

A SystemC – HDL Cosimulation Framework

Christian Bernard, CEA/LETI

Nicolas Tribié, CEA/LETI

Marcello Coppolla, ST/AST

Agenda

- **Motivations**
- **Cosimulation usages**
- **Framework Architecture – C++ Code Generation**
- **Example : Performances Achieved**
- **Future Improvements**
- **Q & A**

Motivations

- Introduce SystemC for system-level specification ,
- Lack of SystemC synthesis tools : RTL description still in VHDL/Verilog
- Validation : Check RTL / SystemC conformance.
- Simulation speed up : Testbench described at transaction level (SystemC)

➔ Need for a tool allowing  ↔ VHDL cosimulation :

- Low cost (SystemC is free !), focus on cosimulation *only*
- Compatible with main RTL simulators (Modelsim™, NCSim™..)
- Minimize user-written additional code

Usage example

Memory Controller Designer's view

- Focus on the memory controller implemented down to RTL using VHDL.

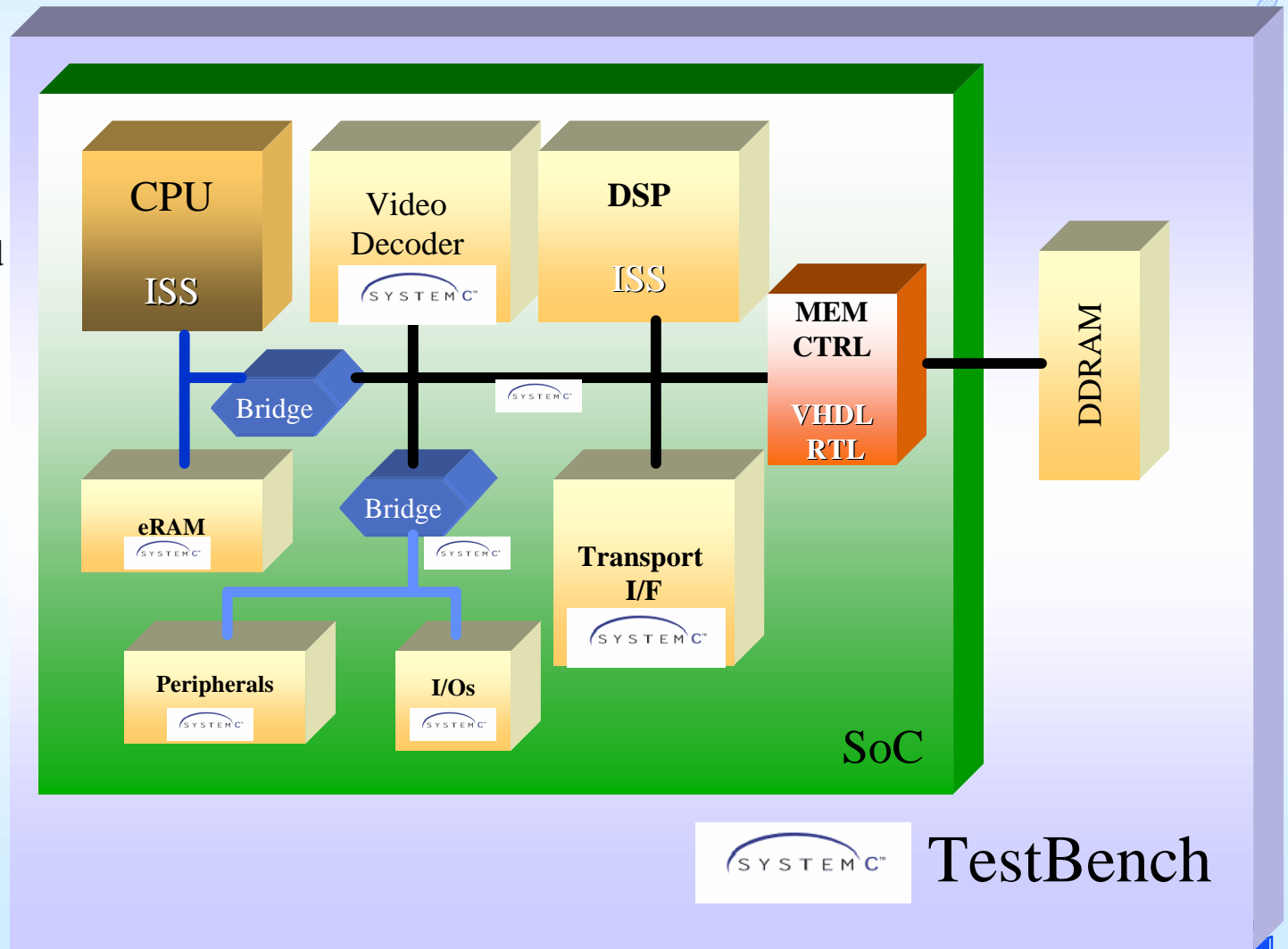
- Others modules are still described at transaction level

