# Using Synopsys VCS to connect a Company's SystemC Verification Methodology to Standard Concepts of UVM

Dipl.-Inform. Frank Poppen

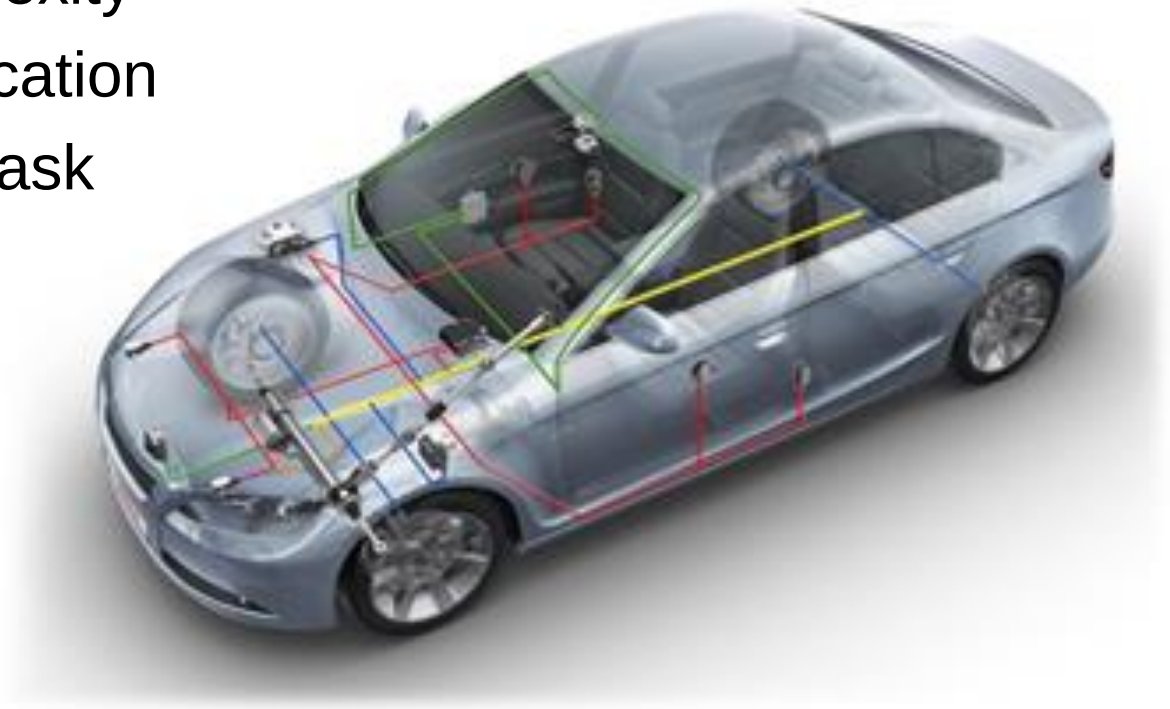
OFFIS Institute For Information Technology

June 25, 2015

SNUG Germany

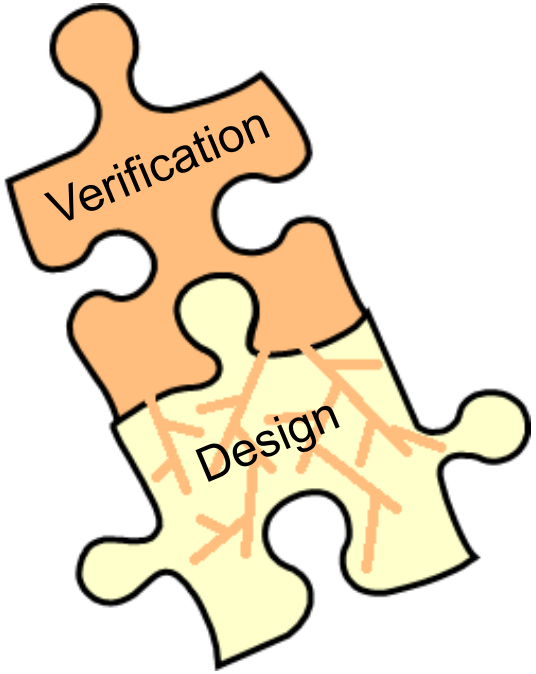
# The Challenge

- electronics in heterogeneous systems
- ambient and safety relevant
- increasing complexity
- design and verification
- lining up for the task
  - tailored solutions
  - standards
  - languages
  - tools



# No „One Size Fits All“

- verification engineers choose and combine what …
  - fits best for the company
  - the design-team
  - the application domain
  - (budget, roadmap, …)
- deep roots in the design process
- changes endanger productivity
- change carefully and incrementally



