# Specification and Validation for Heterogeneous Multiprocessor SoC

**Ahmed A. Jerraya,**

**System-Level Synthesis Group**
**TIMA Laboratory**
**46 Avenue Felix Viallet**
**38031 Grenoble Cedex France**
**Tel: +33 476 57 47 59**
**Fax: +33 476 47 38 14**
**Email: Ahmed.Jerraya@imag.fr**

# System Specification

- Context: Systems are heterogeneous
  - Abstraction levels: clock-cycle accurate, system state, message and transaction level communication, …
  - Execution models: native execution, model simulation, ISS, …
  - Domains: HW, SW, RF, …, environment.
- Challenges:
  - Specify interconnect for heterogeneous objects
  - Execution model for heterogeneous systems
- Requirements
  - Modular design, separation communication/computation
  - Multi and mixed level Co-simulation

# Outline

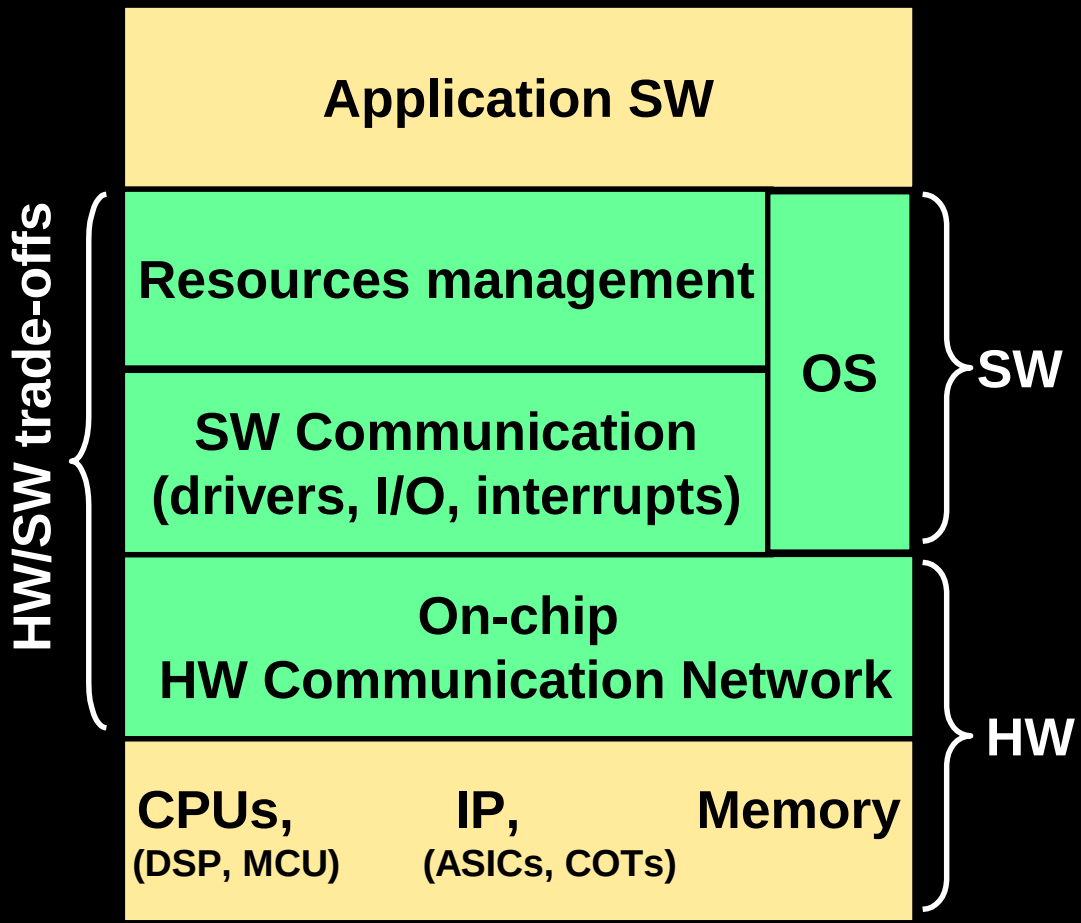
# Multi-Processor System on Chip

- SoC: put on a chip what we used to put on one or several boards: 90% of ASICS in 2003
  - Heterogeneous components (CPU, memory, bus, ASICs, non digital)
  - Focus on architecture design to meet performances. Generic technology TTM, you don't have to reinvent the O every day.
- Multi-processor
  - More than one instruction set processor on chip
  - Complex on-chip HW/SW communication network
- Multi-processor System on chip:
  - Tighter TTM constraints
  - Application-specific architecture
  - Design automation required

# Multi-processor SoC Examples (with Silicon share >> CPUs)

| Components / Application | Data Computation | Control Computation | On Chip Memory | On Chip Communi-cation | Specific Logic | Typical design |
|---|---|---|---|---|---|---|
| Wireless terminal, XDSL | 1 DSP | 1 MCU | > MB | Bridge | > M gates | STEP 1, VDSL (ST) |
| Multimedia | Few DSPs | 1 MCU | >> MB | Network switch, cross bar | < M gates | TRIMEDIA (Philips) |
| Network processor | Many DSPs | Few MCUs | >> MB | On chip network | > M gates | IXPIZDE (INTEL) |
| Game processors | Few DSPs | Few MCUs | >> MB | On chip hierarchical network | >> M gates | Play station |

# System-on-Chip Architectures

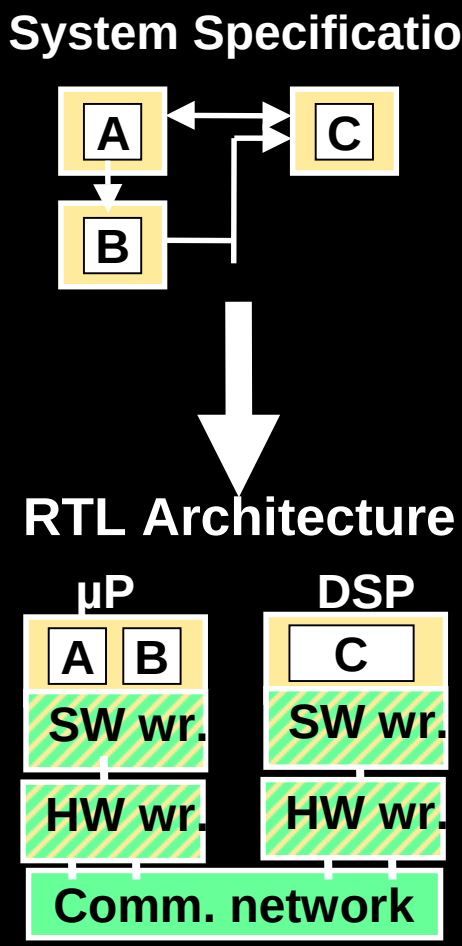



- SoC Architecture:
  - HW Components
  - Application Software
  - HW-SW Communication
    - On Chip network (HW)
    - support Package (SW)
    - Programming Layer (SW)
- Multilevel APIs required
- Design Automation
  - HW component design: OK
  - SW Application design: OK
  - Architecture design: Still need to be invented

# Outline

1. Multiprocessors SoC (MP SoC)
   - 1.1. Multiprocessor SoC
   - 1.2. Multiprocessor SoC design
   - 1.3. This Course
2. Specification and validation of electronic systems
   - 2.1. Basic concepts
   - 2.2. Specification languages
   - 2.3. Heterogeneous systems modeling and validation
3. COLIF: A Design Model for MP SoCs
   - 3.1. COLIF: the Meta-model and the external syntax
   - 3.2. Mixed and multilevel model execution
4. A VDSL design example
   - 4.1. The application
   - 4.2. The design process
5. Summary

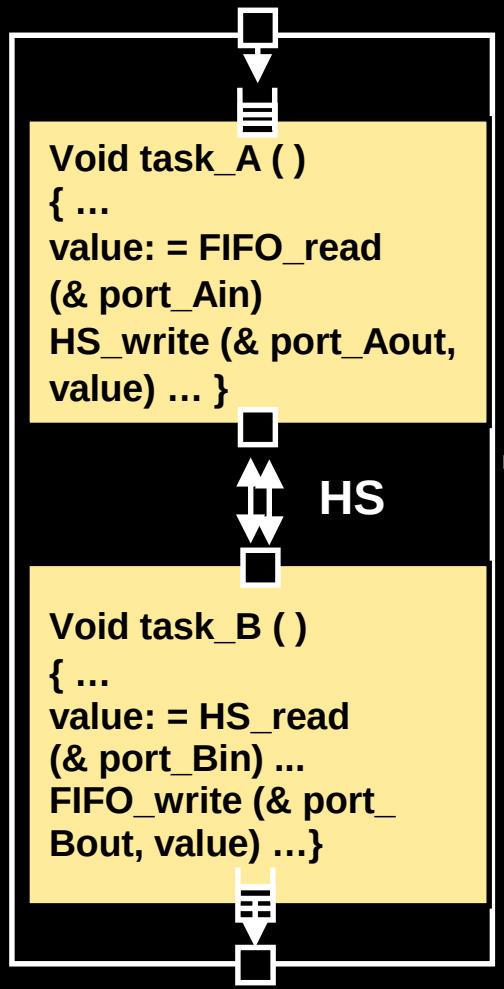
# Multi-Processor SoC Design




- Design: produce an RTL multi-processor architecture and the corresponding support software from system specification
- Key decisions:
  - Multiple vs. single OS
  - Static vs. dynamic allocation of tasks
  - Existing OS vs. Generated OS
  - Architecture platform vs. IP assembling
- System specification
  - Abstract components/functions
  - Abstract interconnect model
  - Architecture design decisions
- Design steps
  - Architecture definition & specilization
  - Application mapping
  - HW/SW communication design
  - Validation at different design steps