➔ • Simulation speed up

- Testbench can be written in SystemC

➔

- Eases development
- Benefits from C++
- Easy to maintain & reuse



Cosimulation Principles

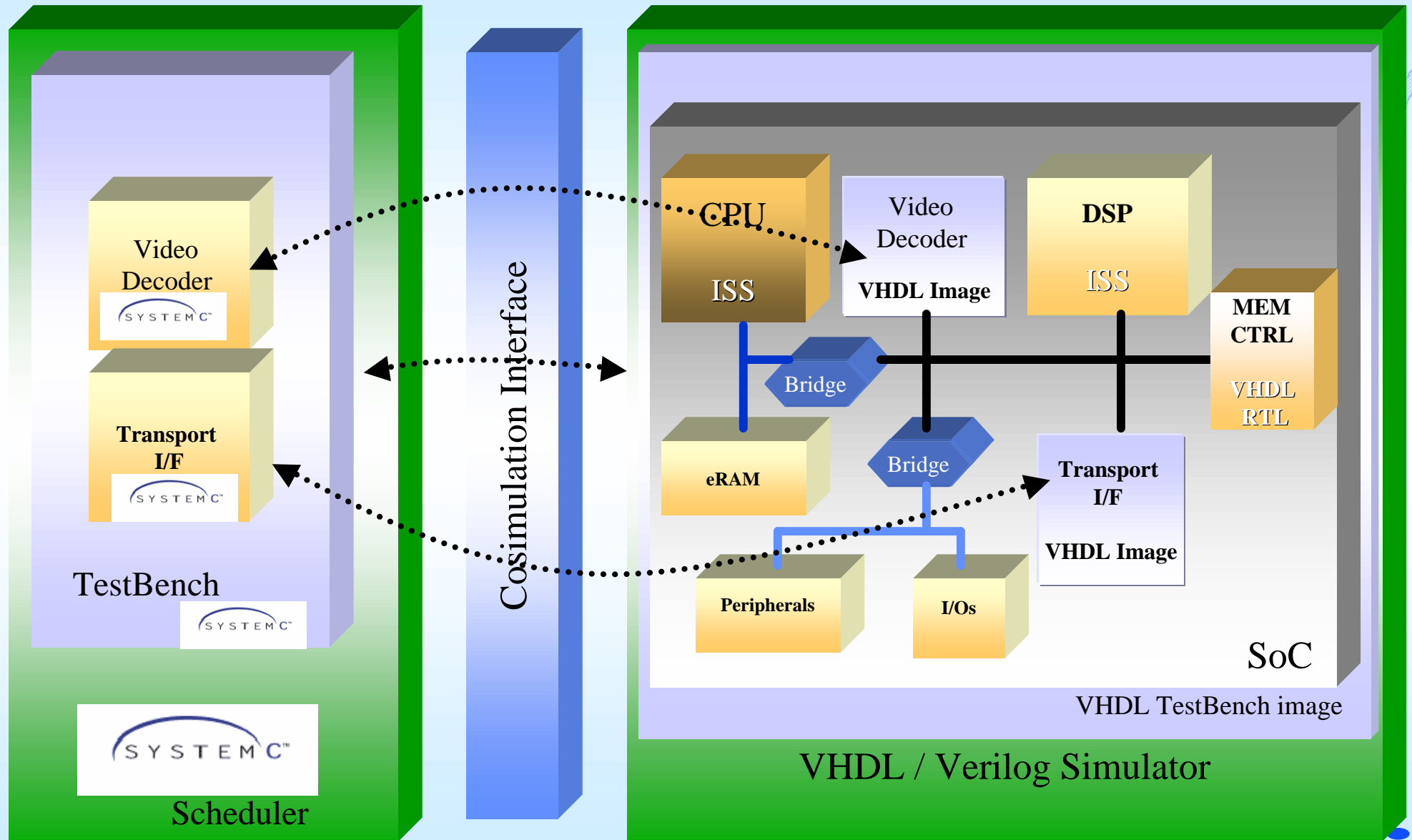
Assumptions :

- ASIC hierarchy described in VHDL / Verilog
- SystemC Hierarchy description is also possible
- Blocks described in SystemC are replaced by a VHDL image in the hierarchy
- Clocks are replicated both in the HDL simulator and SystemC scheduler
- Several clocks domains are allowed