# Agenda

Motivation

Verification Islands

Matching Concepts of UVM and IFS

Bus Arbiter Test Case

UVMC can do / cannot do

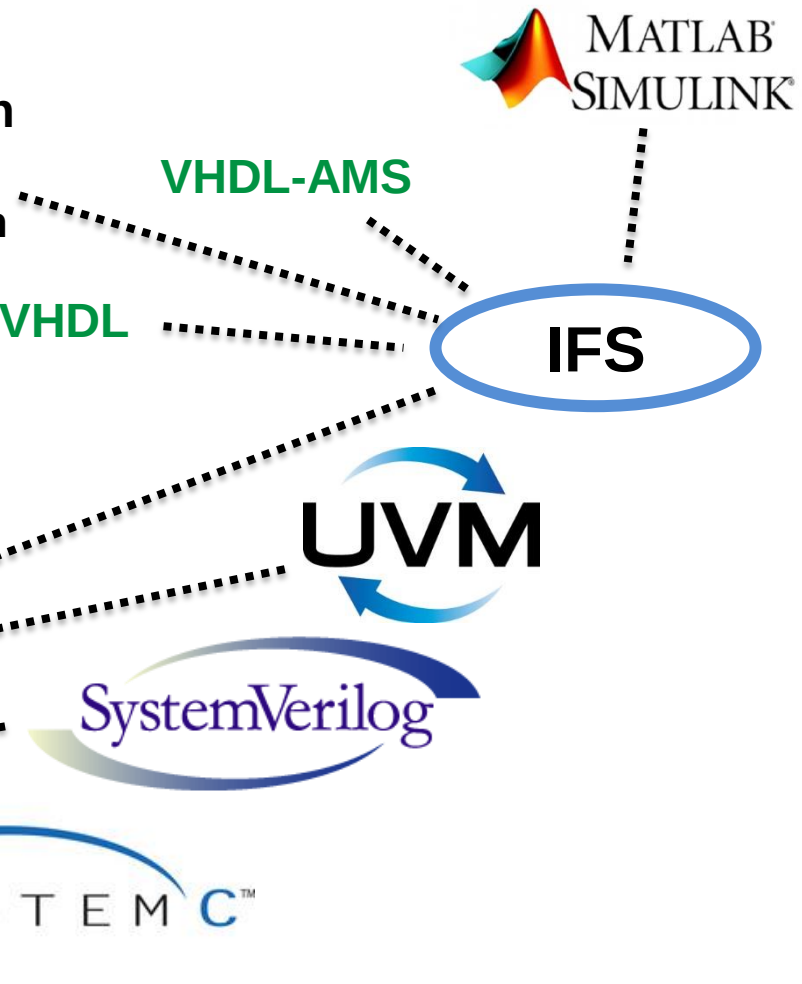
Architecture of the Experiment

Conclusions

# Bring it All Together

**Integrated Functional Verification Script Environment**

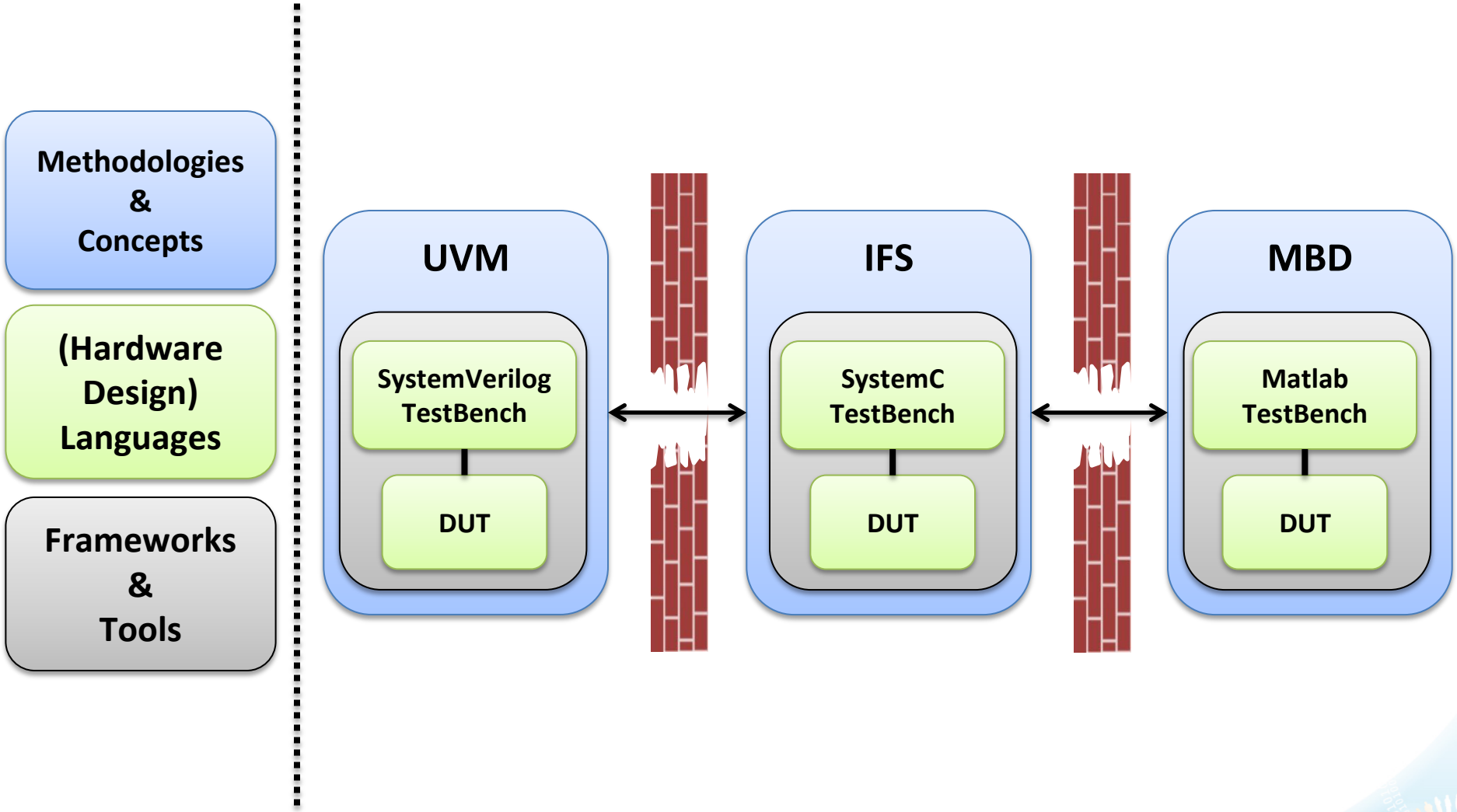
- **separate testcase from testbench**
- **reuse of testbench IP**




- lining up for the task
  - tailored solutions
  - standards
  - languages
  - tools

# Verification Islands

Synopsys Users Group

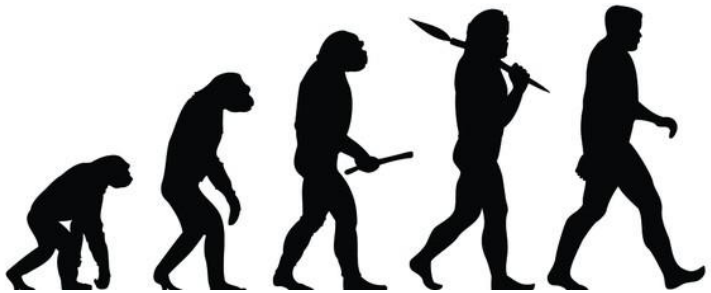


Synopsys Users Group

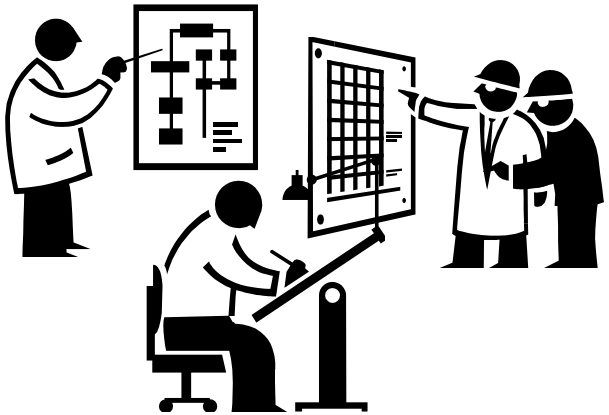
# Matching Concepts of UVM and IFS

# IFS long before SystemVerilog

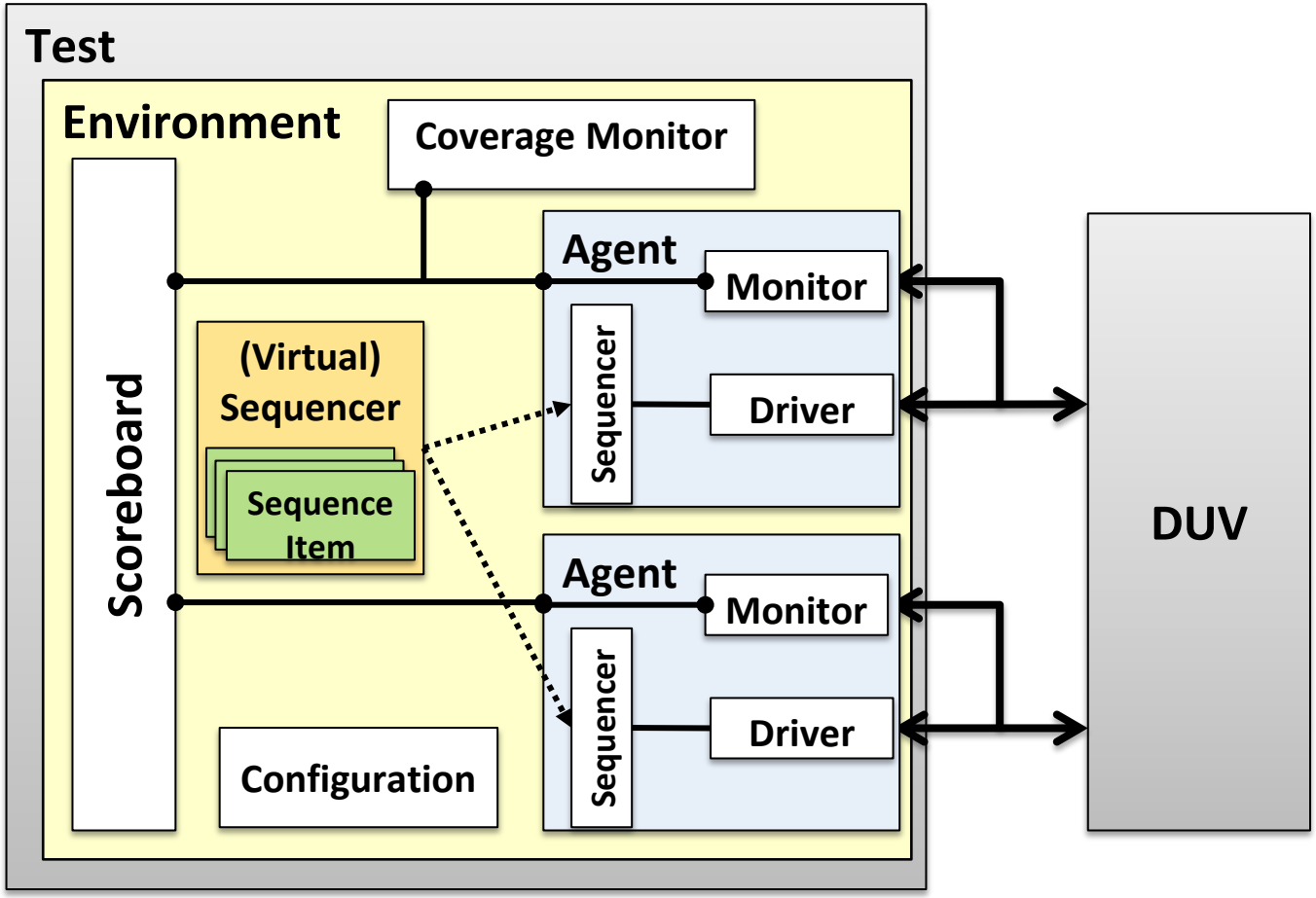
- enhanced from VHDL with …
  - VHDL-AMS
  - SystemC
  - Matlab/Simulink
  - (and now SystemVerilog and UVM)