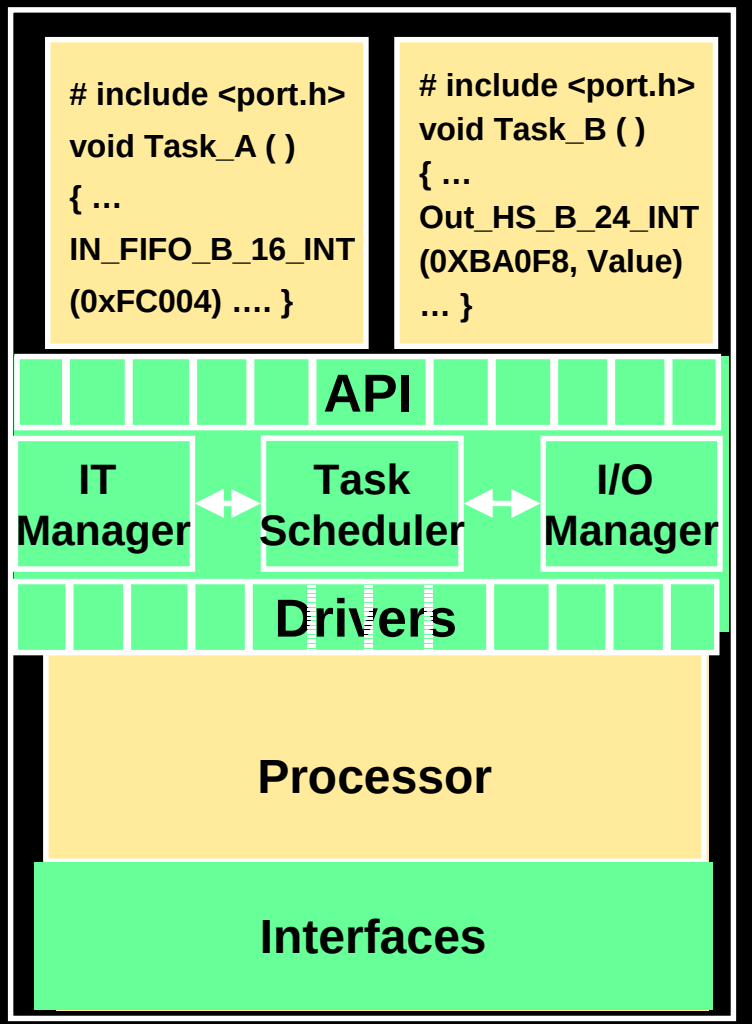
# Hardware/Software Communication Design

- Design of HW and SW communication layers to model abstract interaction between tasks/modules and with external world




Macro-architecture (e.g. SystemC)


Micro-architecture (e.g. Low-level C/VHDL)



# Outline

1. Multiprocessors SoC (MP SoC)
   - 1.1. Multiprocessor SoC
   - 1.2. Multiprocessor SoC design
   - 1.3. This Course

2. Specification and validation of electronic systems
   - 2.1. Basic concepts
   - 2.2. Specification languages
   - 2.3. Heterogeneous systems modeling and validation

3. COLIF: A Design Model for MP SoCs
   - 3.1. COLIF: the Meta-model and the external syntax
   - 3.2. Mixed and multilevel model execution

4. A VDSL design example
   - 4.1. The application
   - 4.2. The design process

5. Summary

# Application-Specific Multi-Processor SoC

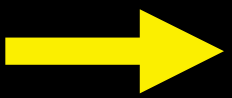
## Summer school

sponsored by IEEE Circuits and Systems Society and EDAA

| Monday 9th July | Tuesday 10th July | Wednesday 11th July | Thursday 12th July | Friday 13th July |
|---|---|---|---|---|
| Welcome | Building Systems on a Chip with Trimedia technology *K. Vissers, TriMedia Technologies*<br><br>From Applications to Multi-Processor DSP Architectures *P. Pirsch, U. Hannover*<br><br>Energy-efficient design and manage-ment of SoCs *G. De Micheli,Stanford U* | From a distributed embedded RTOS to a pragmatic framework for multi-core SoC *E. Verhulst, Eonic Solutions*<br><br>Task-level run-time scheduling approach for dynamic multi-media systems. *F. Catthoor, IMEC*<br><br>Modeling real-time systems *J. Sifakis, Verimag* | Challenges in Network Processor Architectures and Embedded S/W Tools *P. Paulin, STMicroelectronics F. Karim, STMicroelectronics*<br><br>Adaptive EPIC Processors and Compilation Techniques *K. Palem, Georgia Tech* | Communication architectures for deep-submicron VLSI Systems *J. van Meerbergen, Philips*<br><br>System Architectures: Hardware or Software dominant ? A Case Study: xDSL modems *M. Genoe, Alcatel* |
| Specification and Validation for Heterogeneous Multiprocessor SoC *A. Jerraya, TIMA*<br><br>Platform based design of embedded systems-on-chip *W. Rosenstiel, U. Tuebingen & FZI* | Real-Time Operating Systems: Principles and a Case Study *K. Shin, U. Michigan*<br><br>RTOS for Embedded Systems and SoC *M. Potkonjak, UCLA*<br><br>Real-Time Inter-Pro-cessor Synchronization Algorithms *H. Takada, Toyohashi UT* | The Architecture of Multiprocessor Systems on a Chip *T. Mudge, U. Michigan*<br><br>SOC Multiprocessor Architecture and Modeling *R. Ernst, TU Braunschweig*<br><br>Architectural challenges and opportunities for systems on a chip *B. Rau, Hewlett-Packard* | Multiprocessor SoCs for Video Processing *W. Wolf, Princeton U.*<br><br>Configuring the Jazz VLIW-DSP Core for Application Specific Requirements *O. Levia, Improv Systems*<br><br>Static scheduling for embedded systems *L. Lavagno, U. Udine* | System on Chip: Embedded Test Strategies *Y. Zorian, LogicVision*<br><br>Architecture and Imple-mentation of Application-Specific Multi-processor SOCs for Digital TV (DTV) and Media-Processing Applications, *S. Dutta, Philips*<br><br>Testing Future System-on-Chips: Challenges and Emerging Techniques *S. Dey, U. California, San Diego* |

# Outline

1. Multiprocessors SoC (MP SoC)
    - 1.1. Multiprocessor SoC
    - 1.2. Multiprocessor SoC design
    - 1.3. This Course



2. Specification and validation of electronic systems
    - 2.1. Basic concepts
    - 2.2. Specification languages
    - 2.3. Heterogeneous systems modeling and validation

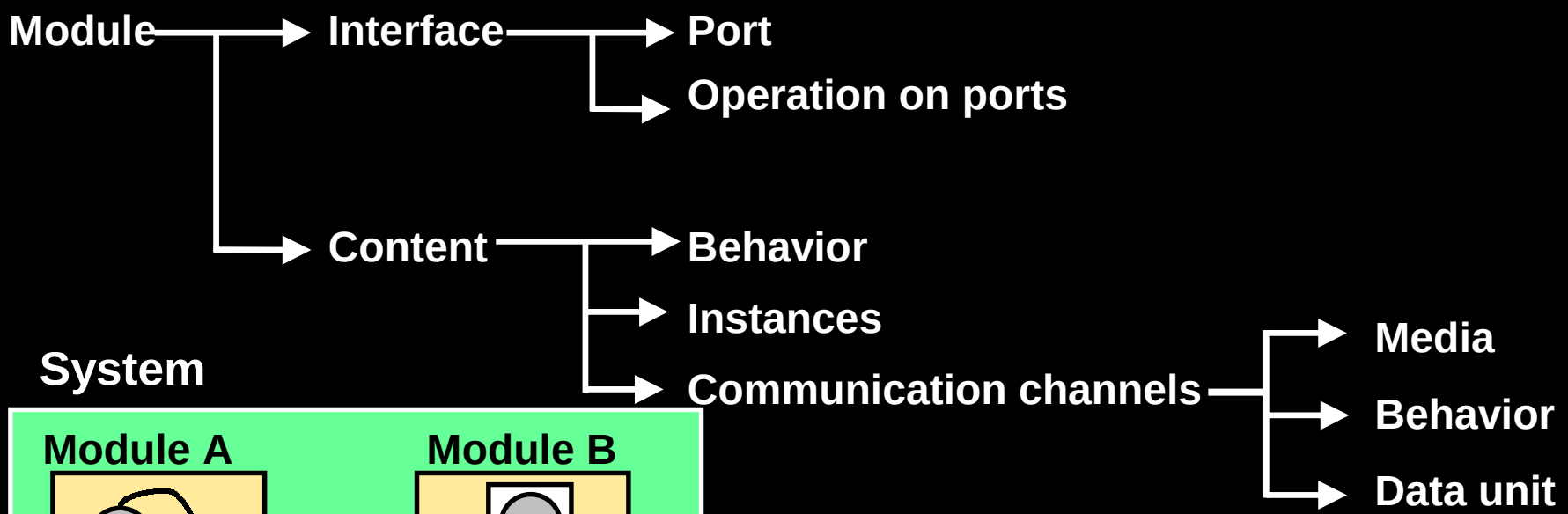
3. COLIF: A Design Model for MP SoCs
    - 3.1. COLIF: the Meta-model and the external syntax
    - 3.2. Mixed and multilevel model execution

4. A VDSL design example
    - 4.1. The application
    - 4.2. The design process

5. Summary

# Electronic System Specification

- **Basic Model: a set of hierarchically interconnected modules representing an abstract architecture**
- **Basic concepts:**




**Different abstraction levels for behavior and communication**

# Abstraction levels in behavior

| | Behavioral model | | |
|---|---|---|---|
| Abstraction level | Timing Unit | Typical language | Typical model |
| Untimed | Partial order | CSP, SDL | Communicating processes |
| Superstate | Computation/ control step | VHDL, SystemC, Esterel | Synchronous scheduling |
| RTL | C/K cycle | VHDL, SystemC, Verilog | Interconnected FSMs |
| Physical level | Physical Time (Delays) | VHDL/Verilog | Gates |