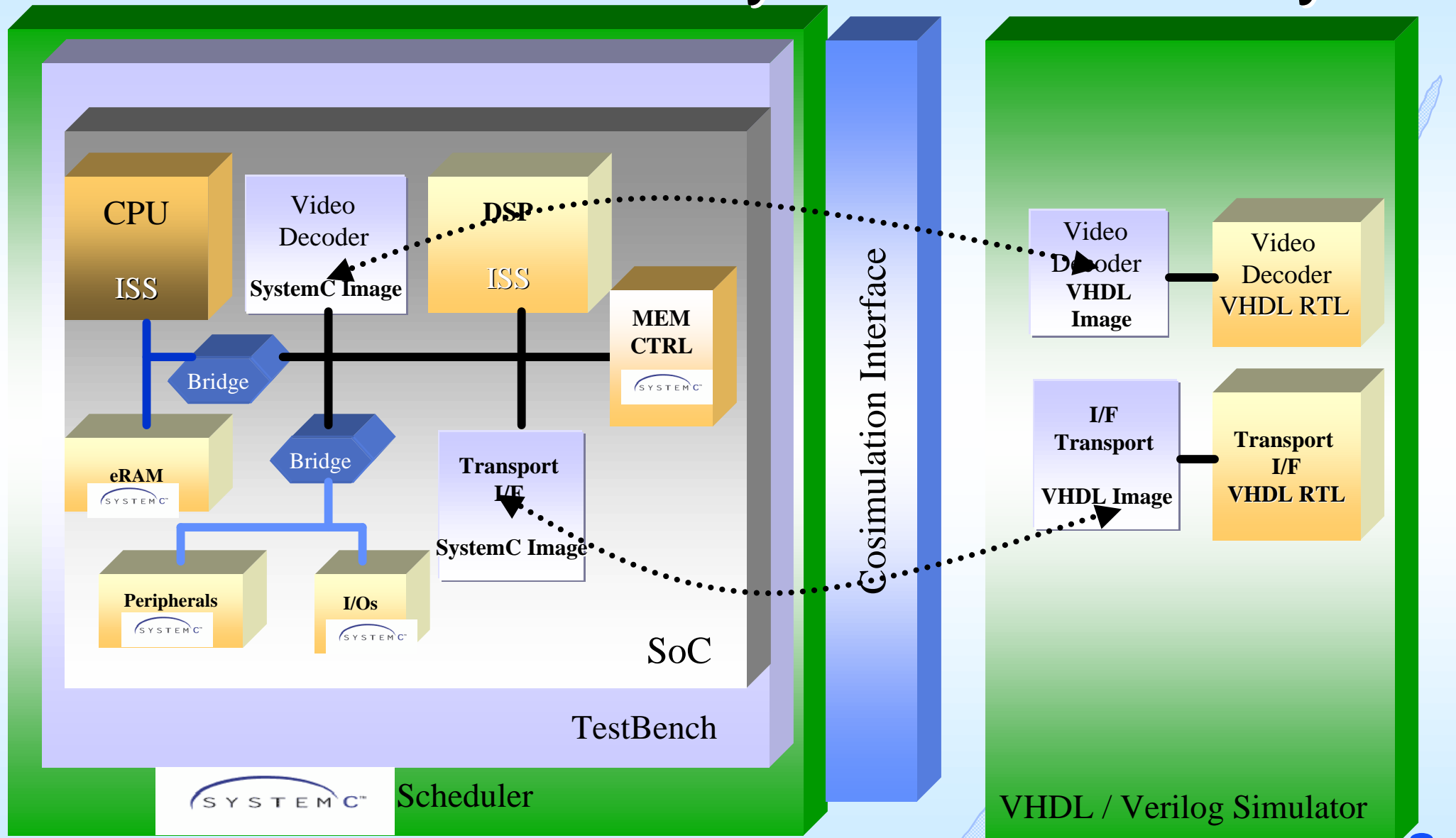
Restrictions :

- Blocks interfaces **must be synchronous to a global clock**
- Delta cycles are not supported
- Only one exchange between VHDL and SystemC models at each cycle
- No delays in combinatorial logic of the simulated RTL

