OFFIS
INSTITUTE FOR
INFORMATION TECHNOLOGY

BOSCH

Connecting a Company's Verification Methodology to Standard Concepts of UVM

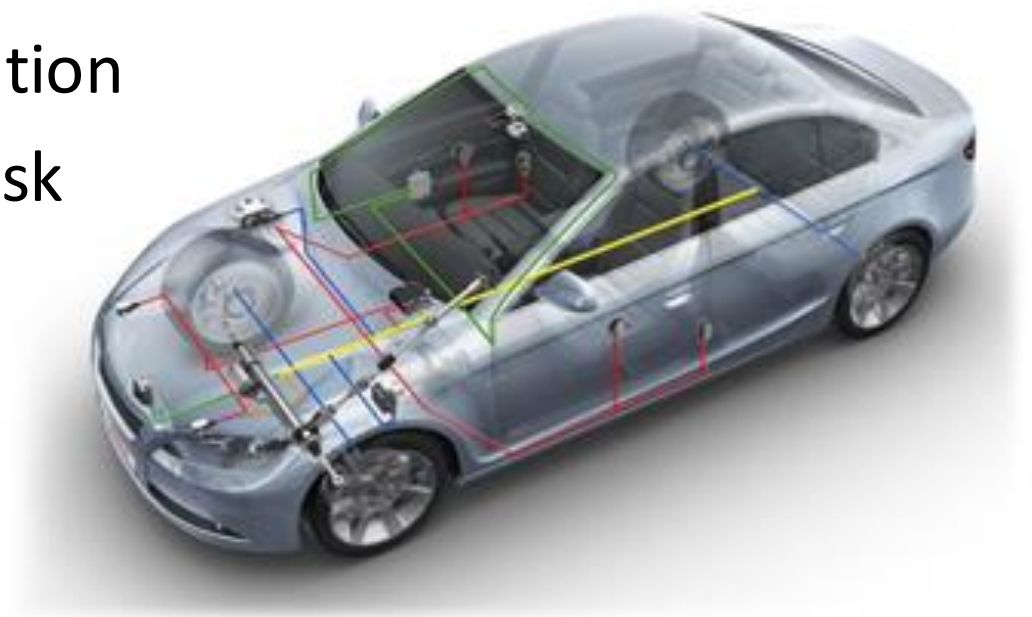
Frank Poppen, OFFIS
Marco Trunzer, Robert Bosch GmbH
Jan-Hendrik Oetjens, Robert Bosch GmbH

accellera
SYSTEMS INITIATIVE

2014
DESIGN AND VERIFICATION
DVCON
CONFERENCE AND EXHIBITION
EUROPE

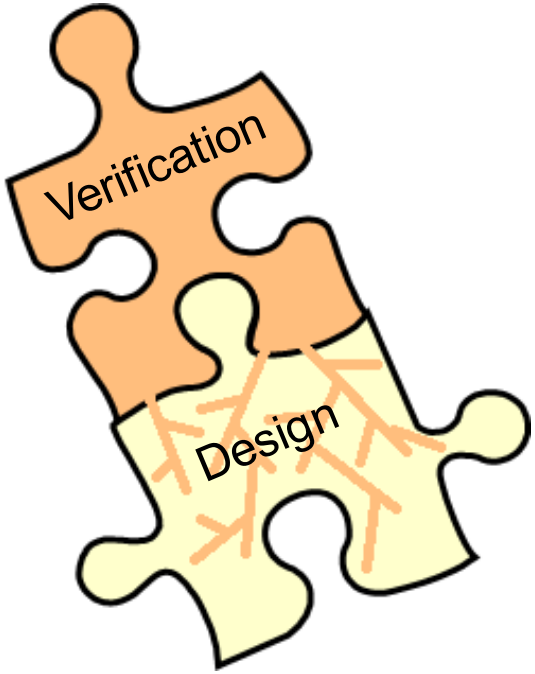
The Challenge

- 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“

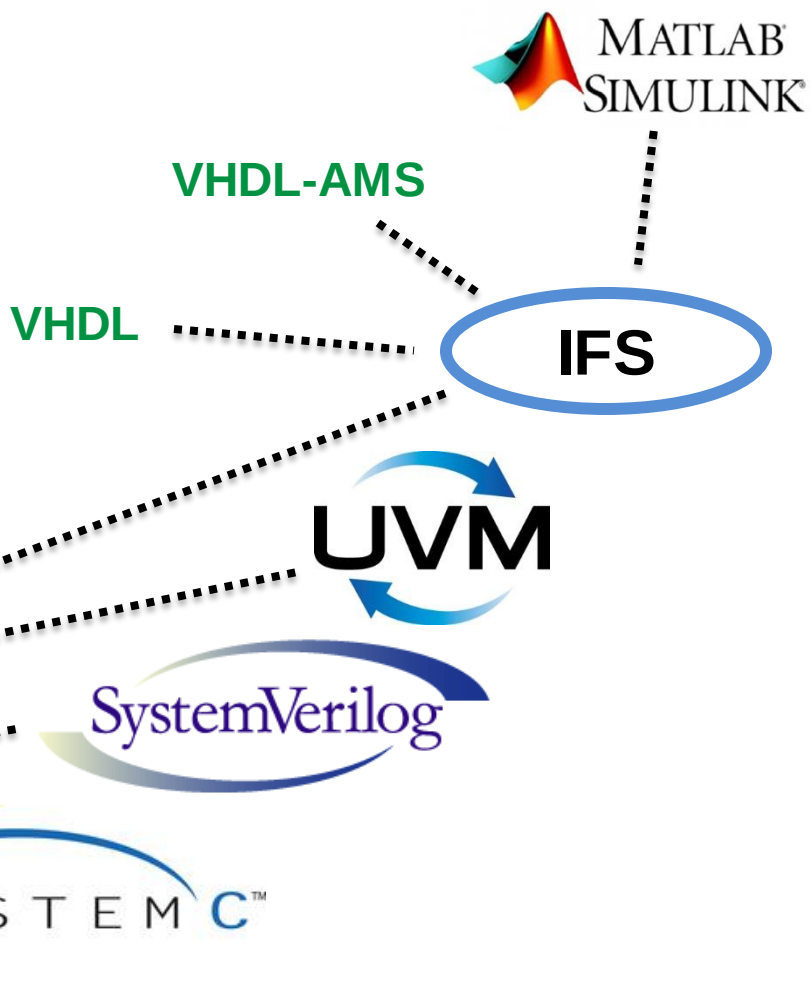
- verification engineers choose and combine what …
  - fits best for the company
  - the design-team
  - the application domain
  - (budget, roadmap, …)
- deep roots in the design process
- changes endanger productivity
- change carefully and incrementally