- SystemC based library simulates with any simulator (IEEE 1666)
- tailored to relevant use scenarios in special contexts
- simple IFS command language for (self-checking) test cases
  - digital designer
  - analog designer
  - verification engineer
  - system engineer
  - software engineer



# Matching Concepts of UVM and IFS




**UVM Environment**

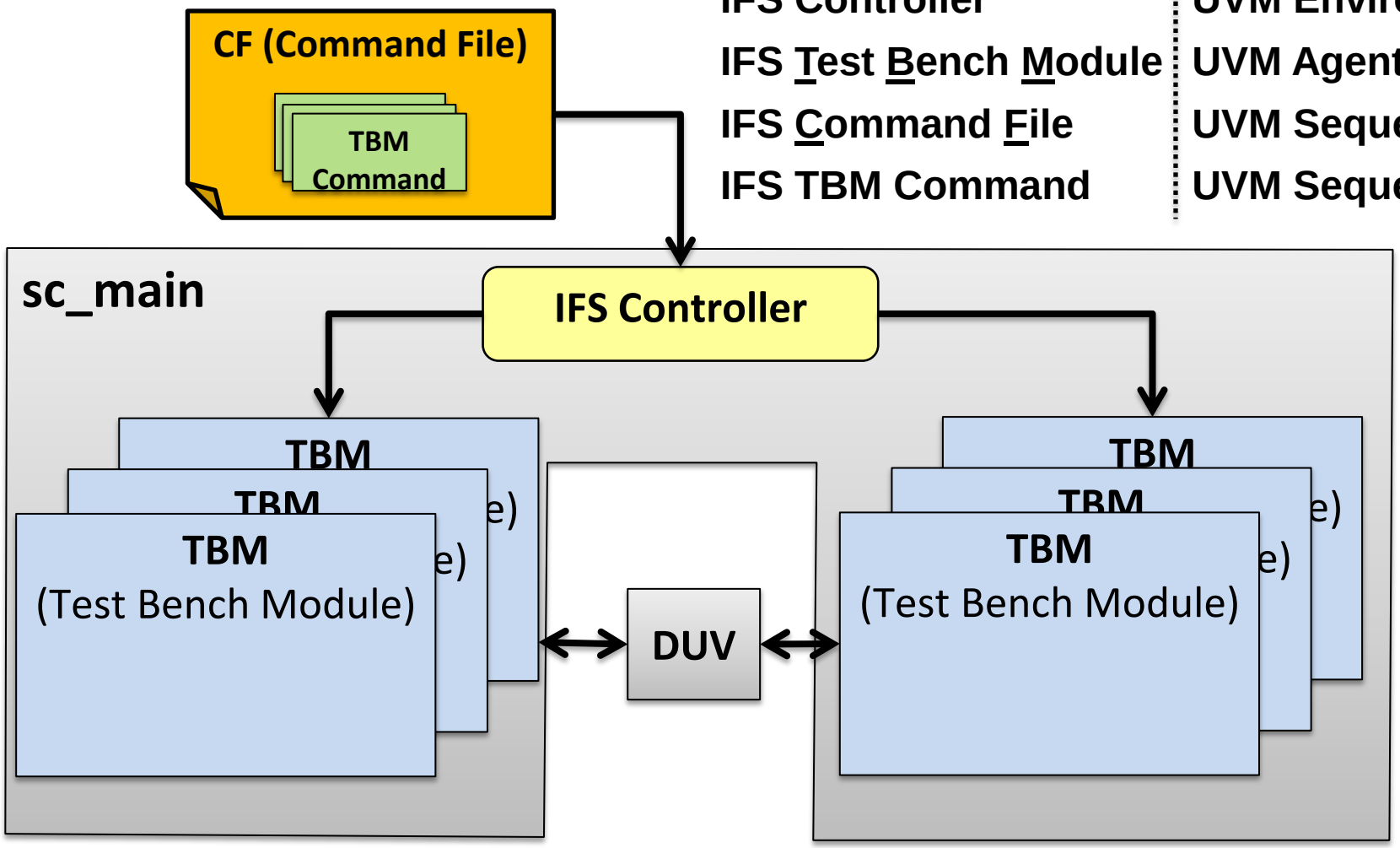
**UVM Agent**

**UVM Sequence**

**UVM Sequence Item**

# Matching Concepts of UVM and IFS

| | |
|---|---|
| IFS Controller | UVM Environment |
| IFS Test Bench Module | UVM Agent |
| IFS Command File | UVM Sequence |
| IFS TBM Command | UVM Sequence Item |