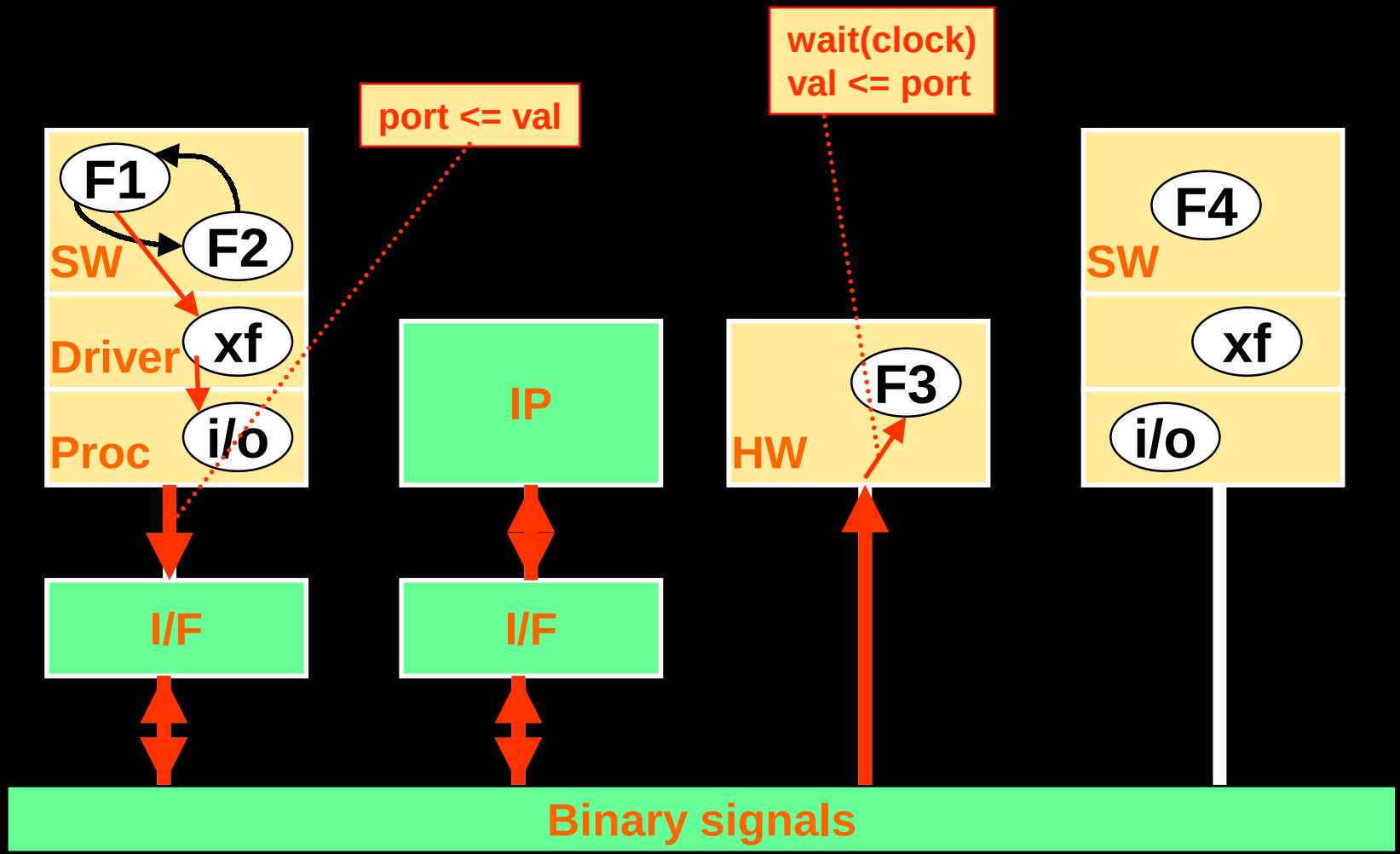
# Abstraction levels in communication

| | Communication Model | | |
|---|---|---|---|
| Abstraction level | Media | Typical Model | Typical communication primitive |
| Client server | Abstract Network | CORBA / UML | Print (file, network) |
| Functional | Active Channel | SDL | Send (file, disk) |
| Macro architecture | Logical connection | VHDL, SystemC | Write (Data, Port) Wait until x = y |
| RTL (Micro arcitecture | Physical connection | VHDL, SystemC, Verilog | Set (Value, Port) Wait (clock) |

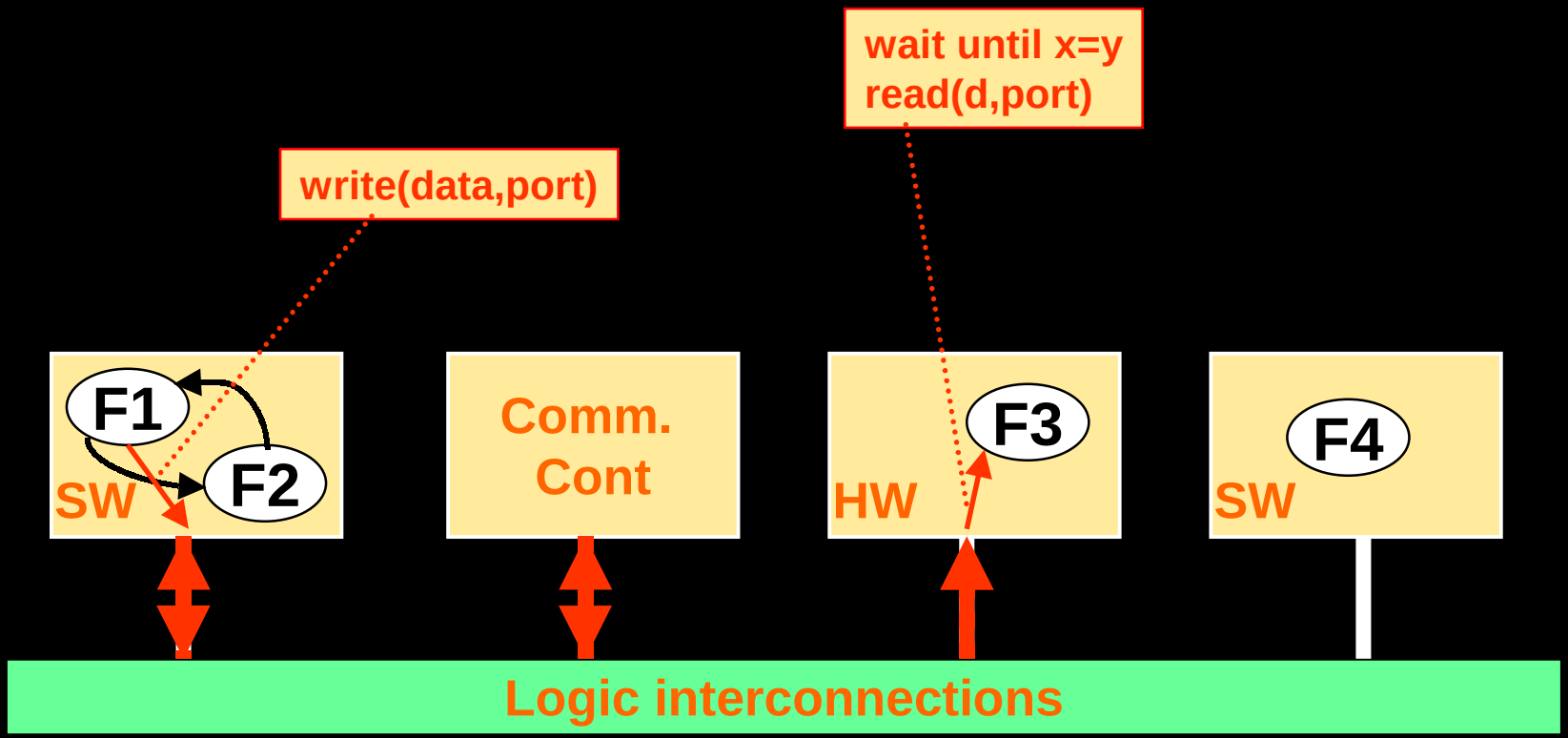
# Modeling Concepts through the Abstraction Level

| Object \ Abstraction level | | C-S level | Functional | Macro-Architecture | RTL |
|---|---|---|---|---|---|
| Module | | Module | Module | Module | Module |
| Interface | Port | Network Access (SAP) | Channel access | Logical port | Physical port |
| | Operation | Service request | Send/Receive to identified process | Read/Write Data | Set/Reset bits |
| Content | Behavior | Concurrent Objects | Partially Ordered Transactions | Computation/ control steps | Cycle-true computation |
| | Instance | Instance | Instance | Instance | Instance |
| | Communication Channel<br>- Media<br>- Behavior<br>- DATA | - Abstract Network<br>- Routing<br>- Request | - Active Channel<br>- Protocol conversion<br>- Generic data transmission | - Abstract wires<br>- Driver-level protocol<br>- Fixed data type | - Physical wires<br>- Transmission<br>- Fixed data representation |