Outline

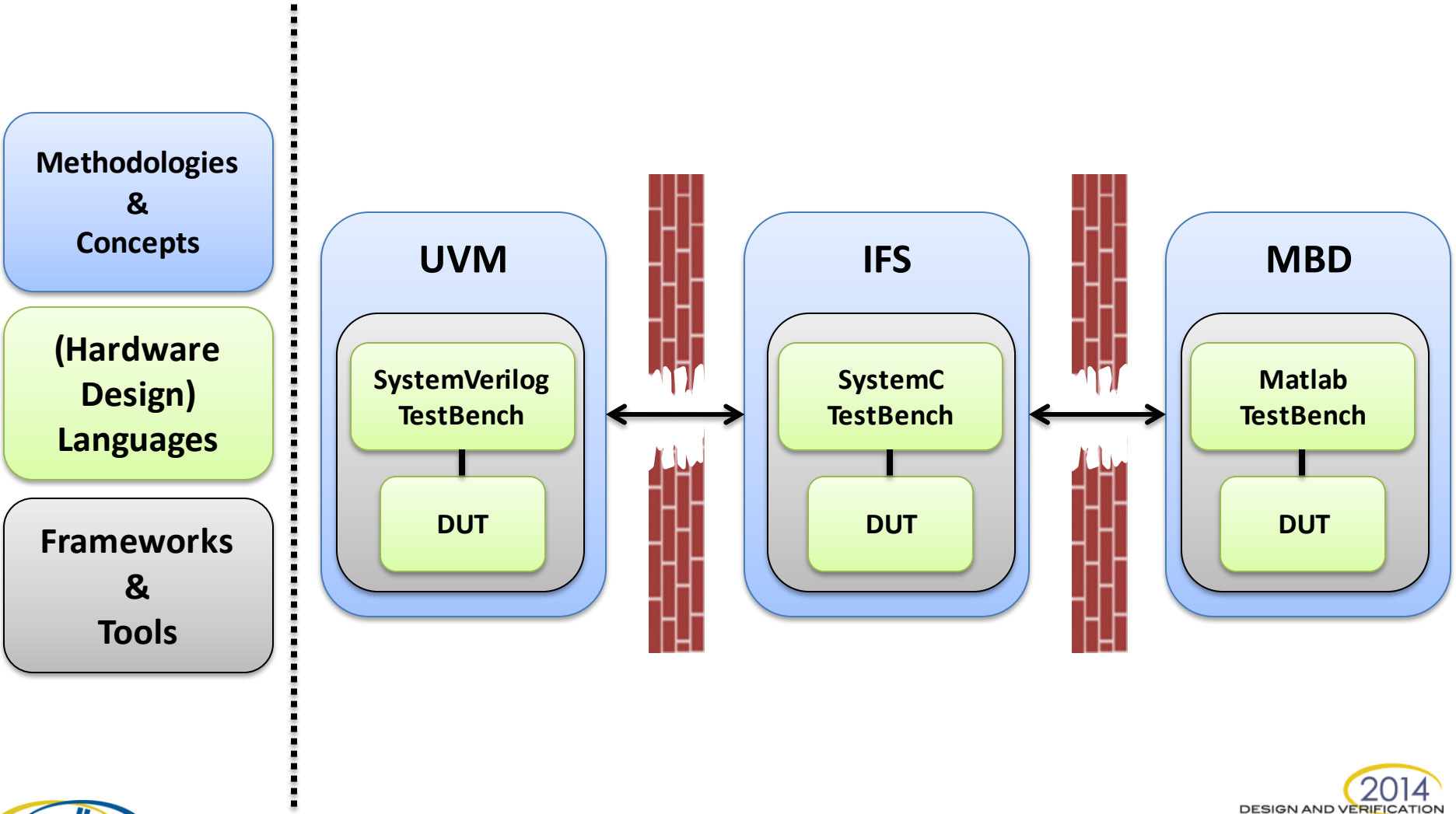
- Motivation
- Verification Islands
- Matching Concepts of UVM and IFS
- Bus Arbiter Test Case
- UVMC can do / cannot do
- Architecture of the Experiment
- Conclusions

Bringing it All Together



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

Verification Islands

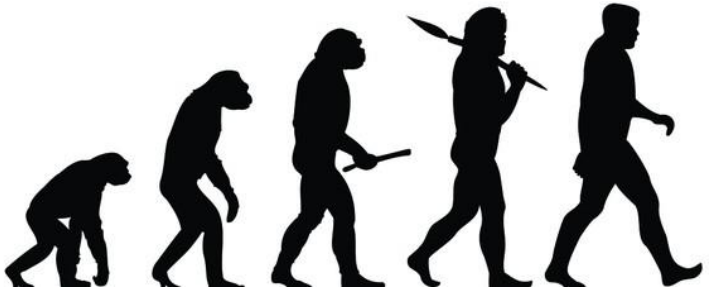


Outline

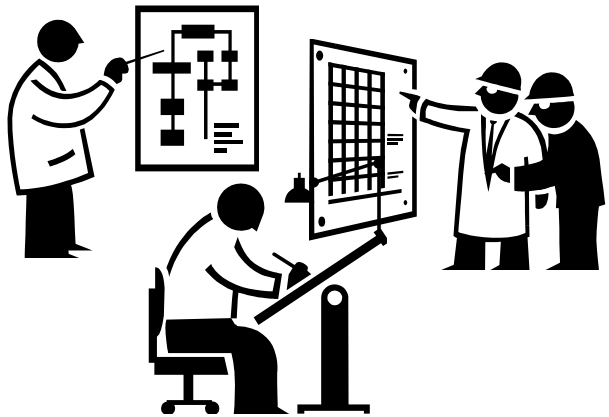
- Motivation
- Verification Islands
- Matching Concepts of UVM and IFS
- Bus Arbiter Test Case
- UVMC can do / cannot do
- Architecture of the Experiment
- Conclusions