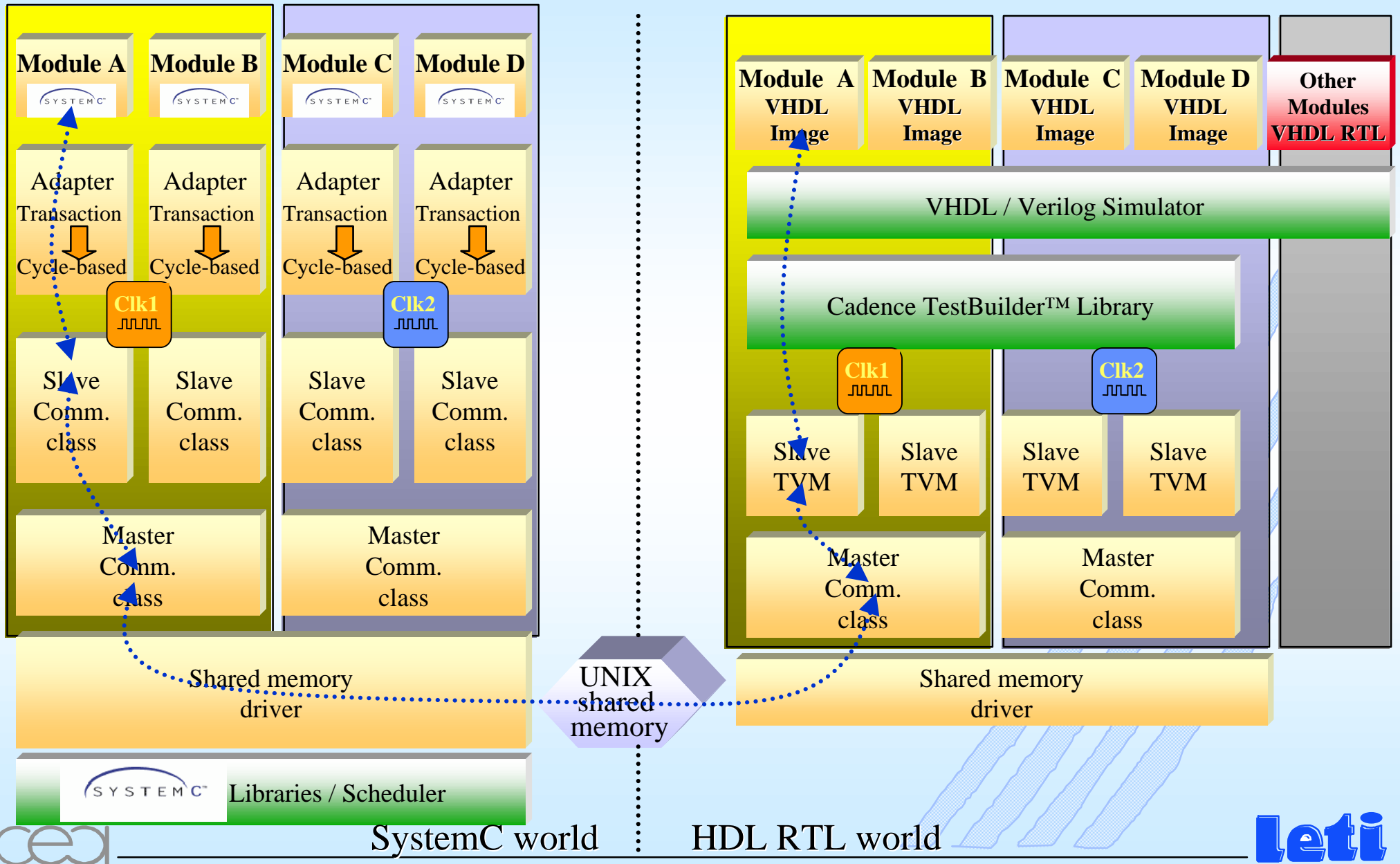
SystemC blocks in a VHDL hierarchy



VHDL blocks in a SystemC hierarchy



Architecture