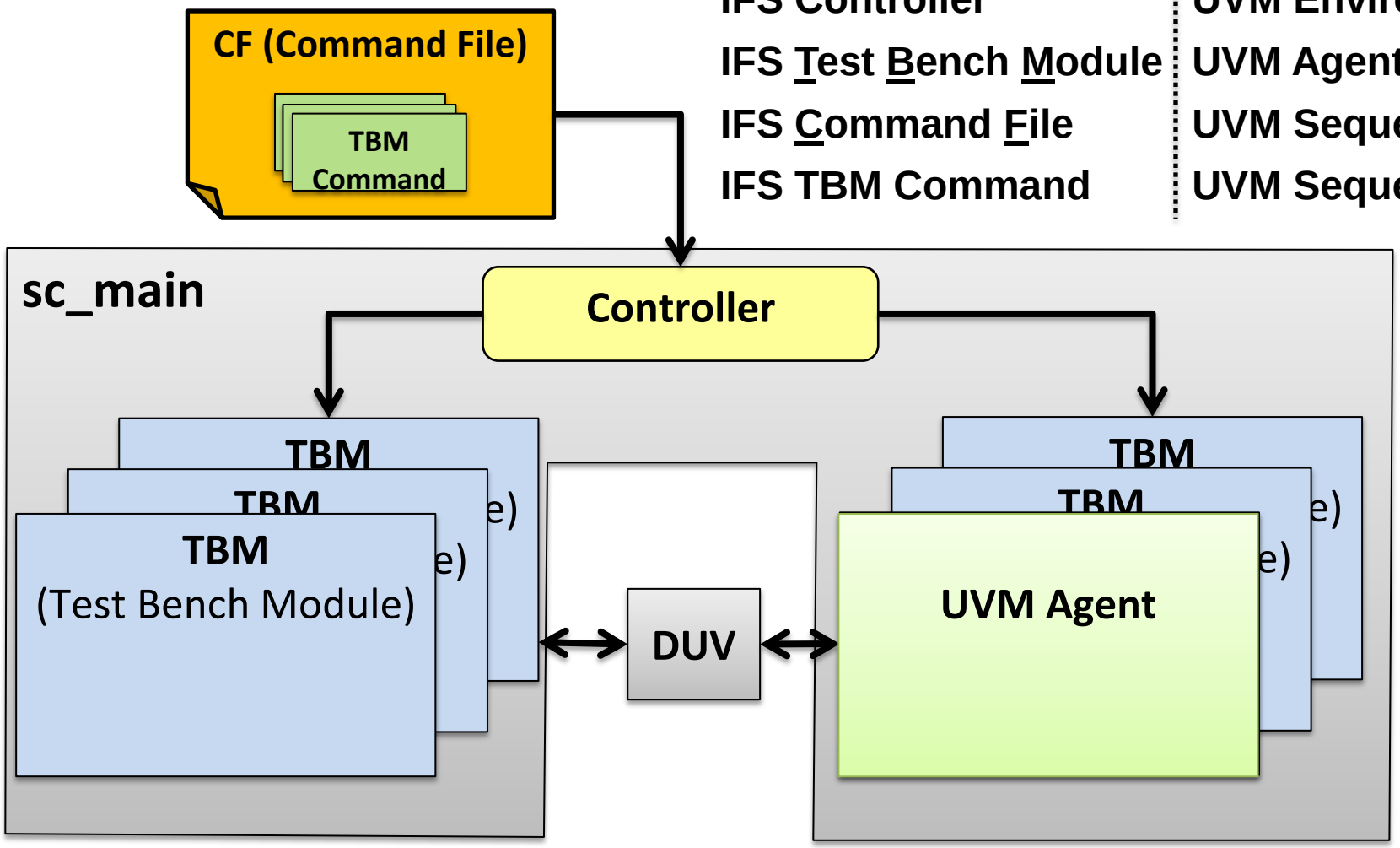


# Matching Concepts of UVM and IFS

- TBM directly correlates to UVM agent
- transactor between test bench and DUV
- translate messages/commands to bit wiggles
- Objective: reuse of UVM agents delivered with UVM test environment / verification IP
- UVM agent can be used inside SystemC

# Matching Concepts of UVM and IFS

| IFS | UVM |
|---|---|
| IFS Controller | UVM Environment |
| IFS Test Bench Module | UVM Agent |
| IFS Command File | UVM Sequence |
| IFS TBM Command | UVM Sequence Item |



# Matching Concepts of UVM and IFS

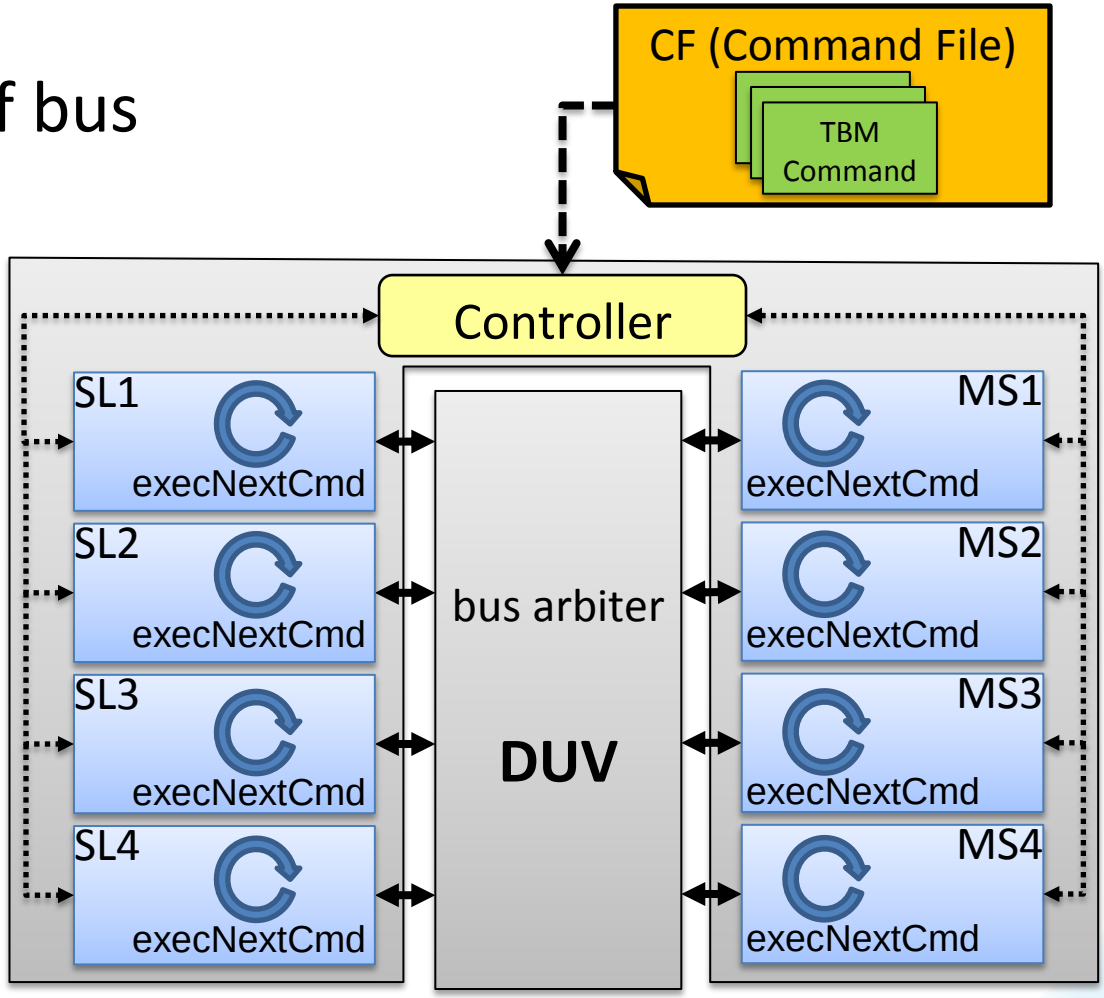
- TBM directly correlates to UVM agent
- transactor between test bench and DUV
- translate messages/commands to bit wiggles
- Objective: reuse of UVM agents delivered with UVM test environment / verification IP
- UVM agent can be used inside SystemC

- <u>connects any SystemC test bench to SV UVM …</u>
  … <u>does not rely on the IFS library!</u>