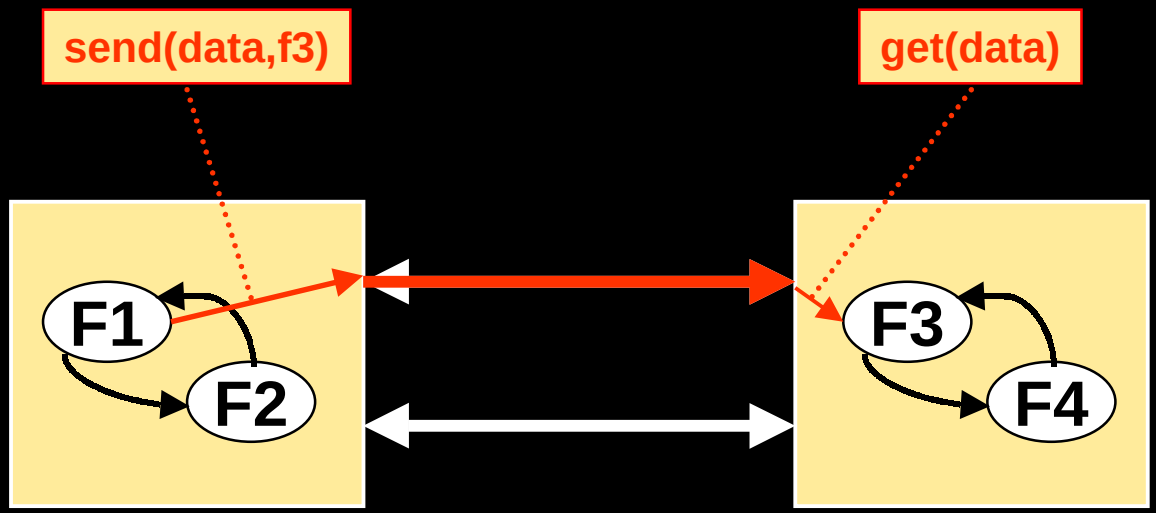
# Register transfer level



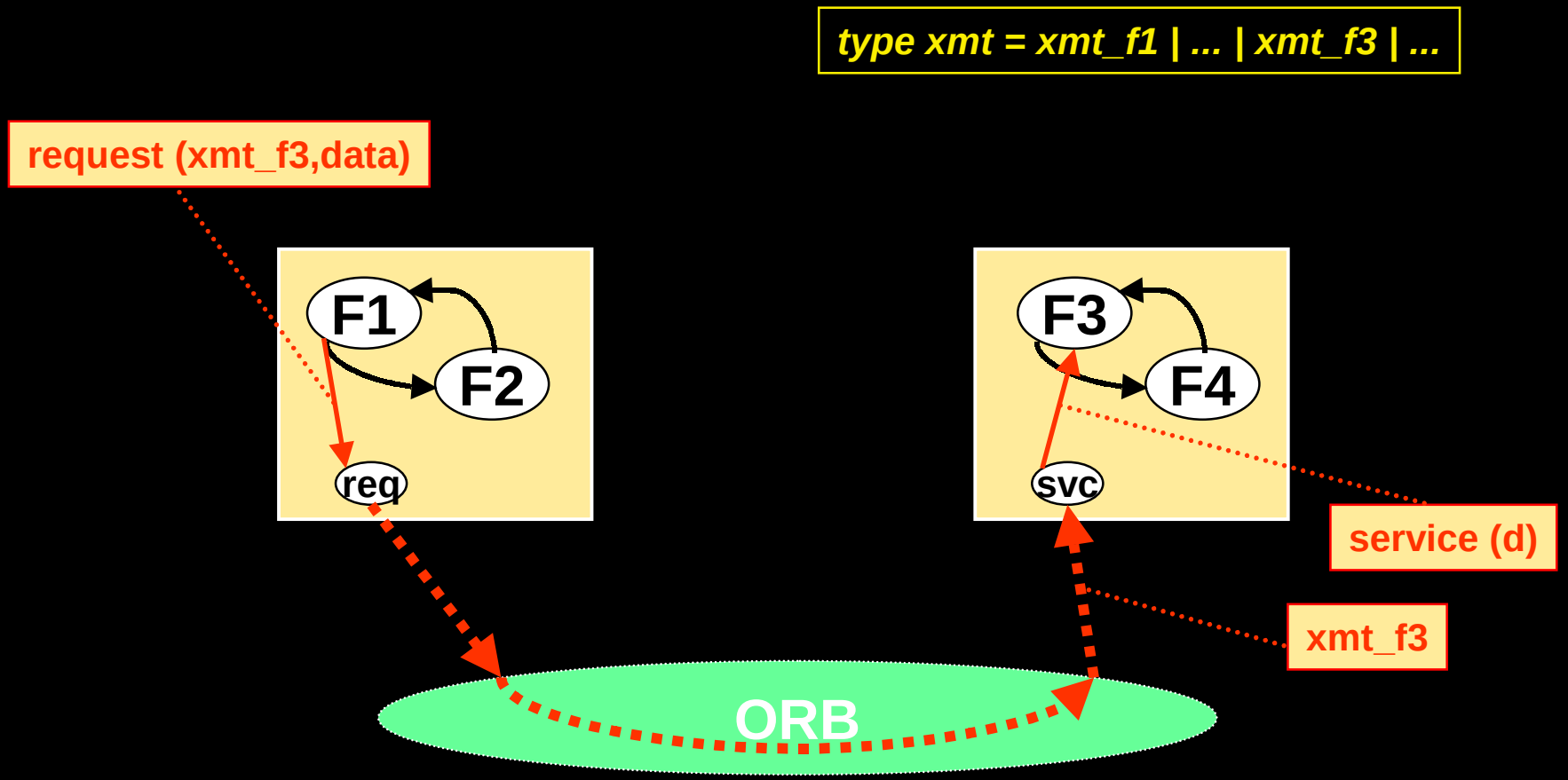
# Driver level



# Message level



# Service level



# Outline

1. Multiprocessors SoC (MP SoC)
   - 1.1. Multiprocessor SoC
   - 1.2. Multiprocessor SoC design
   - 1.3. This Course
2. Specification and validation of electronic systems
   - 2.1. Basic concepts
   - 2.2. Specification languages
   - 2.3. Heterogeneous systems modeling and validation
3. COLIF: A Design Model for MP SoCs
   - 3.1. COLIF: the Meta-model and the external syntax
   - 3.2. Mixed and multilevel model execution
4. A VDSL design example
   - 4.1. The application
   - 4.2. The design process
5. Summary

# Welcome to the Jungle of specification languages

- Which purpose: Modeling, Simulation, Synthesis
- Which abstraction: Communication, Behavior
- Which computation model:
  - Task: Predictable execution time, Run to completion, Interactive with internal state (VHDL, SDL)
  - Control model: State-based or control-driven (SDL, ESTEREL) Data-driven (LUSTRE, COSSAP, Matlab)
  - Concurrency (Composition): Pure Synchronous (StateCharts, Esterel, Lustre), Synchronous Clocked ( SystemC, VHDL, StateCharts), Asynchronous (SDL), Single thread (C, C++)
  - Communication behavior: Zero Delay (StateChart, VHDL), Non-zero delay (SDL, Corba)
- Which community

# Description Languages: Different Communities

- Methods: UML, OMT, SART, ...
- ADL: WRITE, RAPID, IDL ...
- RT systems:
  - Synchronous: LUSTRE, ESTEREL, StateCharts, …
  - Asynchronous: SDL, Objectime, ...
- System modeling:
  - Universal: Matlab, Matrixx, …
  - DSP: Cossap, SPW
- Semiconductor:
  - HDL: VHDL, Verilog, e,
  - C/C++ extensions: SystemC, SpecC, ...

**None of the existing languages covers all system-level concepts at all abstraction levels**

# Missing Concepts in Specification Languages

| Concept \ Language | | | SystemC .9 HDL | SC 1.0 Cossap, SPW | Real Time Synchronous | Real Time, Asynchronous UML, ADLs |
|---|---|---|---|---|---|---|
| Module | | | OK | OK | OK | OK |
| Interface | Port | | Abstract ports | Generic data port | Active ports | Physical ports |
| | Operation | | Protocol independent operations | Data Generic Operation | Non fixed delay operation | - Regular data stream operations<br>- Detailed protocols |
| Content | Process Task | | Synchronized multiple task | Data dependent computation | Non regular cycle free computation | - Cycle-true model<br>- Implementation related operation |
| | Instance | | OK | OK | OK | OK |
| | Communication Channel | Media | Abstract Channels | Active Channels | Hierarchical & distributed communication | Physical signal buses |
| | | Behavior | Non transmission behavior | Protocol Conversion | Other than broadcasting | Regular fixed data streams |
| | | DATA | Generic data types | Generic data types | Generic data types | Fixed data representation |

# System-level specification trends

- extend an existing language: N2C, SpecC, SystemC
  - + fast executable model for verification
  - - task/comm. model is bound to the simulation model
- create a new language: Rosetta
  - + formal verification
  - - inflexible communication models
- Semantic models: OVI-SRM, VSIA
  - + language neutral approach
  - - peer-to-peer comm. and run-to-completion tasks
  - - complex refinement process
- “Meta-models”: GSRC, COLIF “abstract syntax”
  - syntax and semantic independent

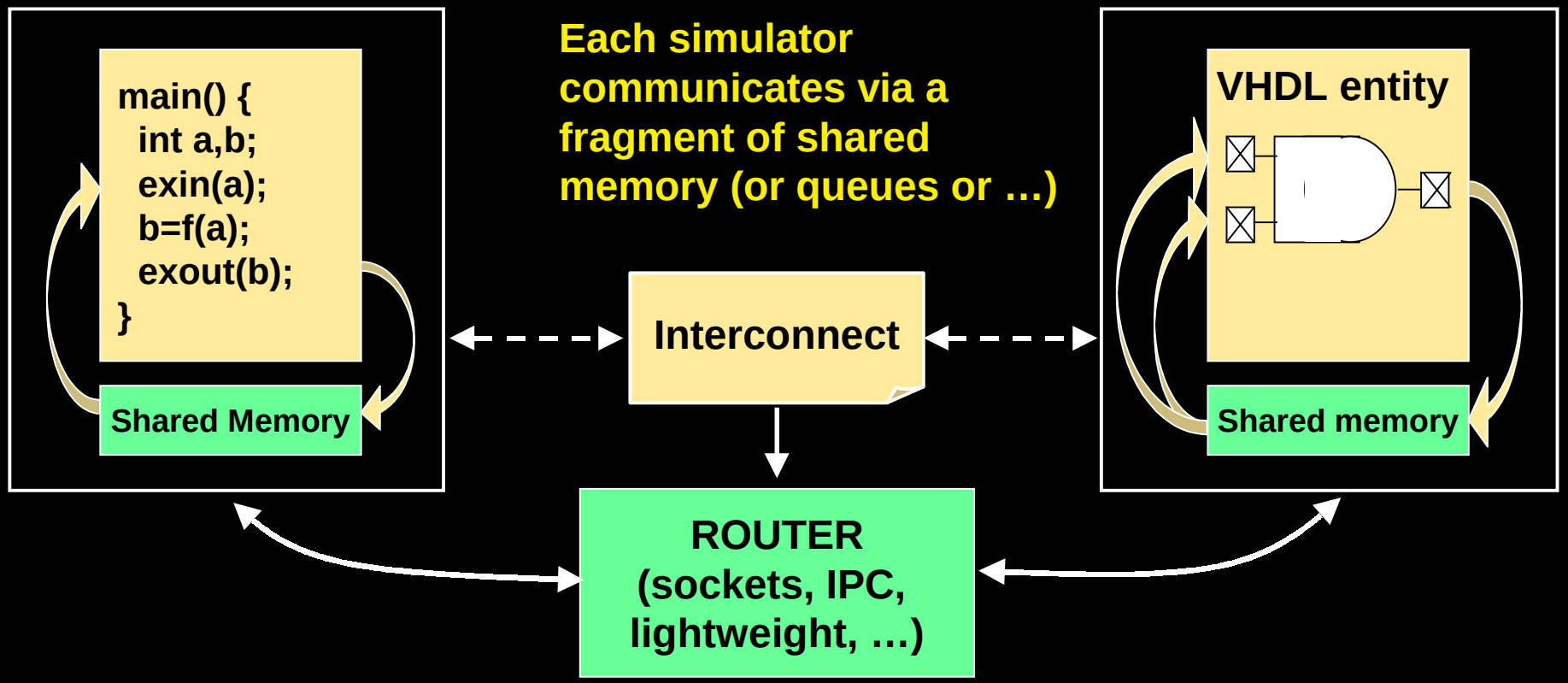
# Outline

1. Multiprocessors SoC (MP SoC)
   - 1.1. Multiprocessor SoC
   - 1.2. Multiprocessor SoC design
   - 1.3. This Course
2. Specification and validation of electronic systems
   - 2.1. Basic concepts
   - 2.2. Specification languages
   - 2.3. Heterogeneous systems modeling and validation
3. COLIF: A Design Model for MP SoCs
   - 3.1. COLIF: the Meta-model and the external syntax
   - 3.2. Mixed and multilevel model execution
4. A VDSL design example
   - 4.1. The application
   - 4.2. The design process
5. Summary

# Heterogeneous Models

- Components execution models
  - Hardware: Gates, RTL, Functional, Client Server,
  - Software: CA/ISS, IA/ISS, Native Bus Functional CA, Native with OS, Functional, Client Server
  - Functional Architecture, before partitioning
  - Environment, Multiple domains (Mechanical, Optical, …), Multiple computation models
- Full System Execution model
  - Different concepts ➨ co-simulation
  - Multiple computation models
    - Separation between computation and communication
    - Single or several languages

# Example: C - VHDL Co-simulation




The router secures the coherence between several memories via sockets.