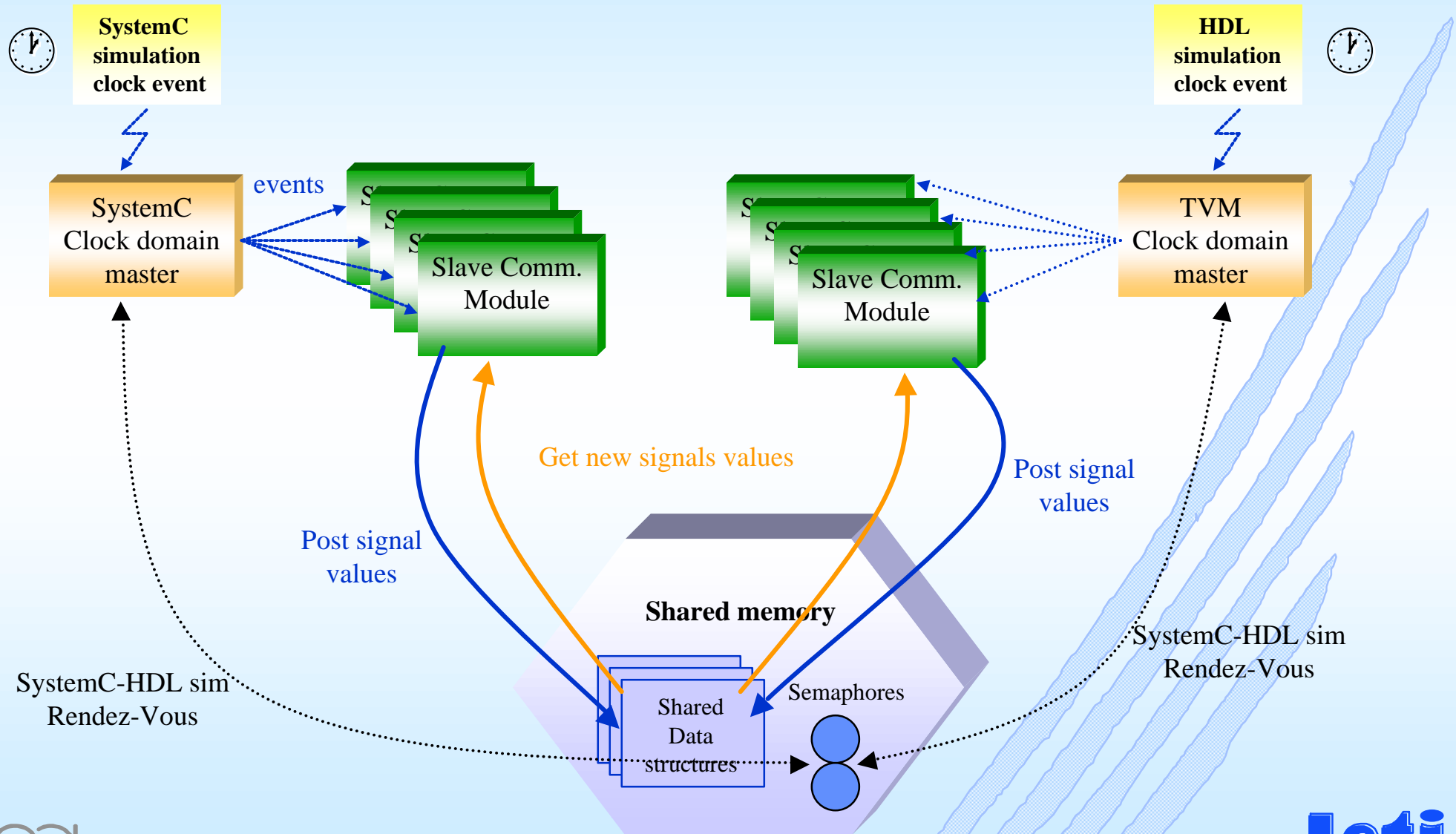


SystemC world

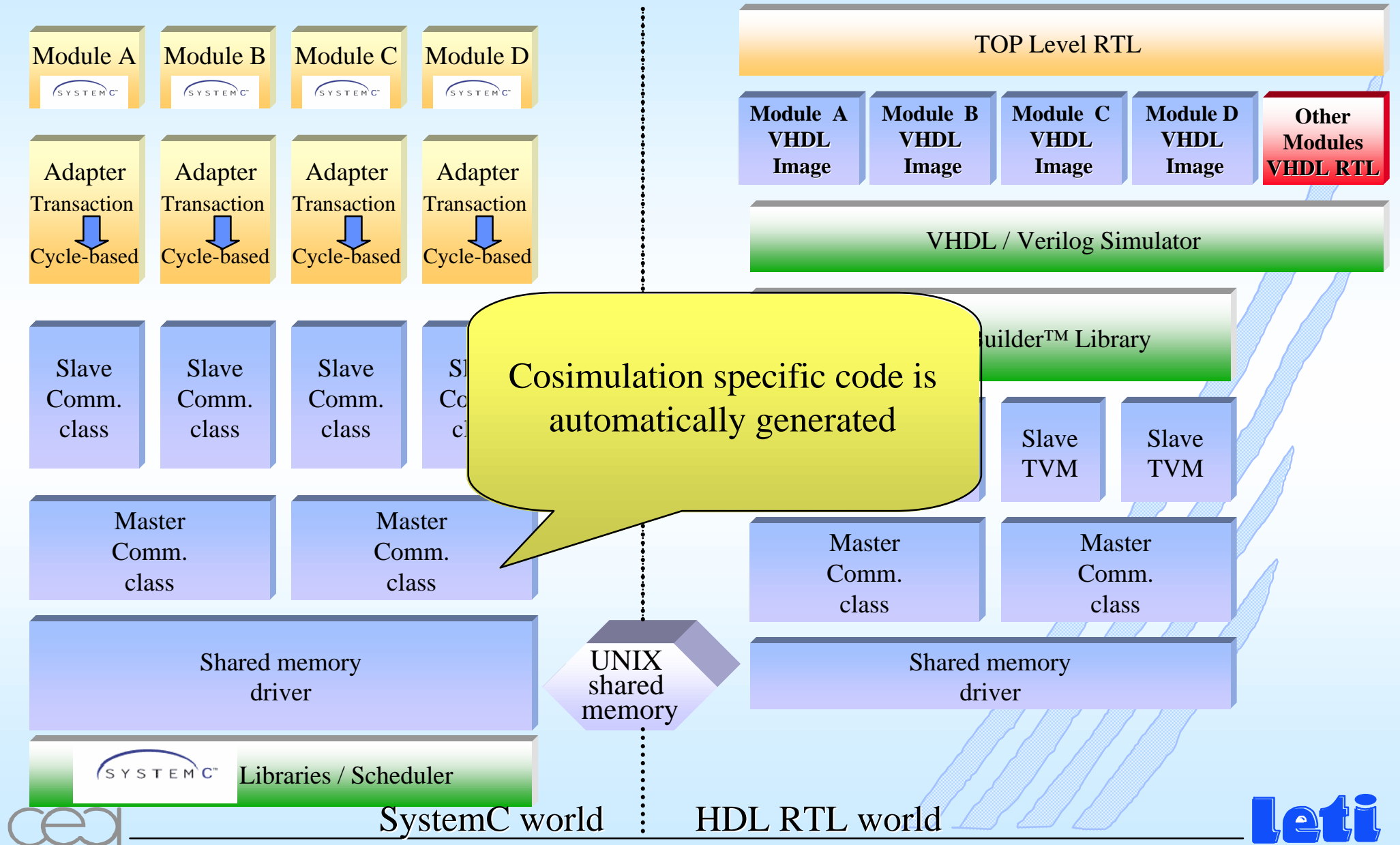
HDL RTL world



Communication Flow