IFS long before SystemVerilog

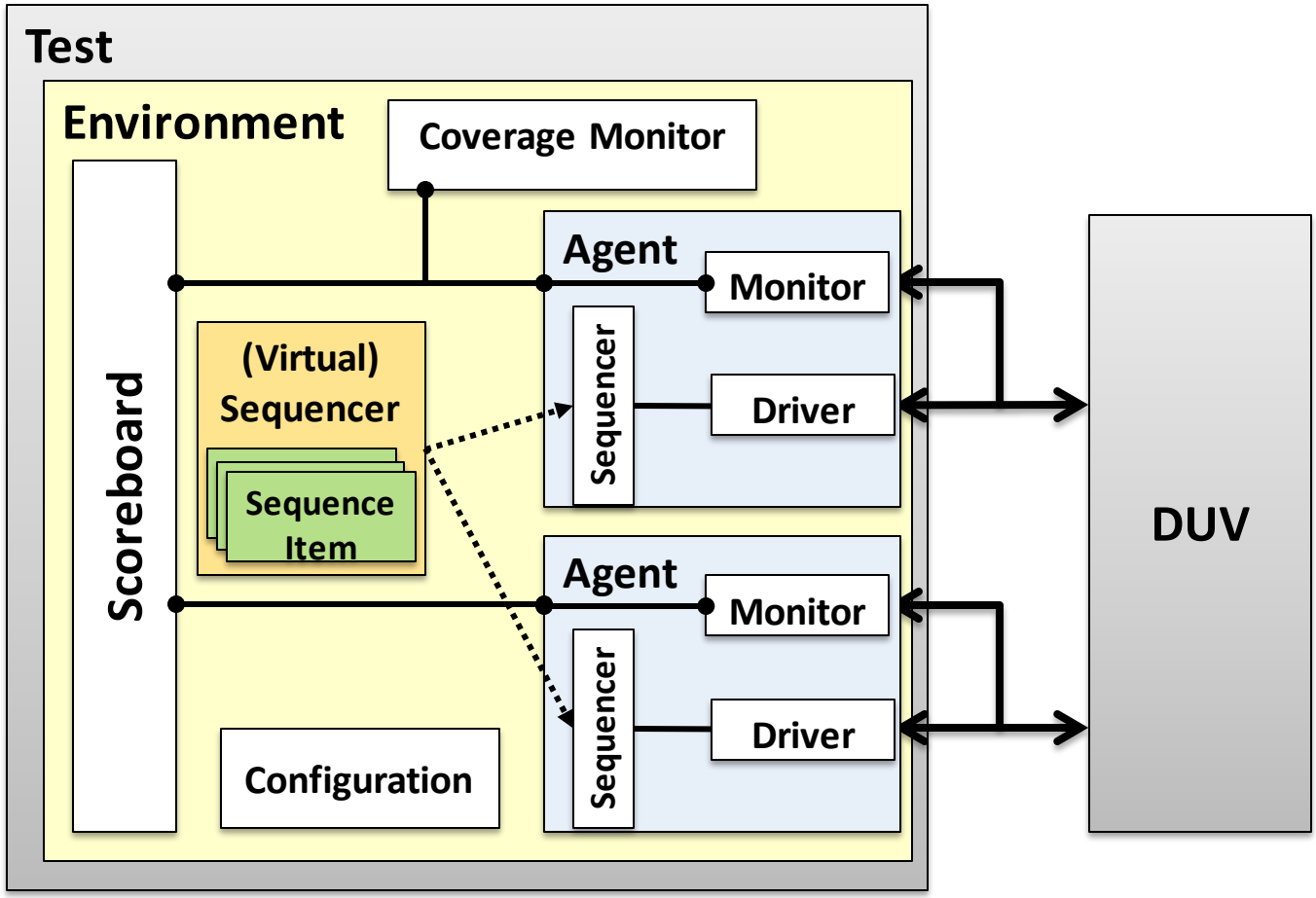
- enhanced from VHDL with …
  - VHDL-AMS
  - SystemC
  - Matlab/Simulink
  - (and now SystemVerilog and UVM)