# Heterogeneous System approaches

- SystemC
  - Module or channel wrapper usage
  - Modules/channels with multiple abstraction levels
  - SystemC provides primitives (e.g. interface concept) and ideas (e.g. BCASH) to design wrappers.
- VADel [Cesario] Module wrapper generation
- Bus Functional Model [Séméria et. al., ASPDAC2001]
- Modeling community: ADL, RAPID, CORBA based

# Summary of system specifications

- Specification languages
  - Few concepts
  - Many from different communities
  - Specific to application domain
- Key issues for MP SoC
  - Specify interconnect for heterogeneous objects
  - Execution model for heterogeneous systems
- Requirements
  - Modular design, separation communication/computation
  - Multi and mixed level Co-simulation

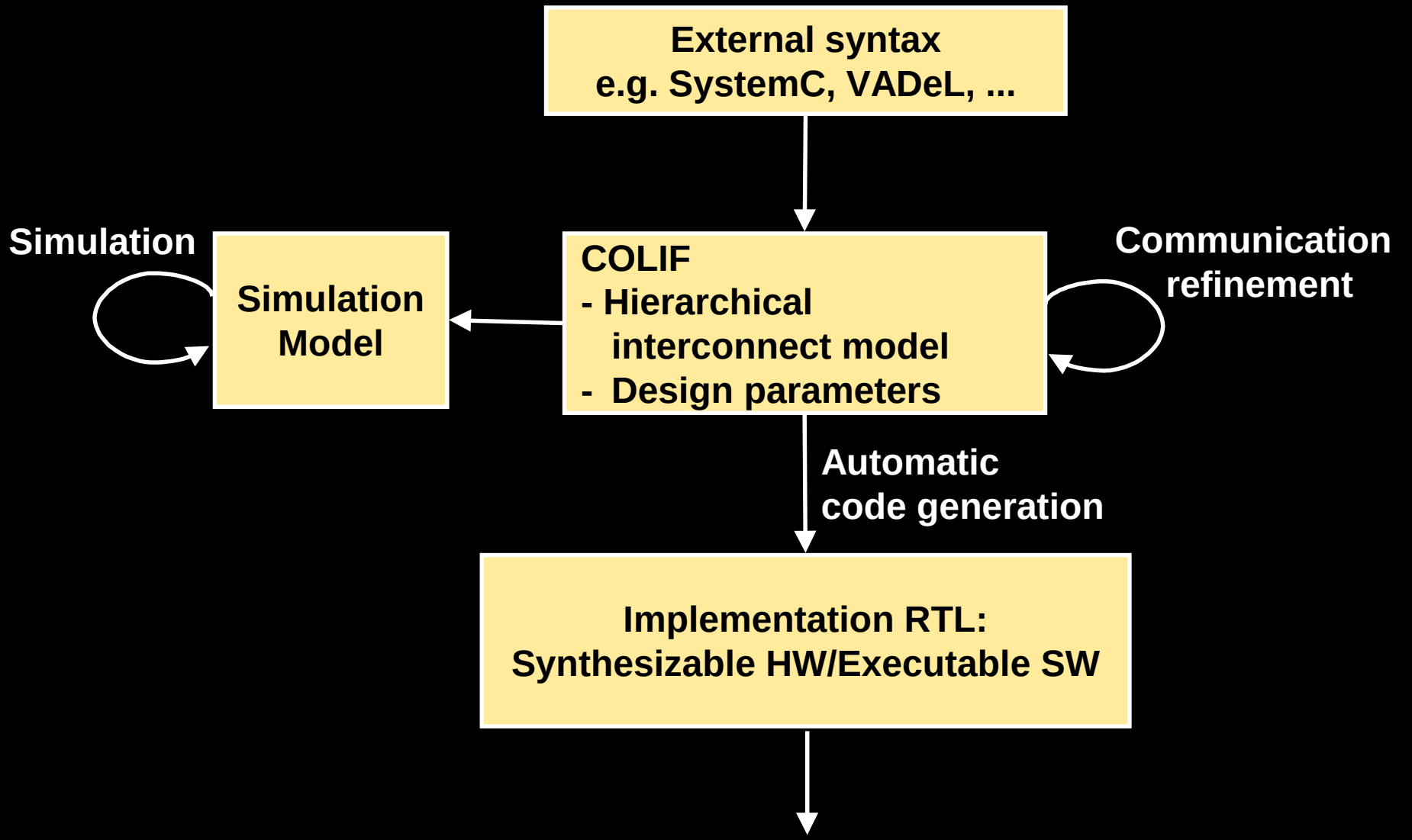
# Outline

1. Multiprocessors SoC (MP SoC)
    - 1.1. Multiprocessor SoC
    - 1.2. Multiprocessor SoC design
    - 1.3. This Course
2. Specification and validation of electronic systems
    - 2.1. Basic concepts
    - 2.2. Specification languages
    - 2.3. Heterogeneous systems modeling and validation
3. COLIF: A Design Model for MP SoCs
    - 3.1. COLIF: the Meta-model and the external syntax
    - 3.2. Mixed and multilevel model execution
4. A VDSL design example
    - 4.1. The application
    - 4.2. The design process
5. Summary

# COLIF: A Design Model for MP SoCs Heterogeneous Specification

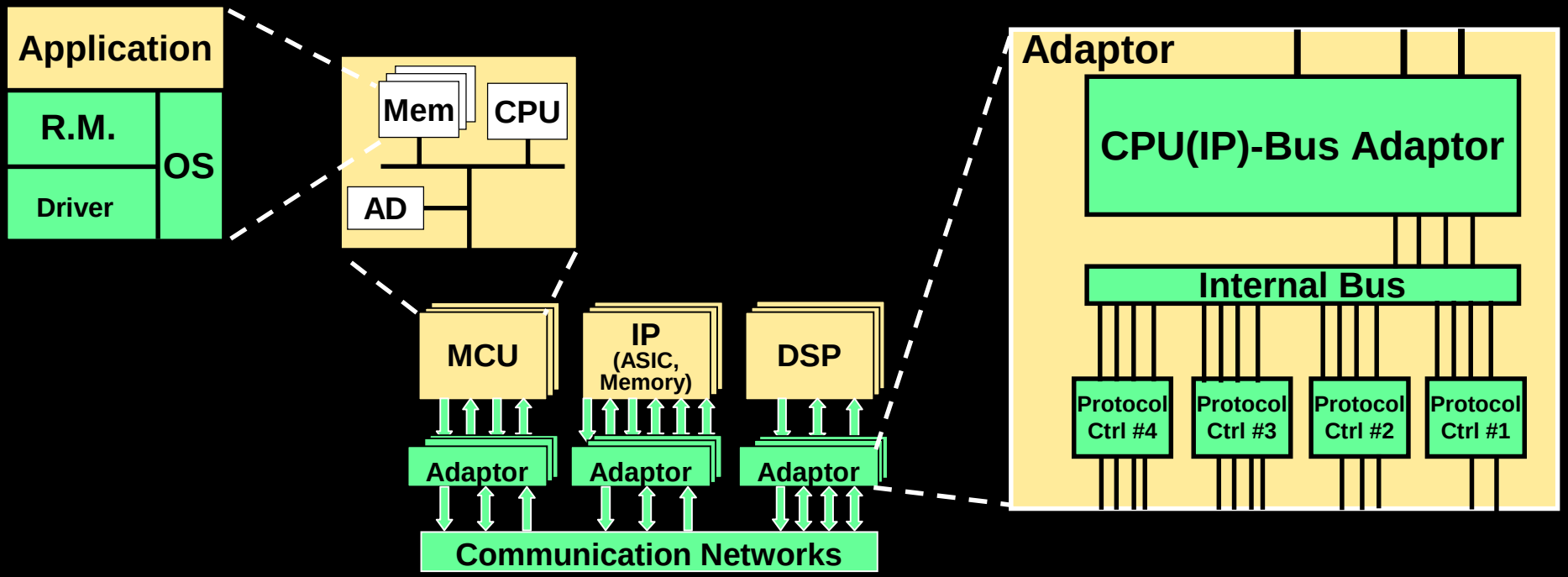
- Basic model: a set of hierarchically interconnected modules representing an abstract architecture
- Basic concepts: Module, Interface (internal, external), Content
- Accommodate both HW-SW models and Co-simulation
- Execution model through automatic wrapper generation
- External Syntax: VADeL, a SystemC extension
- Internal Syntax: XML

# COLIF: An Internal Model for Mixed & Multi-level Refinement



# Target Architecture

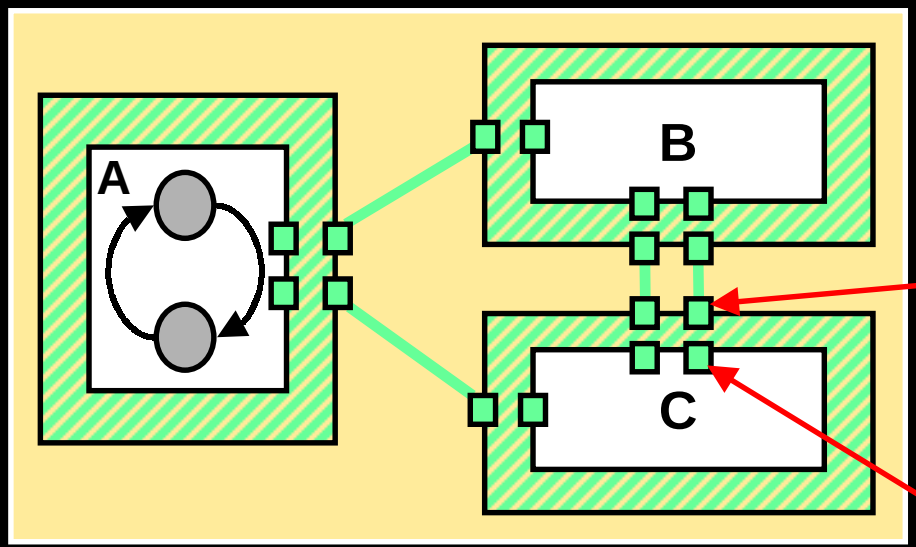
- Dissociate Components and communication network
  - HW: Requires communication co-processors (wrapper, bridge)
  - SW: Multiple Application specific RTOS



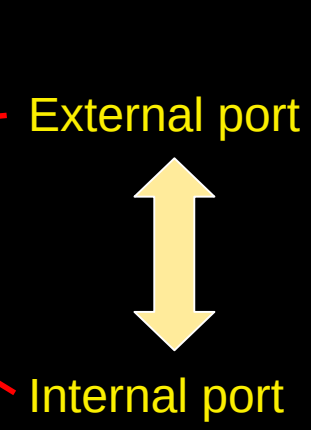

- Enable both Automatic Design and Co-simulation
- Cover both  Hardware and Software
- Systematic Assembling of heterogeneous existing blocs

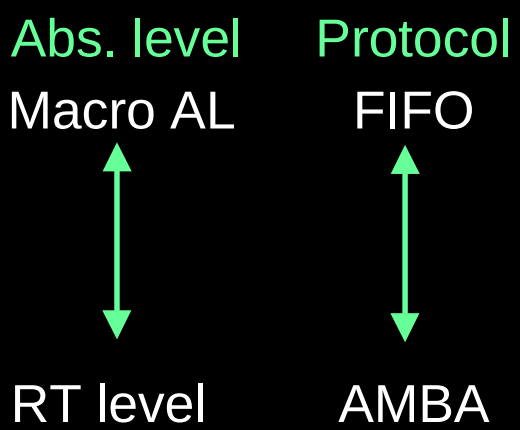
# Heterogeneous System Specification

- **Basic model: a set of hierarchically interconnected modules**
- **Basic concepts:**
  - **Virtual Module**
    - **Interface, set of ports (internal, external)**
    - **Content (**Tasks / Instances + Communication channels)