# Bus Arbiter Test Case

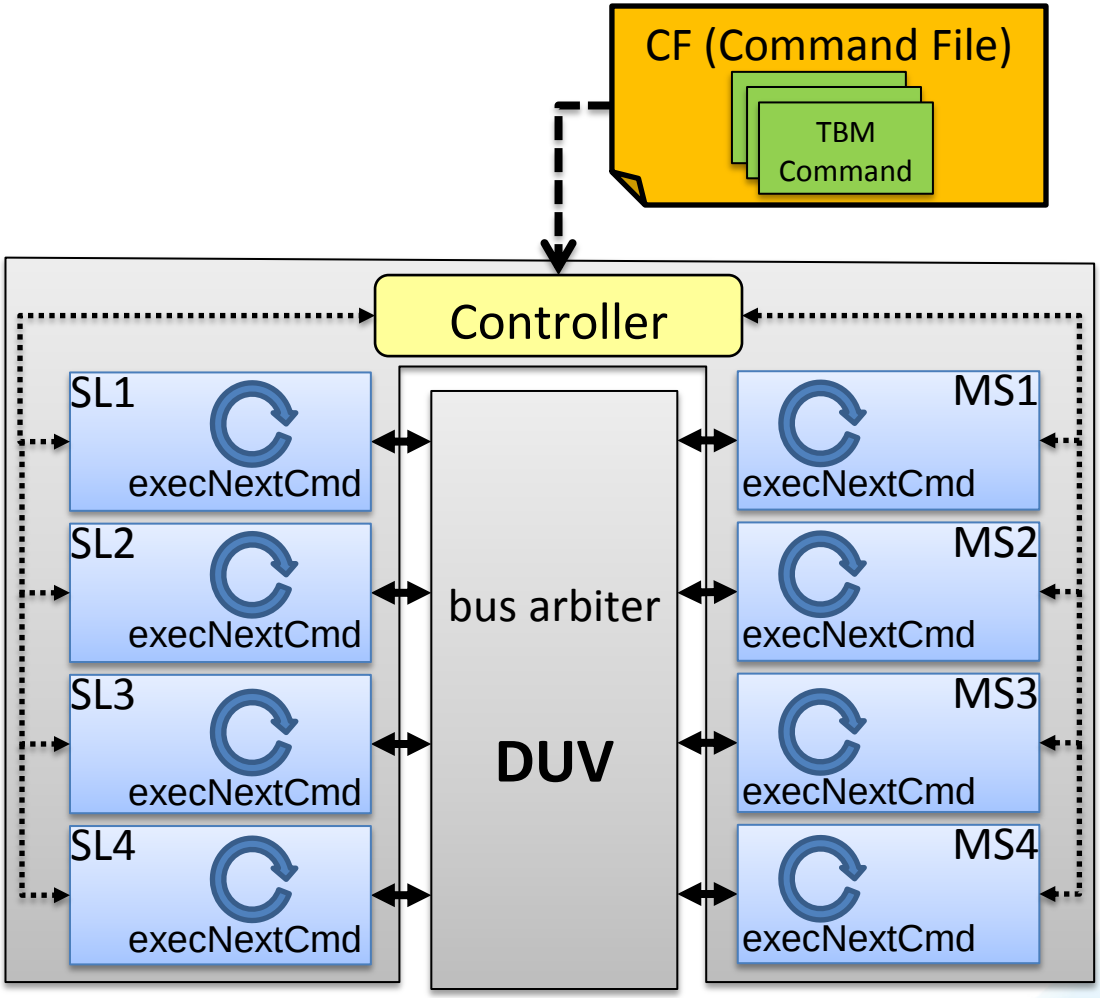
# Simple Switched Bus Arbiter

- configurable number of bus masters and slaves
- suited for experiment
  - simple to understand
  - easy to verify
- used for several mixed-language simulations evaluations
- VHDL, Verilog and SystemC



# Simple Switched Bus Arbiter

```
CLK PERIOD 10 ns
CLK RESET  0  12

ALL SYNC ALL

#loop 4
  ALL SYNC ALL
  MS1 Write $(100*#i) $(100+#i)
  MS1 Read  $(100*#i) $(100+#i)
#eol

ALL SYNC ALL

SL1 print "End Of Test Script"

ALL QUIT
```

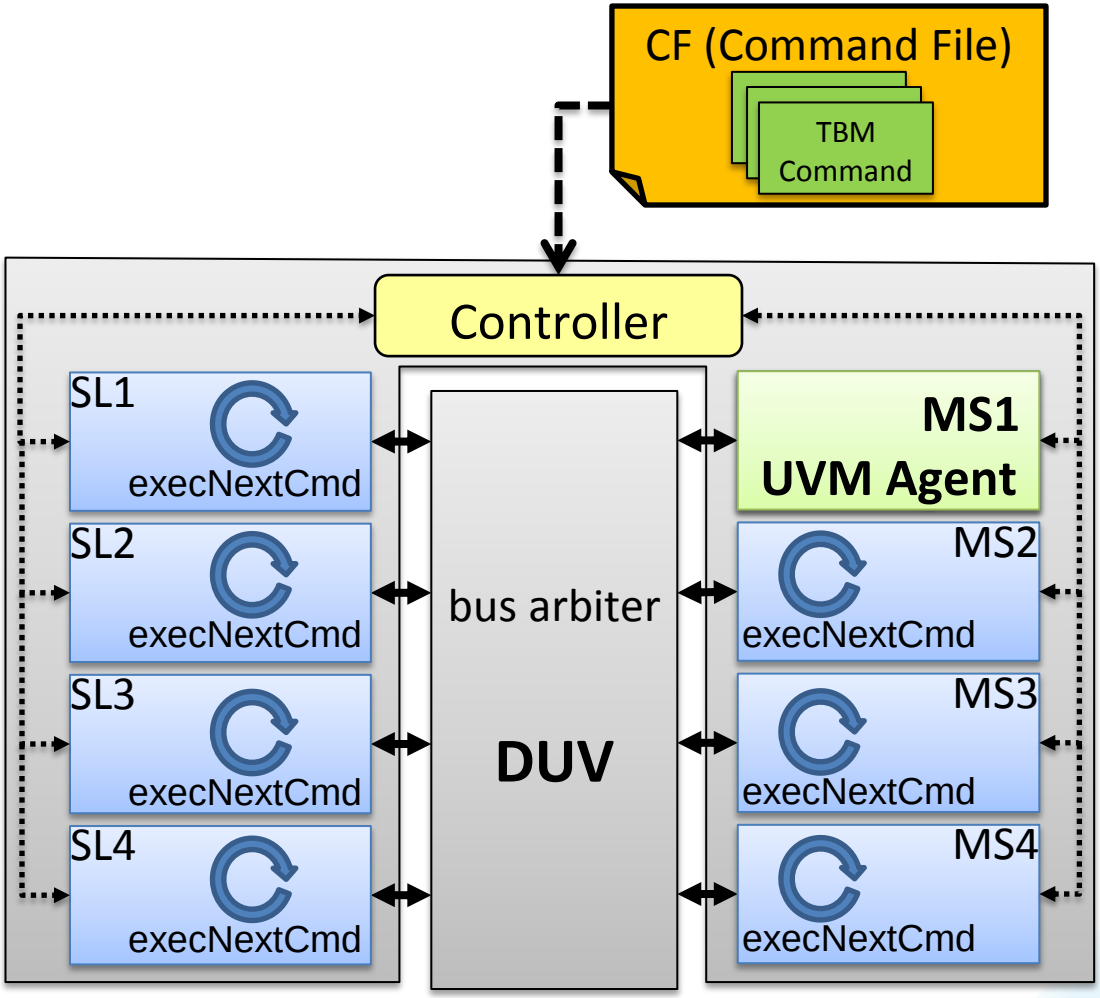


# Simple Switched Bus Arbiter

```
SystemC 2.3.0-ASI --- Nov 29 2013 14:57:17
 Copyright (c) 1996-2012 by all Contributors,
 ALL RIGHTS RESERVED
INFO (0 s) Loading script: 'control.cmd'
```