- 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

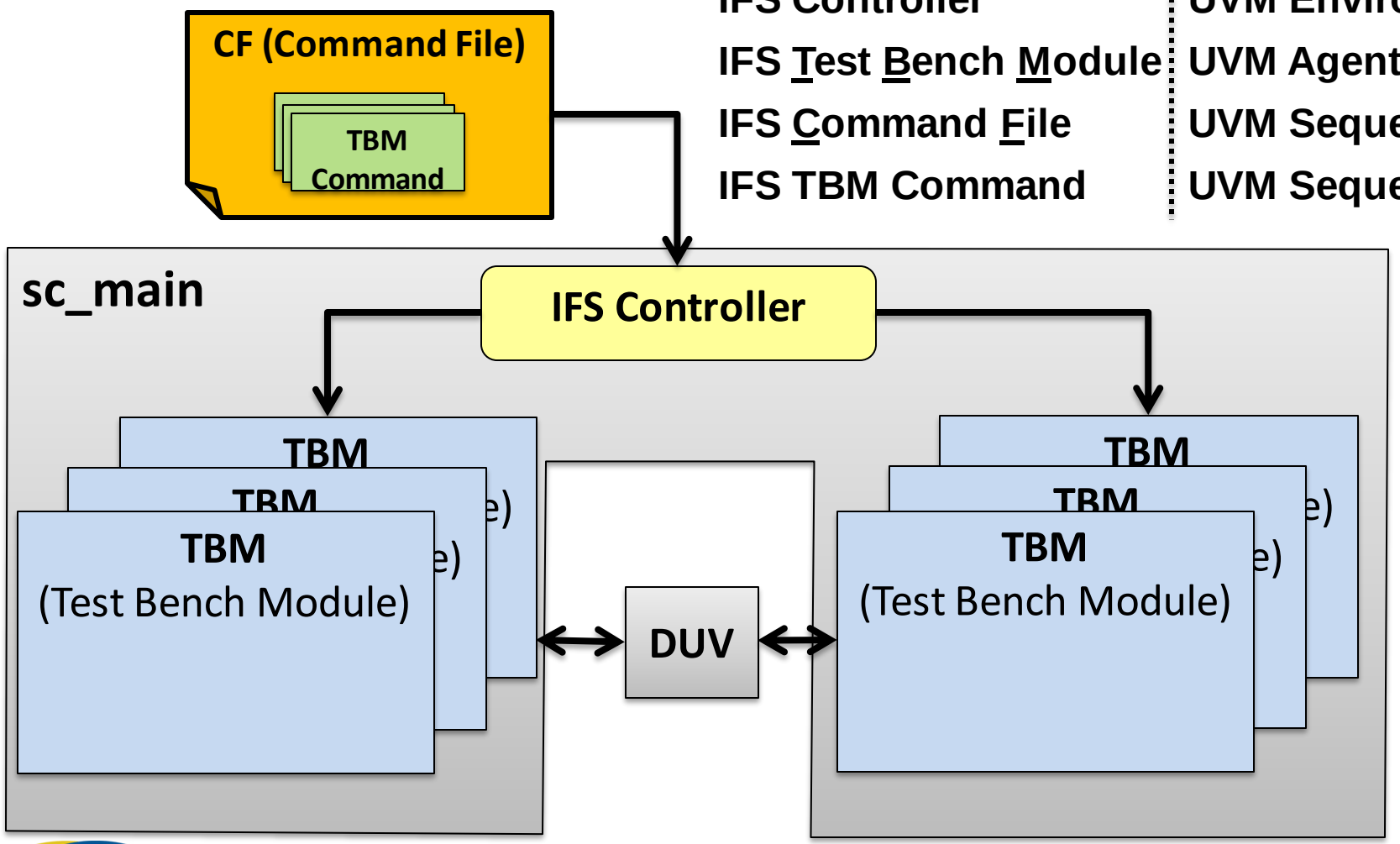
UVM Agent

UVM Sequence

UVM Sequence Item

Matching Concepts of UVM and IFS

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



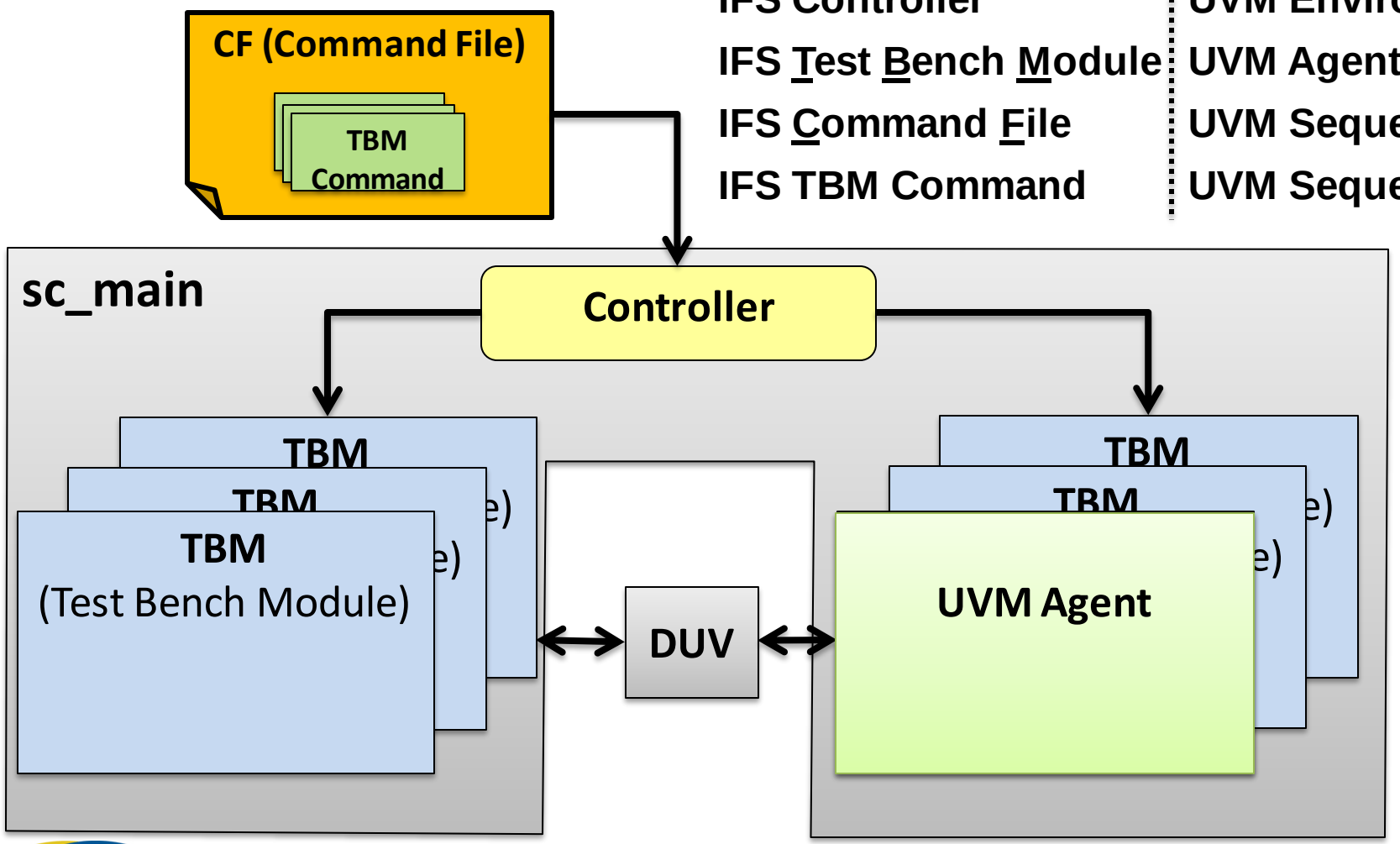
Matching Concepts of UVM and IFS

- 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

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

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

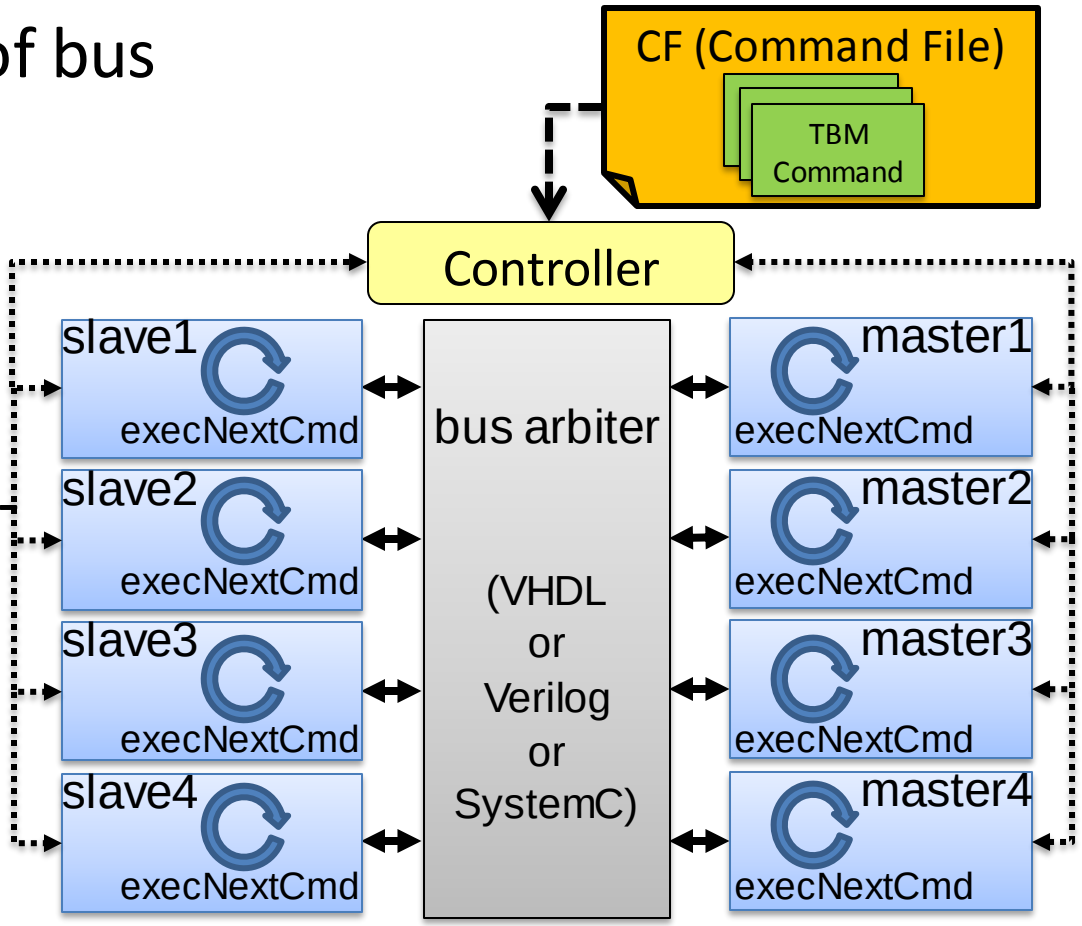


Outline

- Motivation
- Verification Islands
- Matching Concepts of UVM and IFS
- Bus Arbiter Test Case
- UVMC can do / cannot do
- Architecture of the Experiment
- Conclusions

