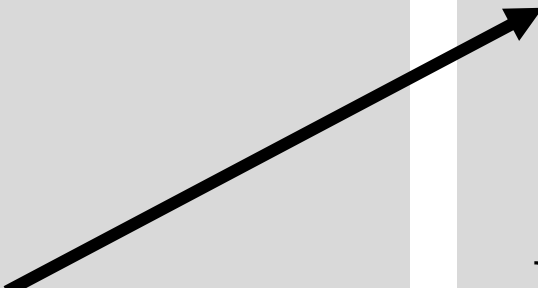
High-level Description of TB Topology

- Global information (DUV name, UVM environment name, etc)
- File path to imported VC meta-data
- Instantiation of VCs (interface parameters, number of agents, etc.)
- Topology of a bus-fabric (components, buses, bus interfaces, address-maps, routing information, etc.)
- CSR model generation attributes (e.g. path to specification)
- CSR address maps and connection to VC analysis ports, adapter/predictor types, etc.

TTD Input in HOCON Format

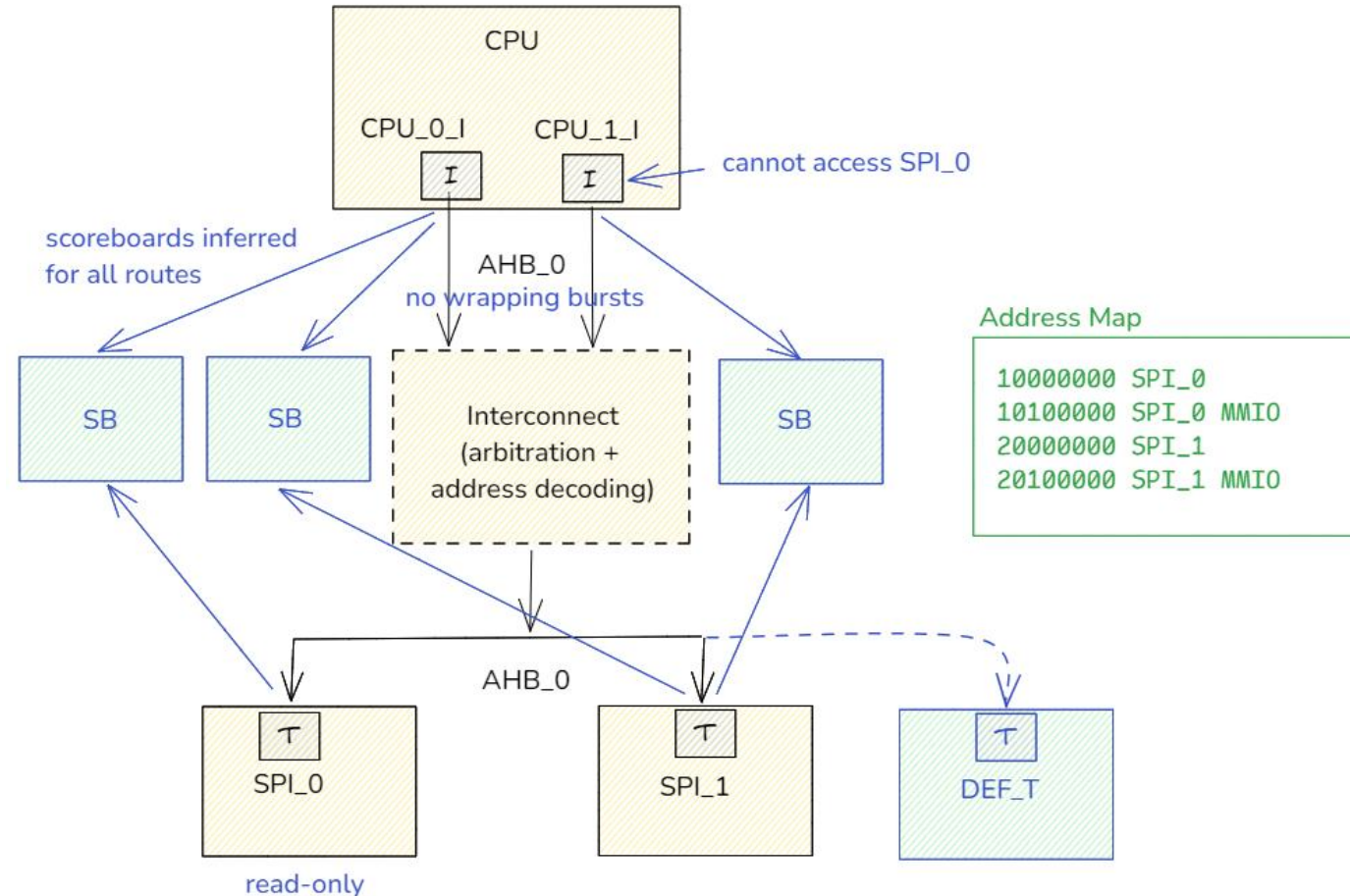
```
global {  
  project: mac_env  
  dut: mac_fe  
  dut_instance: dut_i  
  ...  
}  
  
vc_list: [  
  /home/vips/ahb/meta/ahb_ivc.json  
  /home/vips/libraries/meta/ifx_uvm.json  
  ...  
]
```

```
uvcs {  
  ahb_0 {  
    name      : ahb_ivc  
    hdl_bb    : true  
    bind_to   : dut_i  
    harness   : ahb_ivc_harness  
    embed     : false  
  }  
  scoreboard_0 {  
    name              : ahb_sb  
    embed             : false  
    generate           : true  
    default_conn_schema : all  
  }  
}
```



Stimulus, Checking, Coverage

- A bus-fabric topology has
 - components (peripherals, bridges, ...), buses, bus-interfaces
 - attributes like routing, address maps and supported protocol features
- TBGen supports
 - A sequence framework
 - Automatic scoreboard inference
 - Functional coverage collection on checked scoreboard items



Results

DUV	No. of different iVC/mVC types/instances	No. of different sysVC types/instances	Lines of code for top-level test bench (generated)
SPI memory controller (IP)	8/9	-	18900
CPU with interconnect, DMA, memory controllers	7/22	6/6	144k (120k of which for registers)
System Resources sub-system (clock, reset, power etc.)	49/529	18/18	220k (146k of which for registers)

Questions?