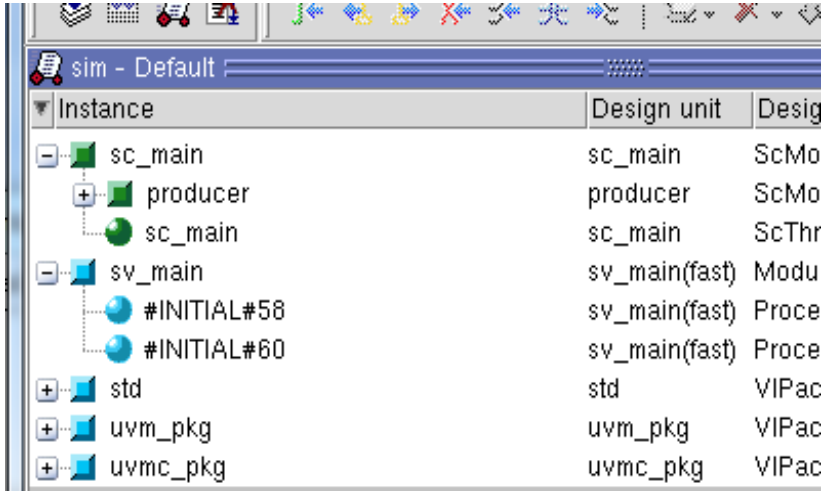
...

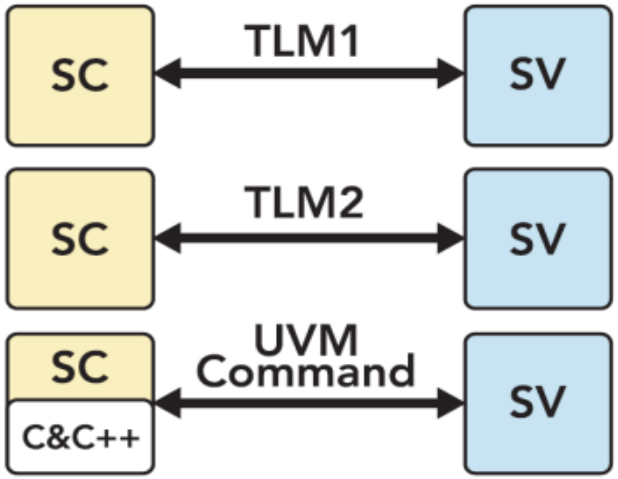
```
[SL3,350 ns]    Read    (Address: 100, Value: 101)
[MS1,360 ns]    Read    (Address: 100, Value: 101)
[MS1,410 ns]    Write   (Address: 200, Value: 102)
[SL2,420 ns]    Write   (Address: 200, Value: 102)
[SL2,490 ns]    Read    (Address: 200, Value: 102)
[MS1,500 ns]    Read    (Address: 200, Value: 102)
[MS1,550 ns]    Write   (Address: 300, Value: 103)
[SL1,560 ns]    Write   (Address: 300, Value: 103)
[SL1,630 ns]    Read    (Address: 300, Value: 103)
[MS1,640 ns]    Read    (Address: 300, Value: 103)
INFO (640 ns)[SL1] End Of Test Script
INFO (640 ns) SIMULATION END FROM COMMAND
FILE
INFO (640 ns) Exiting simulation.
Info: /OSCI/SystemC: Simulation stopped by user.
INFO (640 ns) Report:
INFO (640 ns) Encountered errors: 0
INFO (640 ns) Encountered warnings: 0
```



# UVMC can do / cannot do

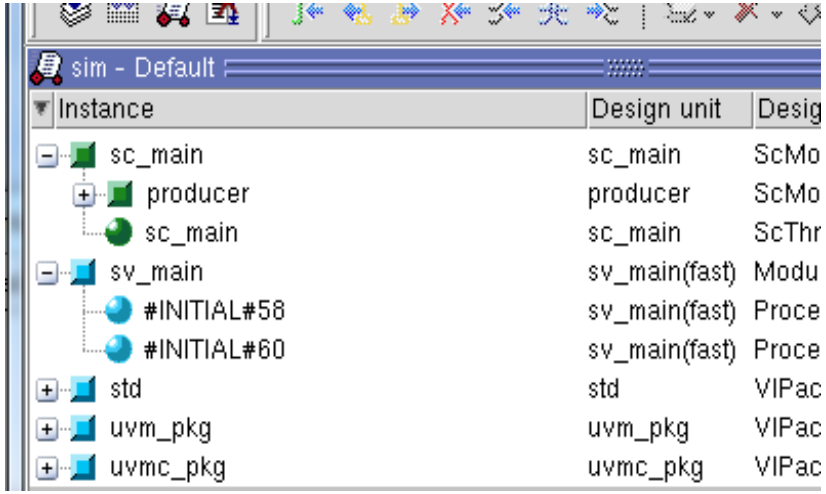
# UVM Connect (Verification Academy)

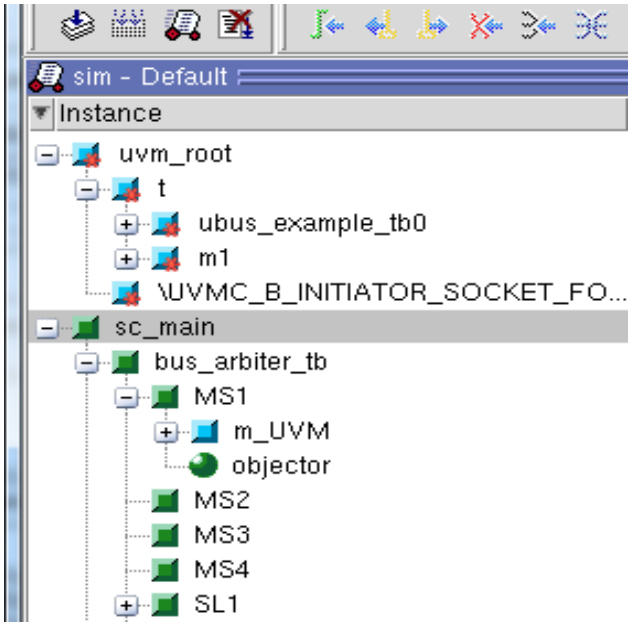



- interconnect UVM and SystemC
- UVMC is an open-source UVM/OVM-based library
- uses System Verilog Direct Programming Interface
- good for TLM messages and control commands
- less good for RT level signal interfacing