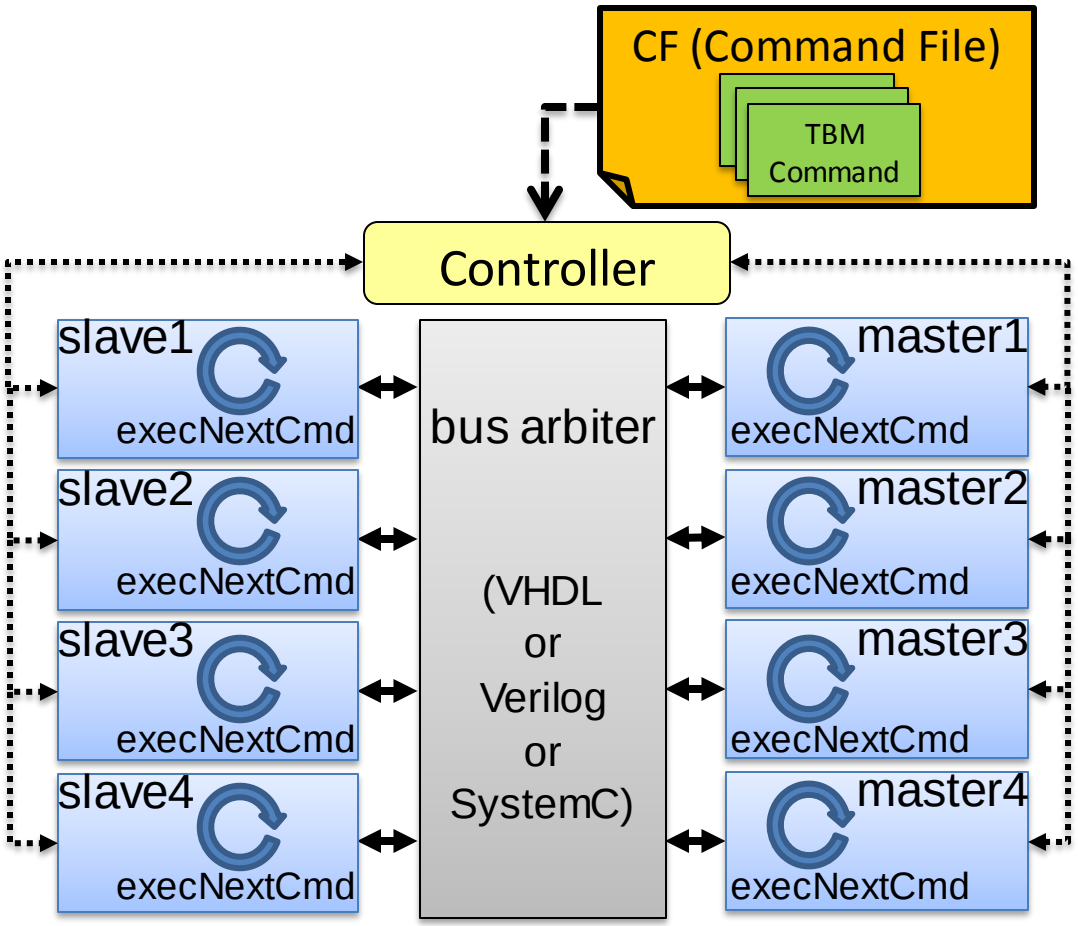
Simple Switched Bus Arbiter Test Case

- 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



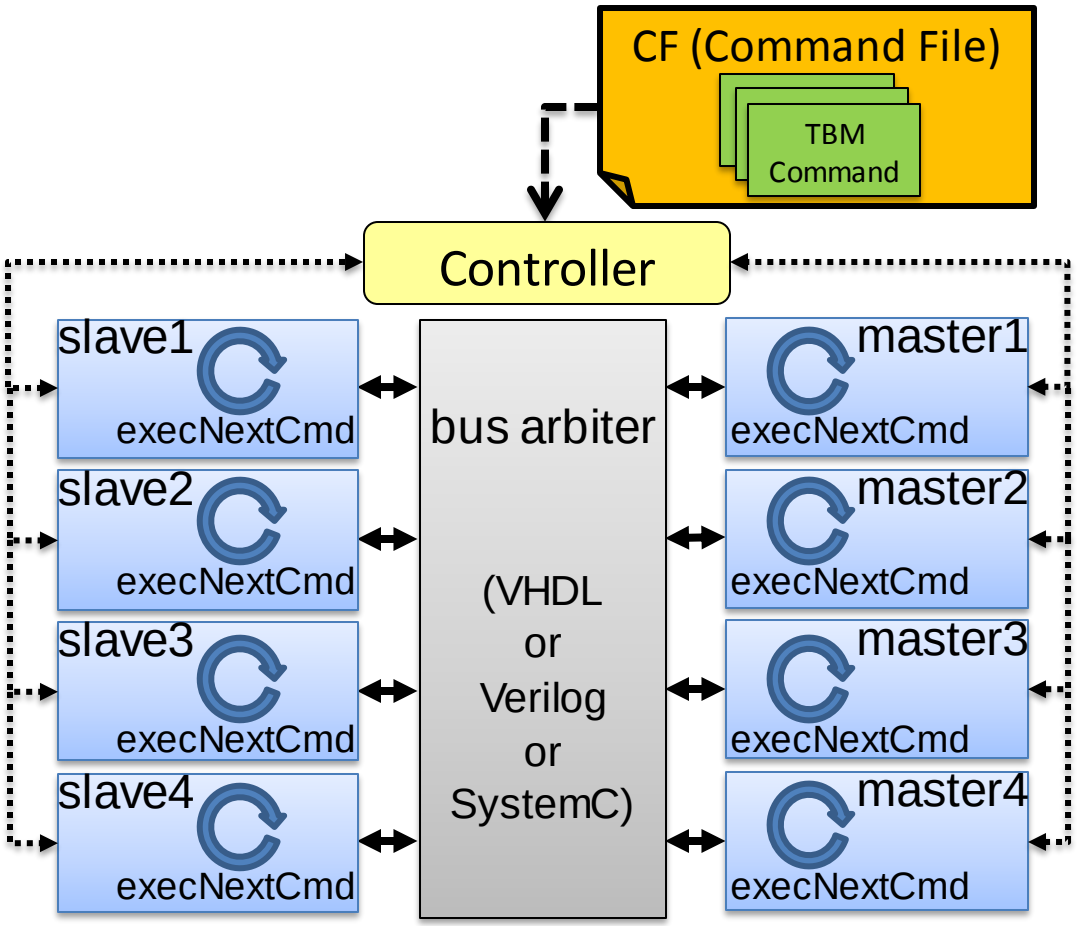
Simple Switched Bus Arbiter Test Case

```
CLK PERIOD 10 ns
CLK RESET  0  12

ALL SYNC ALL

SL1 print "Config slave 1!"
SL1 Set_Offset 300
SL1 Set_I_Wait 100
SL2 print "Config slave 2!"
SL2 Set_Offset 200
SL2 Set_I_Wait 99
SL3 print "Config slave 3!"
SL3 Set_Offset 100
SL3 Set_I_Wait 47
SL4 print "Config slave 4!"
SL4 Set_Offset 0
SL4 Set_I_Wait 69

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 Test Case

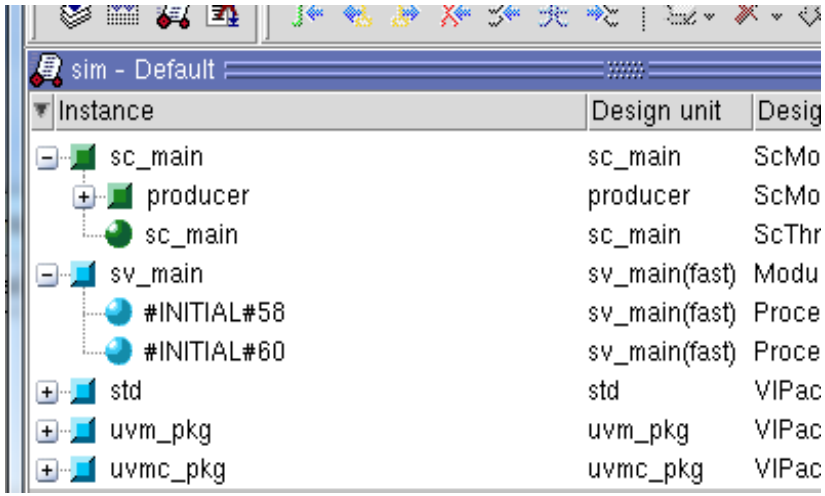
```
SystemC 2.3.0-ASI --- Nov 29 2013 14:57:17
 Copyright (c) 1996-2012 by all Contributors,