System Specification

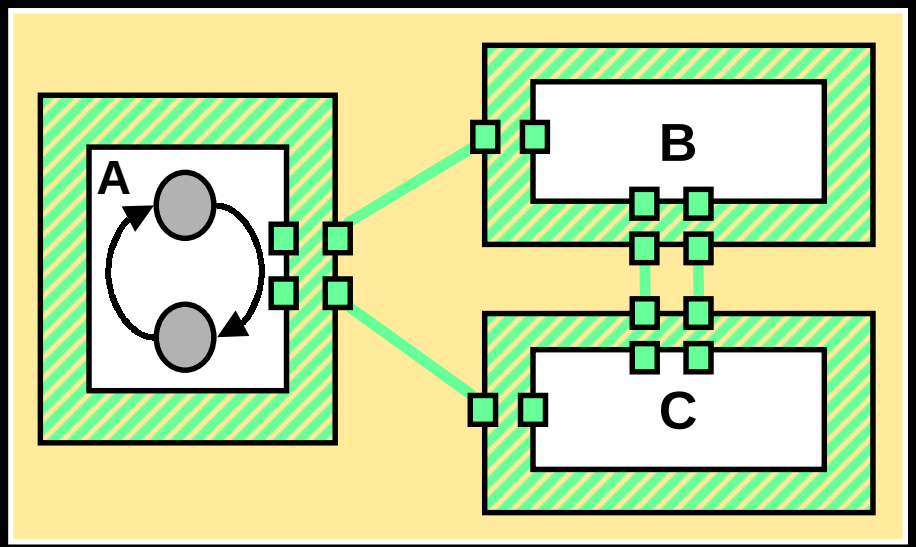


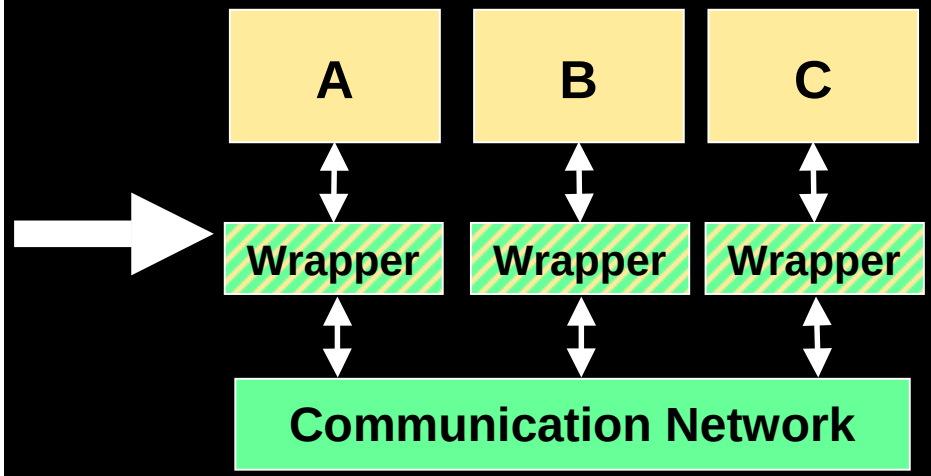

**Not executable due to the difference btw. internal and external ports**

# Heterogeneous System Specification

- Basic model: a set of hierarchically interconnected modules
- Basic concepts:
  - Virtual Module
    - Interface, set of ports (internal, external)
    - Content (Tasks / Instances + Communication channels)




System Specification

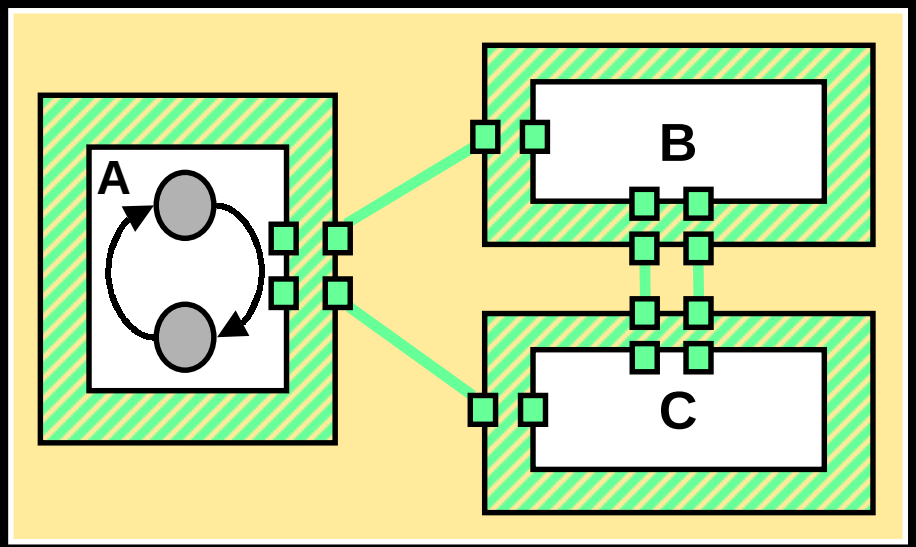



System Architecture

**Executable/implementable by the generated wrappers**

# Heterogeneous System Specification

- **Basic model: a set of hierarchically interconnected modules**
- **Basic concepts:**
  - **Virtual Module**
    - **Interface, set of ports (internal, external)**
    - **Content (**Tasks / Instances + Communication channels)



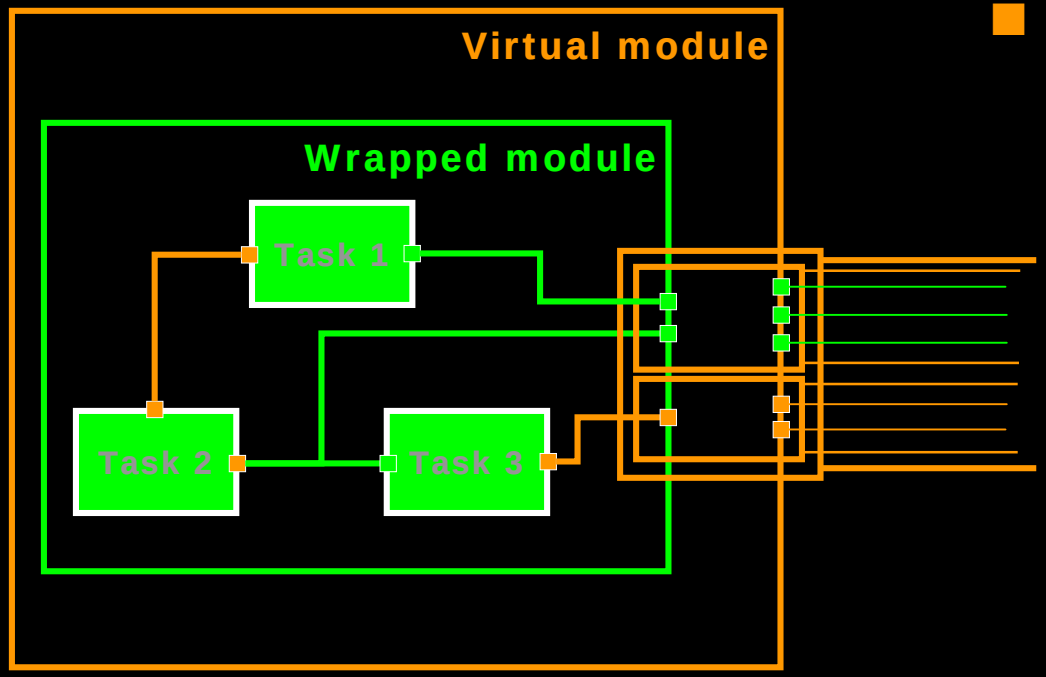

**System Specification**

**Abstraction levels of communication**

- Service: client-server, e.g. CORBA
- System: e.g. send/receive in SDL
- Macro Architecture: e.g. FIFO
- Micro Architecture (RT level): e.g. AMBA

# VADeL: a SystemC extension for COLIF execution

- VADeL: Virtual Architecture Description Language

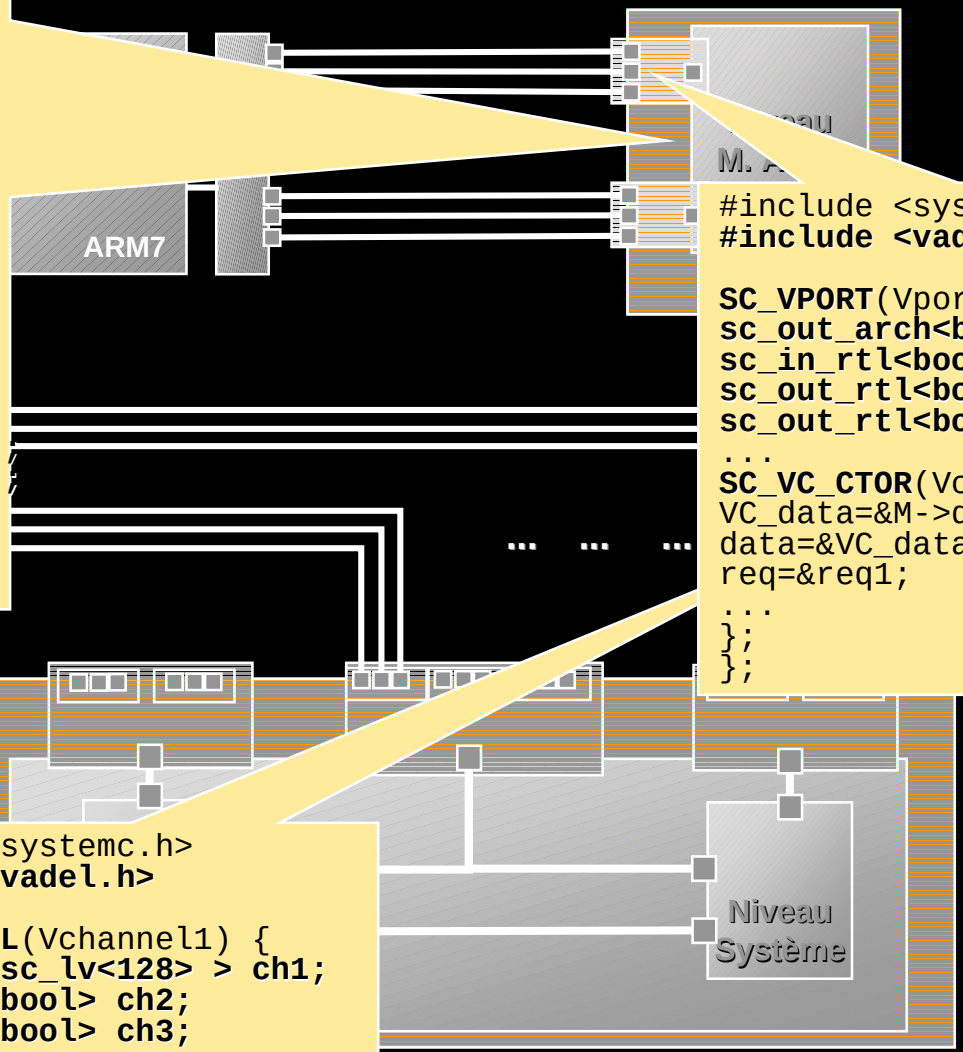



- VADeL
- SystemC

- Embedded processor in VADeL
  - Virtual Module annotated with implementation parameters
  - SystemC modules for each task or IP
  - Virtual Ports for Abstraction-level /Language/Protocol adaptation
  - Virtual nets to group related SystemC signals