# UVM Connect (Verification Academy)

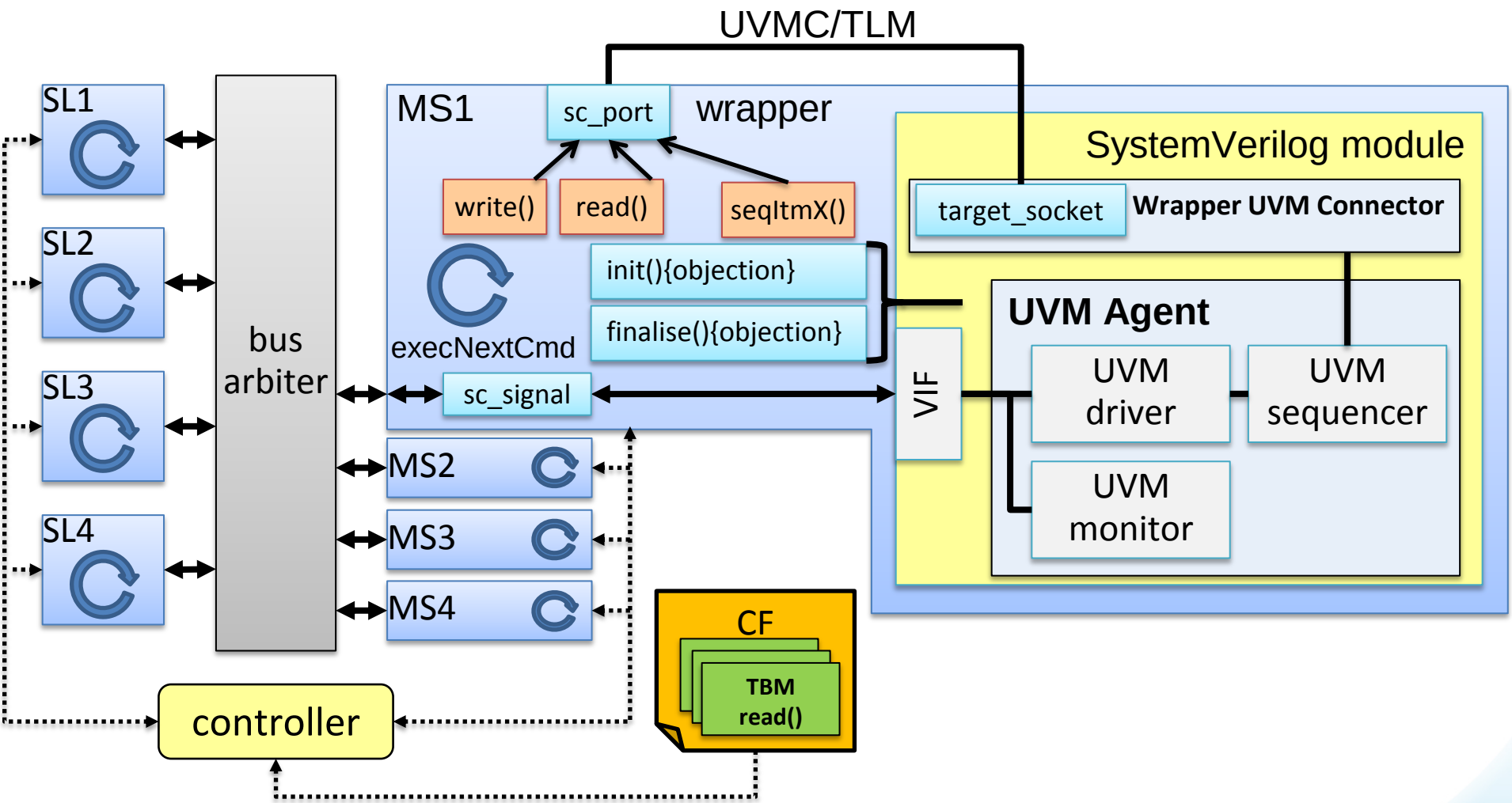



- interconnect UVM and SystemC
- UVMC is an open-source UVM/OVM-based library
- uses System Verilog Direct Programming Interface
- good for TLM messages and control commands
- less good for RT level signal interfacing
  => instantiation through foreign language module

# Architecture of the Experiment

# Architecture of the Experiment



# Running the Experiment

```
1:   *********** wrap_uvm_master::CTOR(): Connecting TLM port
2:   Connecting an SC-side proxy chan for 'bus_arbiter_tb.MS1.port_10' with
     lookup string 'sc_wrap_MS1' for later connection with SV
3:   ************* uvm_test::CTOR(): instantiating UVMC test
4:   INFO (0 s)[bus_arbiter_tb.MS1] Registered module 'bus_arbiter_tb.MS1'
5:   ...(also MS2 to MS4)
6:   INFO (0 s)[bus_arbiter_tb.SL1] Registered module 'bus_arbiter_tb.SL1'
7:   ...(also SL2 to SL4)
8:   INFO (0 s)[bus_arbiter_tb.CLK] Registered module 'bus_arbiter_tb.CLK'
9:   INFO (0 s) Loading script: 'control.cmd'
10:  INFO (0 s) Finished loading
```

- SystemC elaboration phase
- SystemC part of UVMC opens port
- IFS registers test bench modules
- loading command file

# Running the Experiment

```
11:  Chronologic VCS simulator copyright 1991-2014
12:  Contains Synopsys proprietary information.
13:  Compiler version J-2014.12-1; Runtime version J-2014.12-1;  Feb 20 11:15 2015
14:  ----------------------------------------------------------------
15:  UVM-1.1d.Synopsys
16:  (C) 2007-2013 Mentor Graphics Corporation
17:  (C) 2007-2013 Cadence Design Systems, Inc.
18:  (C) 2006-2013 Synopsys, Inc.
19:  (C) 2011-2013 Cypress Semiconductor Corp.
20:  ----------------------------------------------------------------
21:  ************ wrap_uvm_master::initialiseModule(): Raising objection for UVM phase 'run'!
22:  ----------------------------------------------------------------
23:  UVMC-2.2
24:  (C) 2009-2012 Mentor Graphics Corporation
25:  ----------------------------------------------------------------
26:  Registering SV-side 'sc_wrap_MS1.ifs_monitor.in' and
     lookup string 'sc_wrap_MS1' for later connection with SC
27:  UVM_INFO @ 0 ns: reporter [RNTST] Running test ...
```

- UVM phasing starts after SystemC elaboration
- SystemC raising run phase objection
- UVMC starts
- System Verilog  part of UVMC opens port

# Running the Experiment

```
28:  Connected SC-side 'bus_arbiter_tb.MS1.port_10' to SV-side 'sc_wrap_MS1.ifs_monitor.in'
29:  UVM_INFO ../tb_uvm/ubus_example_master_seq_pkg.sv(134) @ 0 ns: sc_wrap_MS1.sequencer@@master_memory_seq
     [master_memory_seq] master_memory_seq starting...
30:  [bus_arbiter_tb.CLK,100 ns] reset gets passiv
31:  [bus_arbiter_tb.SL1,100 ns] set slave_offset = 300
32:  ... (configuring SL1 to SL4)
33:  UVM_INFO ../tb_uvm/ubus_master_driver_pkg.sv(89) @ 110 ns: sc_wrap_MS1.driver [ubus_master_driver] *********
     ubus_master_driver::get_and_drive(): Waiting for Item on seq_item_port!
34:  ************* wrap_uvm_master::Write(): Sending payload {cmd:2 parameters:100 98} to MS1.socket at time 150
35:  UVM_INFO ../tb_uvm/ifs_command_monitor_pkg.sv(75) @ 150 ns: sc_wrap_MS1.ifs_monitor [ifs_command_monitor] *********
     ifs_command_monitor::b_transport(): SC-TLM communication received: cmd - 00000002, parameters - '{"100", "98"}
36:  UVM_INFO ../tb_uvm/ifs_command_monitor_pkg.sv(116) @ 150 ns: sc_wrap_MS1.ifs_monitor [ifs_command_monitor] *********
     ifs_command_monitor::peek(): Informing driver to drive cmd- 2, addr- 100, data- 98!
37:  UVM_INFO ../tb_uvm/ubus_master_driver_pkg.sv(91) @ 150 ns: sc_wrap_MS1.driver [ubus_master_driver] *********
     ubus_master_driver::get_and_drive(): Received Item on seq_item_port!
38:  [bus_arbiter_tb.SL3,190 ns] Write (Address: 100, Value: 98)
39:  UVM_INFO ../tb_uvm/ubus_master_monitor_pkg.sv(173) @ 195 ns: sc_wrap_MS1.monitor [ubus_master_monitor]
     collect_data_phase.
```