 ALL RIGHTS RESERVED
INFO (0 s) Loading script: 'control.cmd'
...
[SL3,350 ns]      Read     (Address: 100, Value: 101)
[MS1,360 ns]      Read     (Address: 100, Value: 101)
[MS1,410 ns]      Write    (Address: 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
```

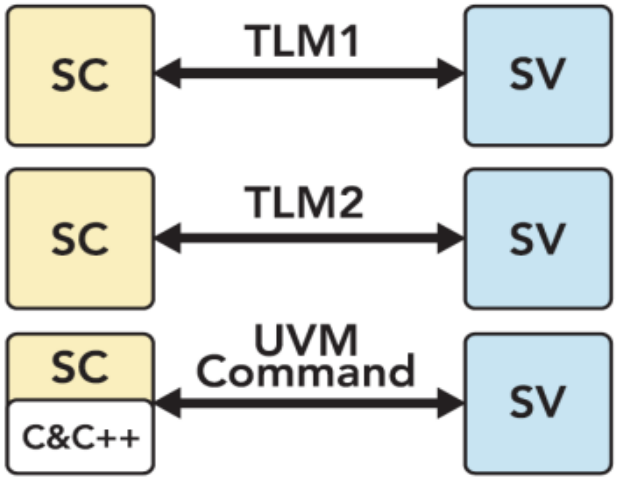


Outline

- Motivation
- Verification Islands
- Matching Concepts of UVM and IFS
- Bus Arbiter Test Case
- UVMC can do / cannot do
- Architecture of the Experiment
- Conclusions

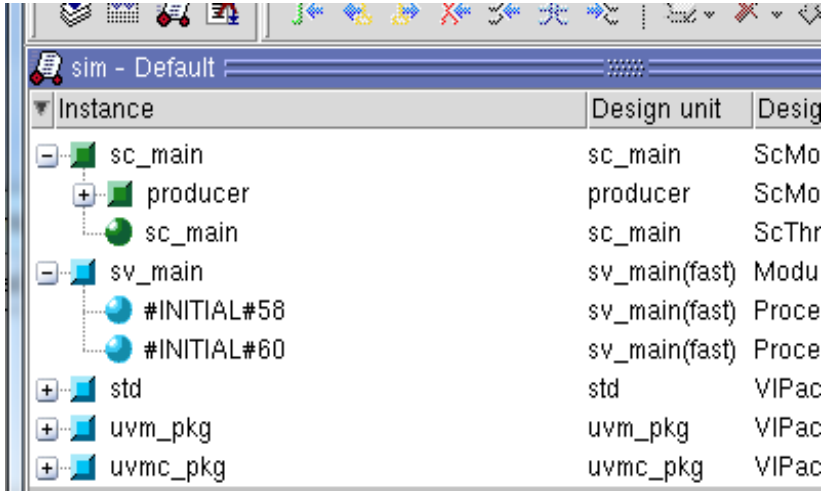
UVM Connect (Verification Academy)

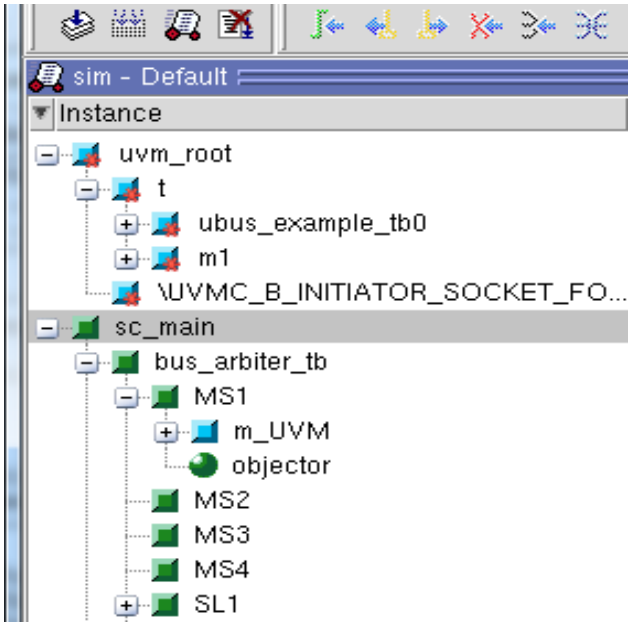




- interconnect UVM and SystemC
- UVMC is an open-source UVM/OVM-based library