# VADeL - Mixed level specification

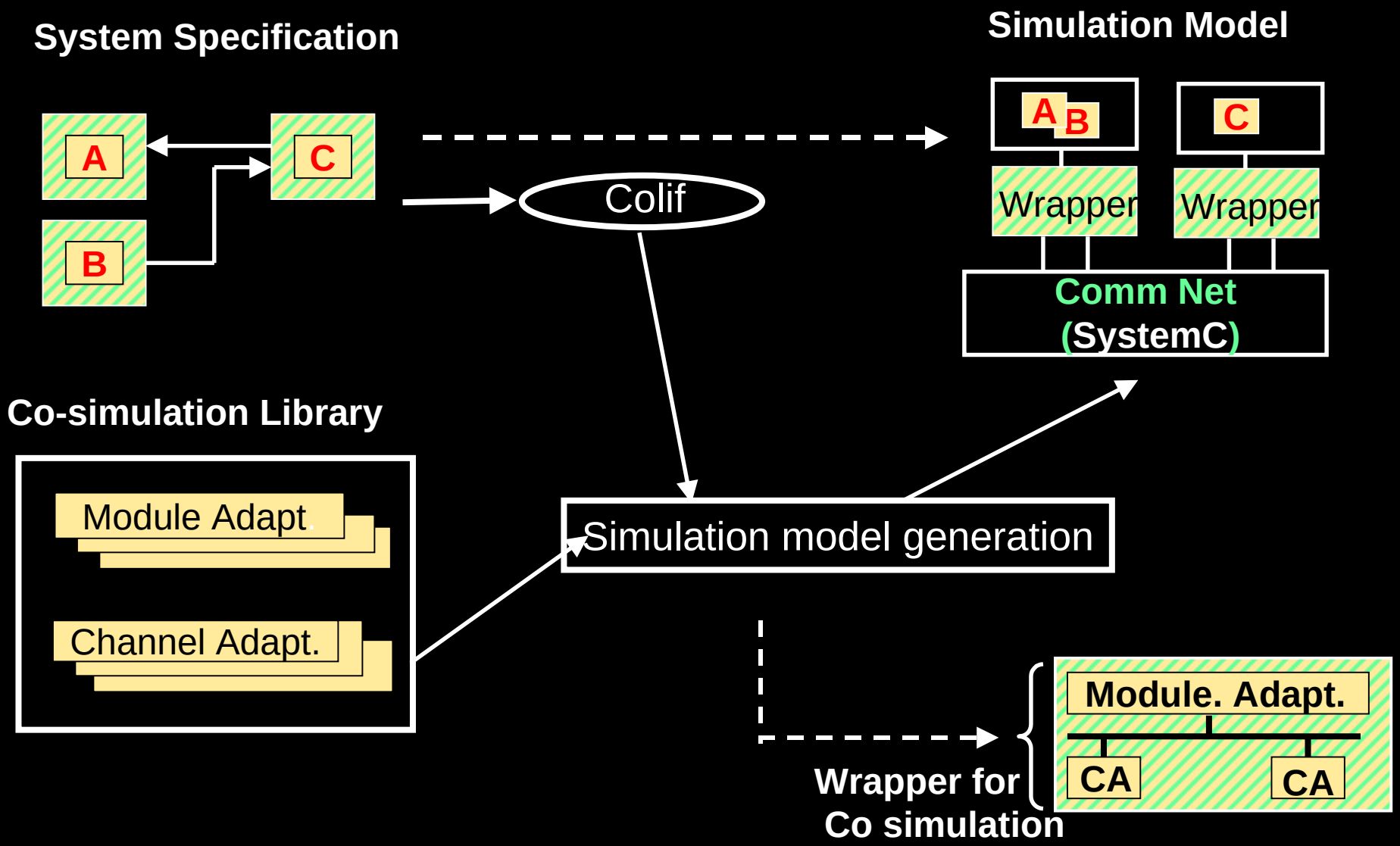
```
#include <systemc.h>
#include <vadel.h>
#include <Vport1.h>
...
#include<token2.h>

SC_VCOMPONENT(VCtoken) {
sc_out_rtl<bool> VC_data1;
sc_out_rtl<bool> req1;
sc_out_rtl<bool> ack1;
...
Vport1* Vp1;
Vport2* Vp2;
...
token* token2;
SC_VC_CTOR(VCtoken) {
token1=new token("token2");
Vp1=new Vport1("Vp1", &token2);
Vp2=new Vport2("Vp2", &token2);
...
};
};
```




```
#include <systemc.h>
#include <vadel.h>

SC_VPORT(Vport1) {
sc_out_arch<bool>* VC_data;
sc_in_rtl<bool>* ack;
sc_out_rtl<bool>* data;
sc_out_rtl<bool>* req;
...
SC_VC_CTOR(Vctoken, M) {
VC_data=&M->data;
data=&VC_data1;
req=&req1;
...
};
};
```

```
#include <systemc.h>
#include <vadel.h>

SC_VCHANNEL(Vchannel1) {
sc_ch_rtl<sc_lv<128> > ch1;
sc_signal<bool> ch2;
sc_signal<bool> ch3;

SC_VC_CTOR(VCtoken) {
};
};
```