C++ Code Generation

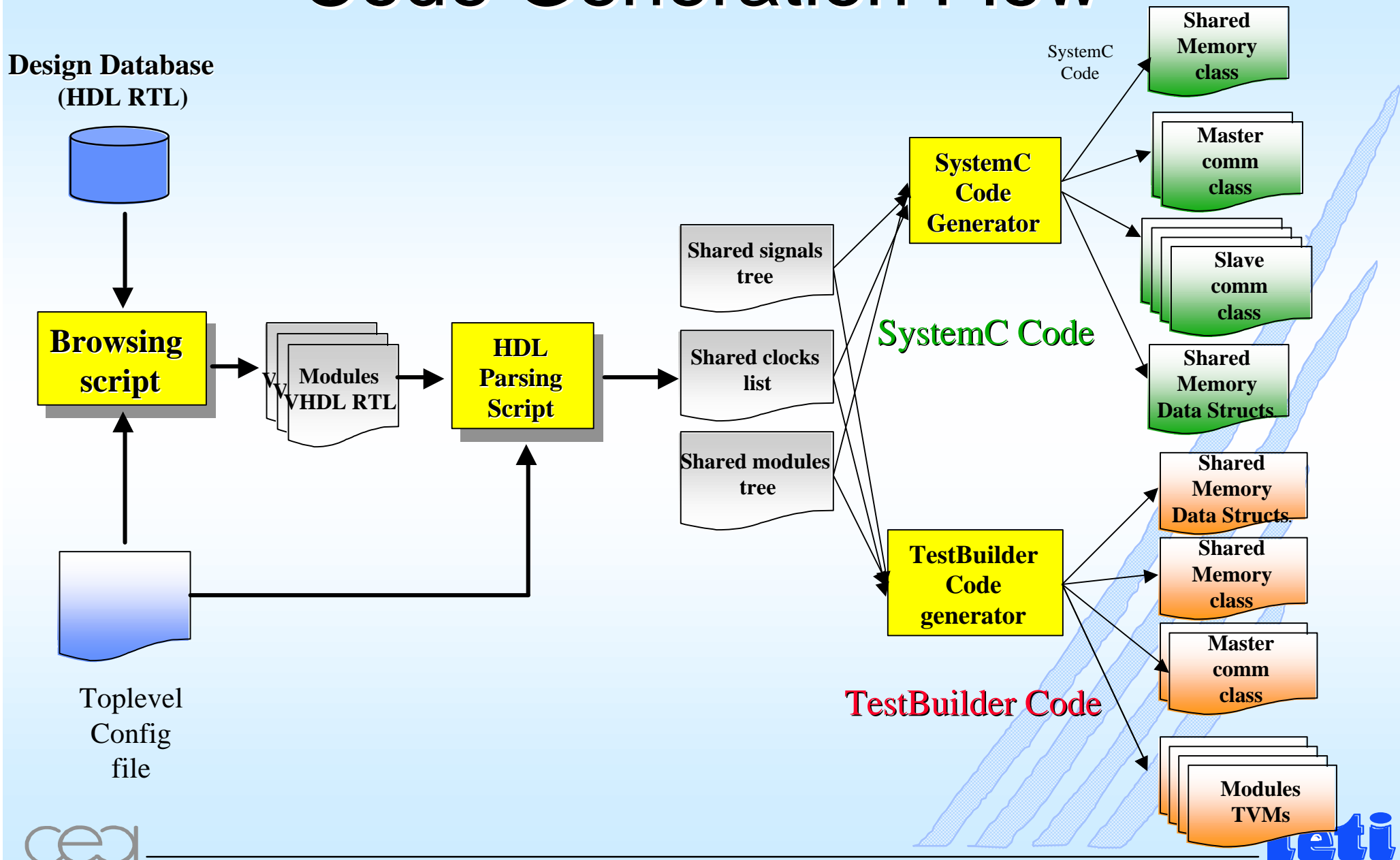


SystemC world

HDL RTL world



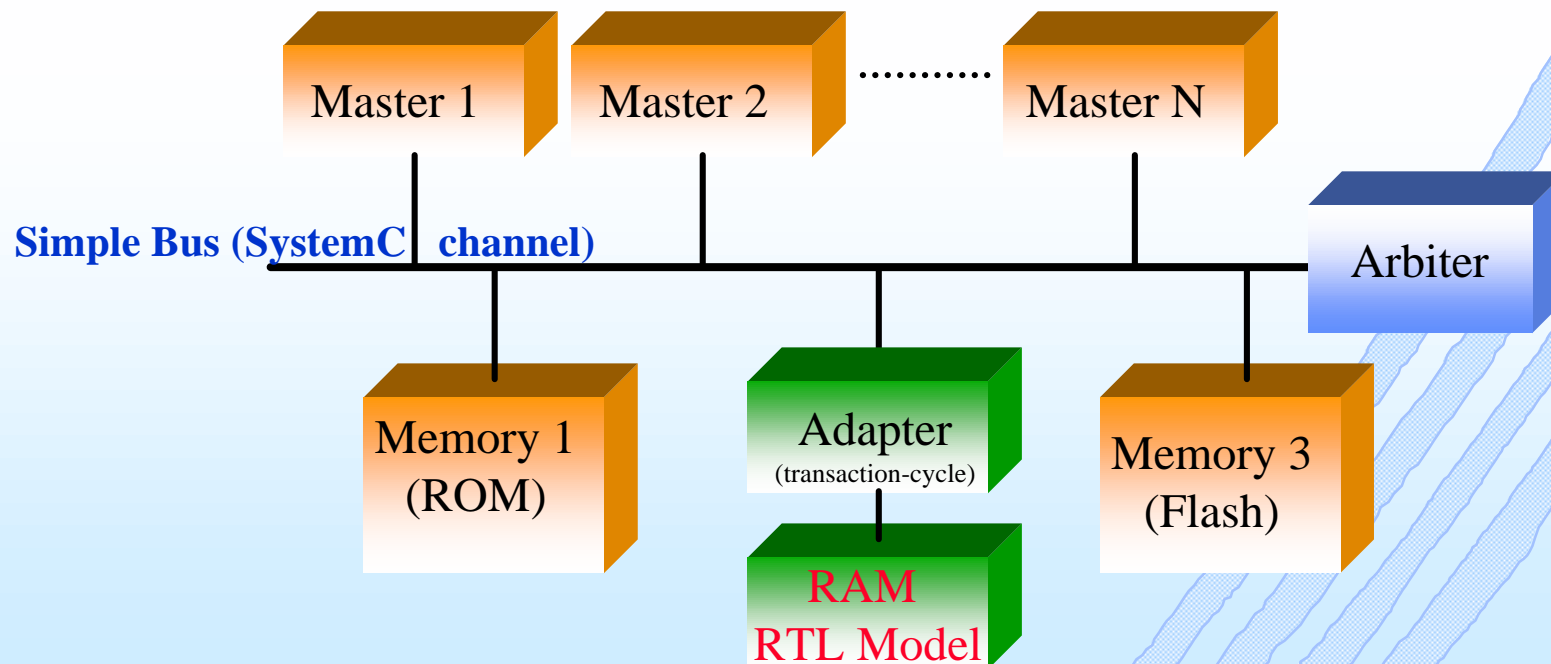