- uses System Verilog Direct Programming Interface
- good for TLM messages and control commands
- less good for RT level signal interfacing

UVM Connect (Verification Academy)



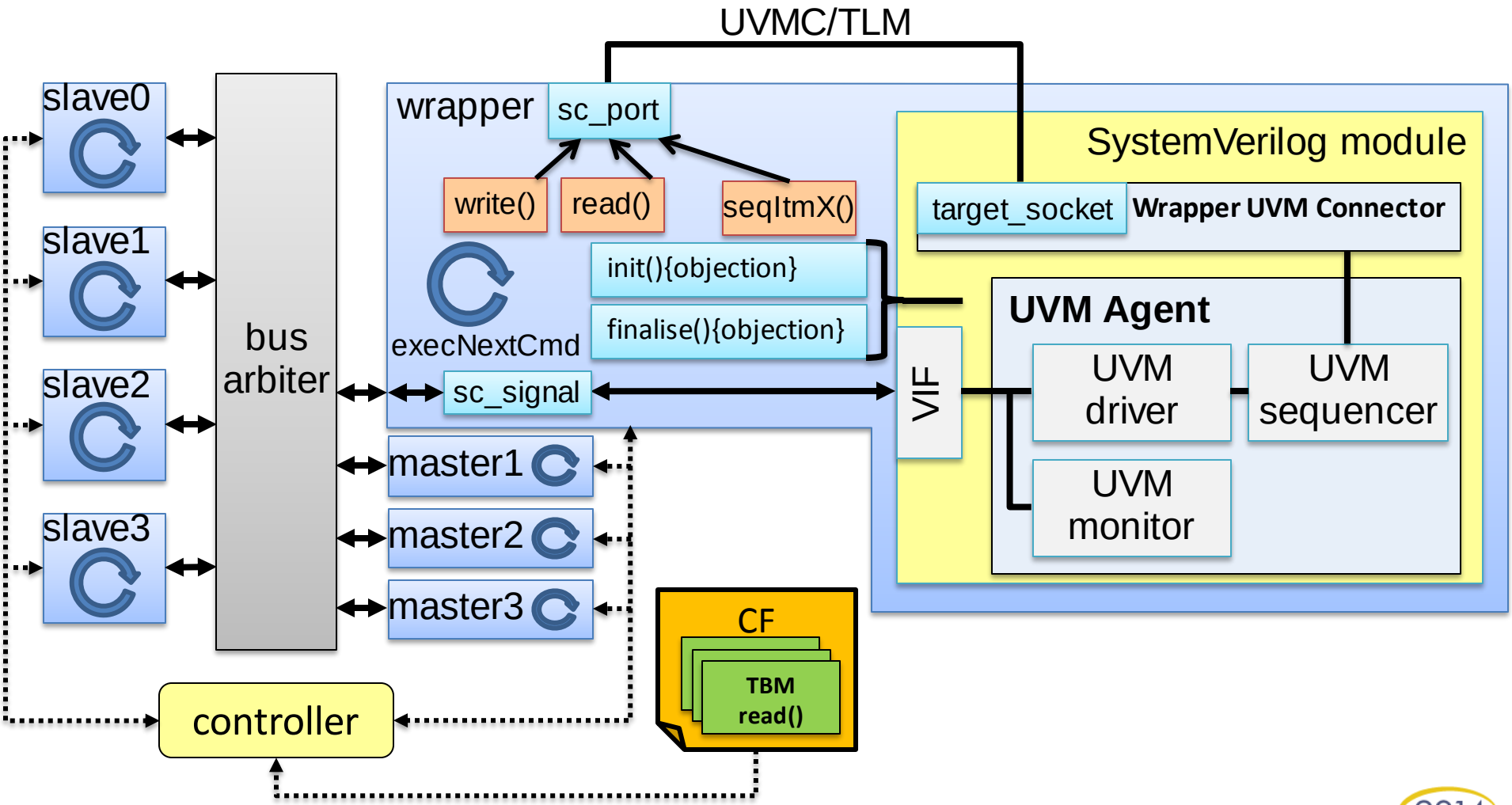


- interconnect UVM and SystemC
- UVMC is an open-source UVM/OVM-based library
- uses System Verilog Direct Programming Interface
- good for TLM messages and control commands
- less good for RT level signal interfacing
  => instantiation through foreign language module

Outline

- Motivation
- Verification Islands
- Matching Concepts of UVM and IFS
- Bus Arbiter Test Case
- UVMC can do / cannot do
- Architecture of the Experiment
- Conclusions

Architecture of the Experiment



Running the Experiment