- both ports SC and SV get connected
- SystemC sending payload
- SystemVerilog receiving payload
- UVM agent (driver) receiving sequence item

# Running the Experiment

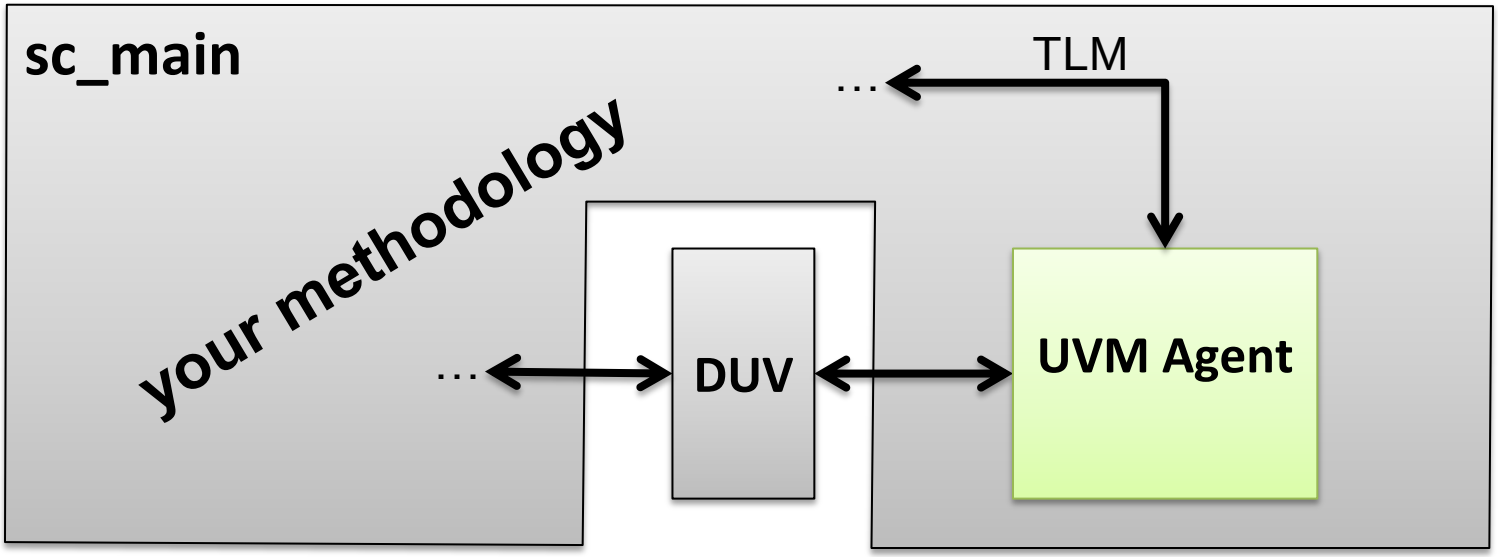
```
INFO (2050 ns)[bus_arbiter_tb.SL1] ----------------------------------------
INFO (2050 ns)[bus_arbiter_tb.SL1] End Of Test Script Reached...
INFO (2050 ns)[bus_arbiter_tb.SL1] ----------------------------------------
INFO (2050 ns) Exiting simulation.
SystemC: simulation stopped by user.
************* wrap_uvm_master::finaliseModule(): Dropping objection for UVM phase 'run'!
        V C S   S i m u l a t i o n   R e p o r t
Time: 2050000 ps
```

- script reached ‘quit’ command
- SystemC wrapper drops UVM objection
- simulation ends

# Conclusions

# Conclusion

- evolution of tailored verification methods to UVM
- UVM agents can be controlled in SystemC test environments
- approach is generic: IFS library is not mandatory

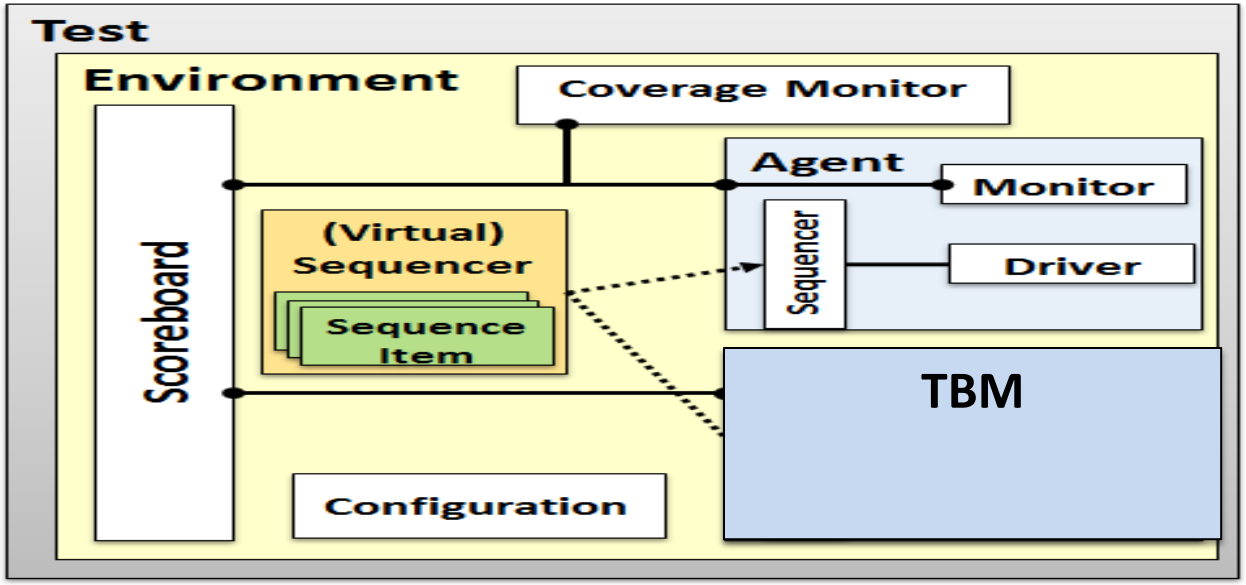


# Conclusion

- evolution of tailored verification methods to UVM
- UVM agents can be controlled in SystemC test environments
- approach is generic: IFS library is not mandatory
- standards for SV/DPI and SC are mature to support UVMC
  - worked for simulators VCS, Questa and Incisive

- complex interferences between SystemVerilog, UVM, UVMC, SystemC and simulator
- future work
  - reverse approach and use TBM in UVM test environment
  - Accellera Multi Language Working Group (MLWG):
    "to create a standard and functional reference for interoperability of multi-language verification environments and components."

# Conclusion




- future work
  - reverse approach and use TBM in UVM test environment
  - Accellera Multi Language Working Group (MLWG):
    "to create a standard and functional reference for interoperability of multi-language verification environments and components."

# Conclusion

- evolution of tailored verification methods to UVM
- UVM agents can be controlled in SystemC test environments
- approach is generic: IFS library is not mandatory
- standards for SV/DPI and SC are mature to support UVMC
  - worked for both simulators Questa and Incisive
  - no real doubt would work with VCS
- complex interferences between SystemVerilog, UVM, UVMC, SystemC and simulator
- future work
  - reverse approach and use TBM in UVM test environment
  - Accellera Multi Language Working Group (MLWG):
    "to create a standard and functional reference for interoperability of multi-language verification environments and components."

# Thank You

**Acknowledgements:** This work has been funded by the German Federal Ministry for Education and Research (Bundesministerium für Bildung und Forschung, BMBF) under the grant 01IS13022 (project EffektiV). The content of this publication lies within the responsibility of the authors.