Code Generation Flow



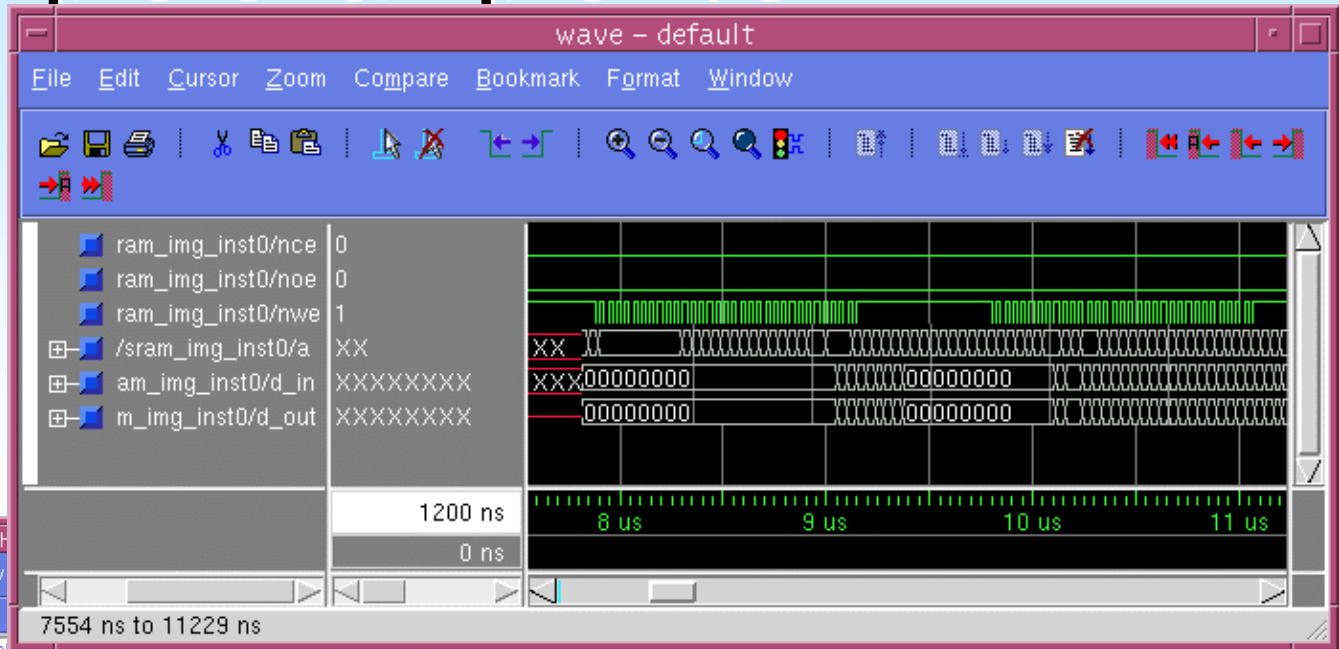
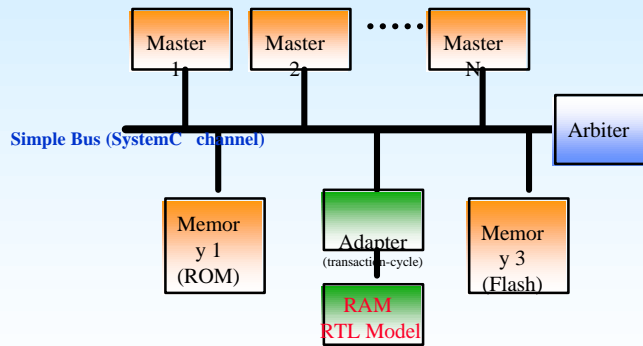
Simple example 1/3