```
1:   irun(64): 13.10-s005: (c) Copyright 1995-2013 Cadence Design Systems, Inc.
2:   Loading snapshot worklib.bus_arbiter_tb:sc_module ..................... Done
3:   ********** foreign_module_master::CTOR(): Elaborating foreign module: sv_uvm_master
4:   ********** wrap_uvm_master::CTOR(): Connecting TLM port
5:   Connecting an SC-side proxy chan for 'bus_arbiter_tb.MS1.port_10' with lookup string
     'sc_wrap_MS1' for later connection with SV
6:   INFO (0 s)[bus_arbiter_tb.MS1] Registered module 'bus_arbiter_tb.MS1'
7:   ...(also MS2 to MS4)
8:   INFO (0 s)[bus_arbiter_tb.SL1] Registered module 'bus_arbiter_tb.SL1'
9:   ...(also SL2 to SL4)
10:  INFO (0 s)[bus_arbiter_tb.CLK] Registered module 'bus_arbiter_tb.CLK'
11:  INFO (0 s) Loading script: 'control.cmd'
```

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

Running the Experiment

```
12:  ncsim> run
13:  ------------------------------------------------------------
14:  CDNS-UVM-1.1d (13.10-s005)
15:  (C) 2007-2013 Mentor, Cadence, Synopsys, Cypress Semiconductor
16:  ------------------------------------------------------------
17:  UVM_INFO    ../tb_uvm/uvm_master_module.sv(57)    @    0:    reporter    [sv_uvm_master]    *********
     sv_uvm_master: Initialising instance sc_wrap_MS1 and its virtual interface.
18:  ------------------------------------------------------------
19:  UVMC-2.2
20:  (C) 2009-2012 Mentor Graphics Corporation
21:  ------------------------------------------------------------
22:  Registering  SV-side  'sc_wrap_MS1.ifs_monitor.in'  and  lookup  string  'sc_wrap_MS1'  for  later
     connection with SC
23:  UVM_INFO @ 0 ns: reporter [RNTST] Running test ...
```

- running simulation
- UVM phasing starts after SystemC elaboration
- UVMC starts
- System Verilog part of UVMC opens port

Running the Experiment

```
24:  ************** wrap_uvm_master::initialiseModule(): Raising objection for UVM phase 'run'!
25:  Connected SC-side 'bus_arbiter_tb.MS1.port_10' to SV-side 'sc_wrap_MS1.ifs_monitor.in'
26:  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...
27:  [bus_arbiter_tb.CLK,100 ns] reset gets passiv
28:  [bus_arbiter_tb.SL1,100 ns] set slave_offset = 300
29:  ... (configuring SL1 to SL4)
30:  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!
31:  ************** wrap_uvm_master::Write(): Sending payload '{cmd:2 parameters:100 98} to MS1.socket
     at time 150
32:  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"}
33:  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!
34:  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!
```

- raising run phase objection
- both ports SC and SV get connected
- SC sending payload / SV receiving payload
- UVM agent (driver) receiving sequence item

Running the Experiment

```
52:   INFO (2030 ns) SIMULATION END FROM COMMAND FILE
53:   INFO (2030 ns) Exiting simulation.
54:   ********* wrap_uvm_master::finaliseModule(): Dropping objection for UVM phase 'run'!
55:   SC simulation stopped by user.
56:   SystemC : SystemC stopped at time 2030
57:   ncsim> exit
```

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

Outline

- Motivation
- Verification Islands
- Matching Concepts of UVM and IFS
- Bus Arbiter Test Case
- UVMC can do / cannot do
- Architecture of the Experiment
- 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
- 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."

OFFIS
INSTITUTE FOR
INFORMATION TECHNOLOGY

BOSCH

Questions

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.

Effektiv

accellera
SYSTEMS INITIATIVE

2014
DESIGN AND VERIFICATION
DVCON
CONFERENCE AND EXHIBITION
EUROPE