Example adapted from Synopsys's « simple_bus » SystemC model:

- A bus shared by N masters is described at transaction level
 - Some masters perform blocking transfers
 - Some others perform nonblocking transfers (polling)
- Three memories can be accessed as slaves
- One of them is described in VHDL RTL



Simple example 2/3



ModelSim SE VHDL

File Edit Design View Project Run Compare Macro Options Window

100

- top: top(behav)
 - sram_inst0: sram(behavior)
 - sram_img_inst0: sram_img(image)
 - tvm: tvb_tvm_connect(tbv_tvm_connect)
 - tvm: tvb_tvm_connect(tbv_tvm_connect)
 - Package tbvlib
 - Package std_logic_textio
 - Package textio
 - Package std_logic_unsigned
 - Package std_logic_arith
 - Package std_logic_1164
 - Package standard

```

# Welcome to TestBench Development Kit, version 1.0.1.1
#
# Kit Created Fri Mar 8 12:56:18 PST 2002
#
# Installed Wed Mar 27 13:41:55 MET 2002
#
# C++ : Init..
#
# Transaction Recording API Version 1.3.172 (SDL_VERSION_2.0)
run -all
# C++: Starting cosimulation at t=0 ns
#
# tvbTvmT full name: /top/sram_img_inst0
#
# definition name: sram_tvmt
#
# instance name: sram_img_inst0
#
# List_Of_Tasks_In_This_Tvm:
#   sram_comm_task
#
# actualFiber: <not_defined>
# internalFiber: <nnt_defined>
    
```

Library sim

Now: 0 ns Delta: 2

sim/top

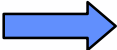
```

mananbolo:output[352] time ./systemc_cosim

SystemC 2.0 --- Oct 29 2001 10:00:44
Copyright (c) 1996-2001 by all Contributors
ALL RIGHTS RESERVED
top.bus :: Registering master Nb 0
top.bus :: Registering master Nb 1
top.bus :: Registering master Nb 2
...
top.master_b_0 :: Average transaction duration : 379.947204 ns
top.master_b_0 :: Average transaction duration : 379.947204 ns
top.master_b_1 :: Average transaction duration : 359.970241 ns
top.master_b_2 :: Average transaction duration : 339.970241 ns
top.master_b_3 :: Average transaction duration : 319.970241 ns
top.master_b_4 :: Average transaction duration : 505.662370 ns
top.master_b_5 :: Average transaction duration : 759.894403 ns
    
```

Simple Example : Performances

- SystemC Only : 18.2 Kcycles / s
- Systemc – VHDL : 2.5 Kcycles/s

 9x slow-down

(To be compared with a full VHDL RTL simulation)

Conclusion- Future developments

Conclusion :

- A free tool allowing SystemC – VHDL cosimulation has been developed
- Interfaces with main VHDL simulators
- Automated code generation : No additional code written by designers
- Needs to be tested in ur design flow

Future directions :

- Verilog support
- Decrease the shared memory bandwidth (observe signals activity, synchronize only when required)
- Perform SystemC-HDL communications at transaction level. Bus Adapters library
- Integration with Cadence Signalscan™ to visualize transactions and ease debug

Questions

Thank you for your attention..

Contacts :

[*Nicolas.tribie@cea.fr*](mailto:Nicolas.tribie@cea.fr)

[*Christian.bernard@cea.fr*](mailto:Christian.bernard@cea.fr)

 **leti**
17 Av Des Martyrs
38000 Grenoble, France