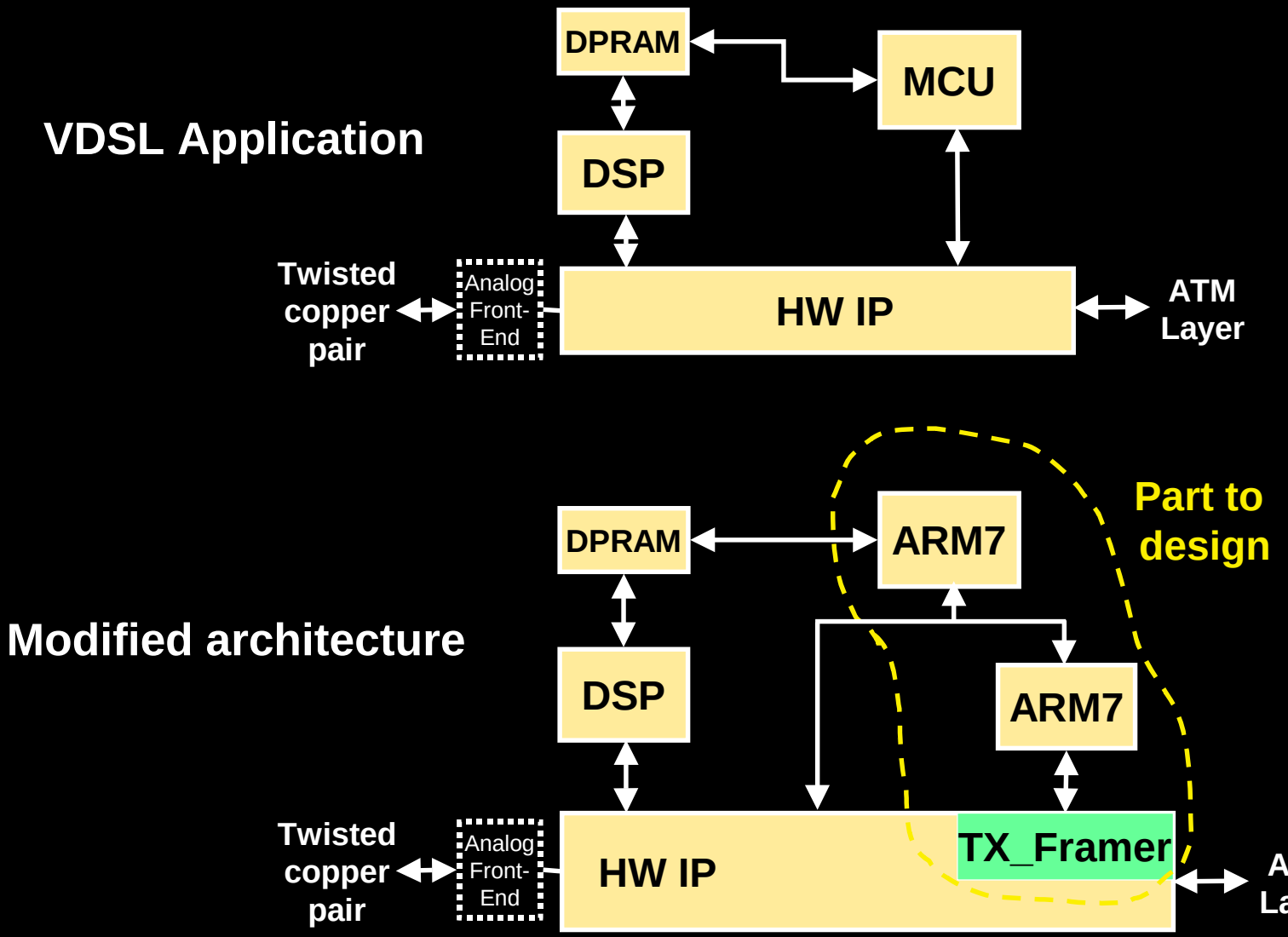
# Executable models generation



# Outline

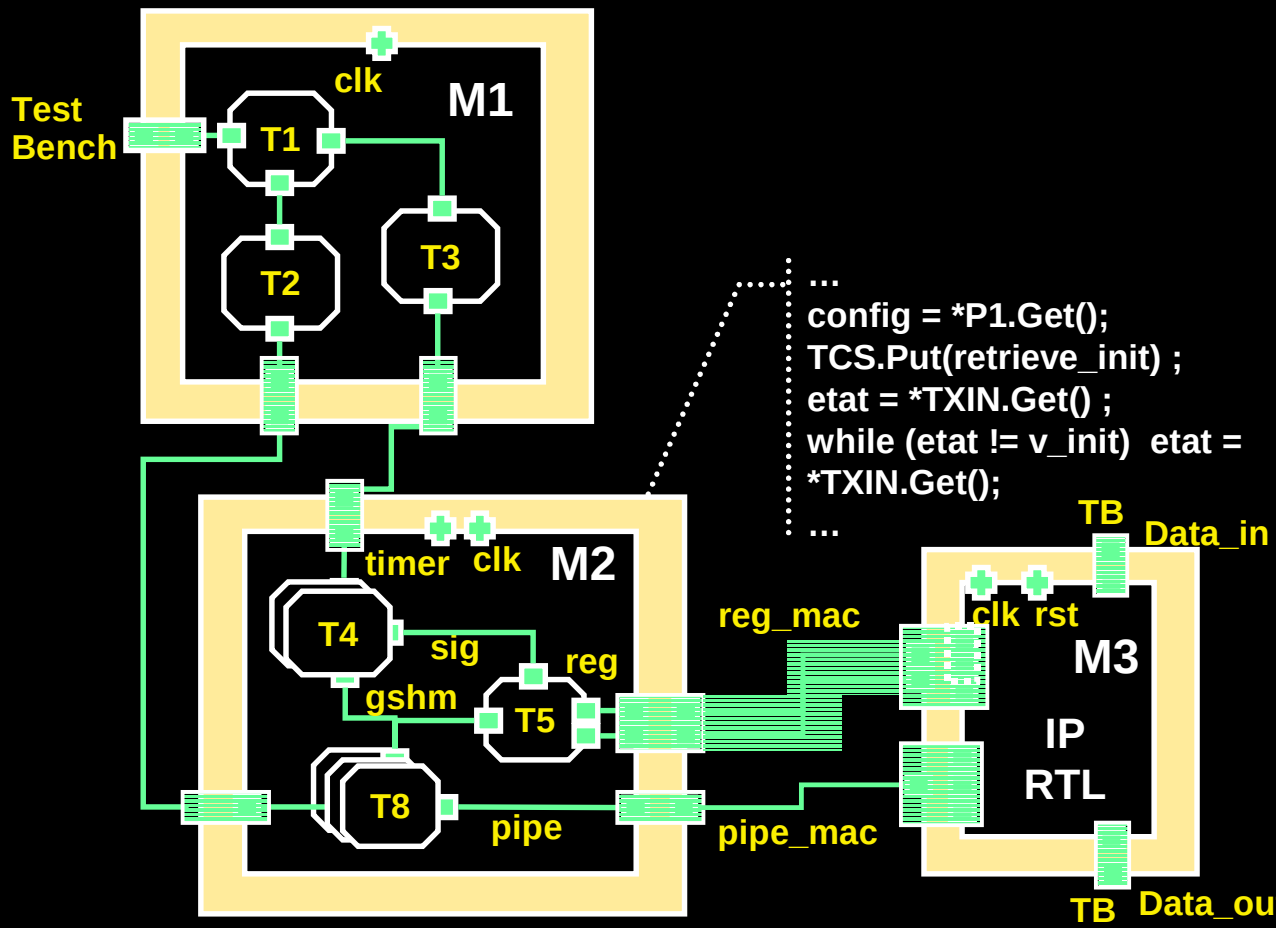
1. Multiprocessors SoC (MP SoC)
    - 1.1. Multiprocessor SoC
    - 1.2. Multiprocessor SoC design
    - 1.3. This Course
2. Specification and validation of electronic systems
    - 2.1. Basic concepts
    - 2.2. Specification languages
    - 2.3. Heterogeneous systems modeling and validation
3. COLIF: A Design Model for MP SoCs
    - 3.1. COLIF: the Meta-model and the external syntax
    - 3.2. Mixed and multilevel model execution
4. A VDSL design example
    - 4.1. The application
    - 4.2. The design process
5. Summary

# Demo: VDSL Design Through Systematic HW/SW Assembly of IP

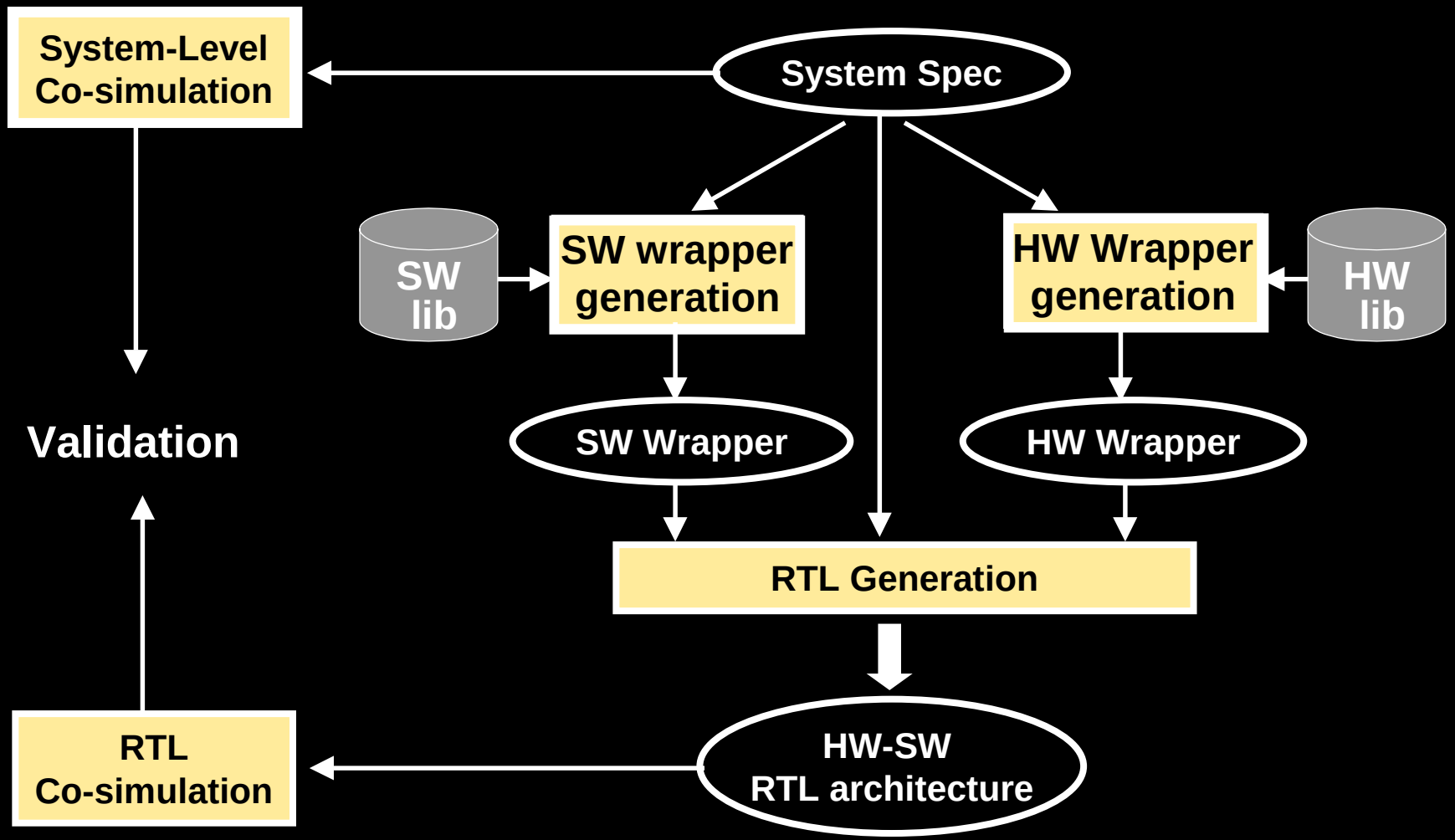


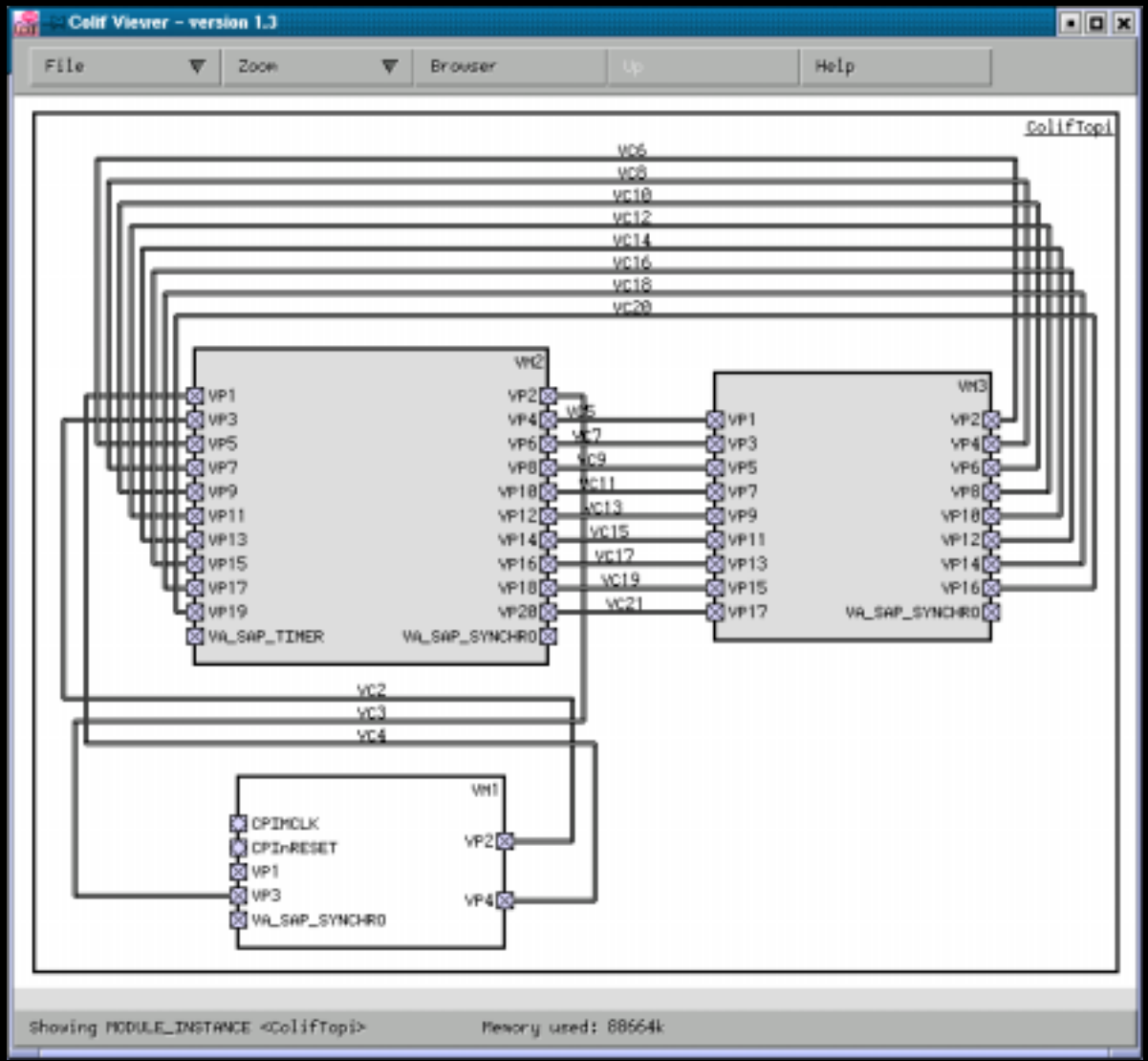
# System Specification & co-simulation

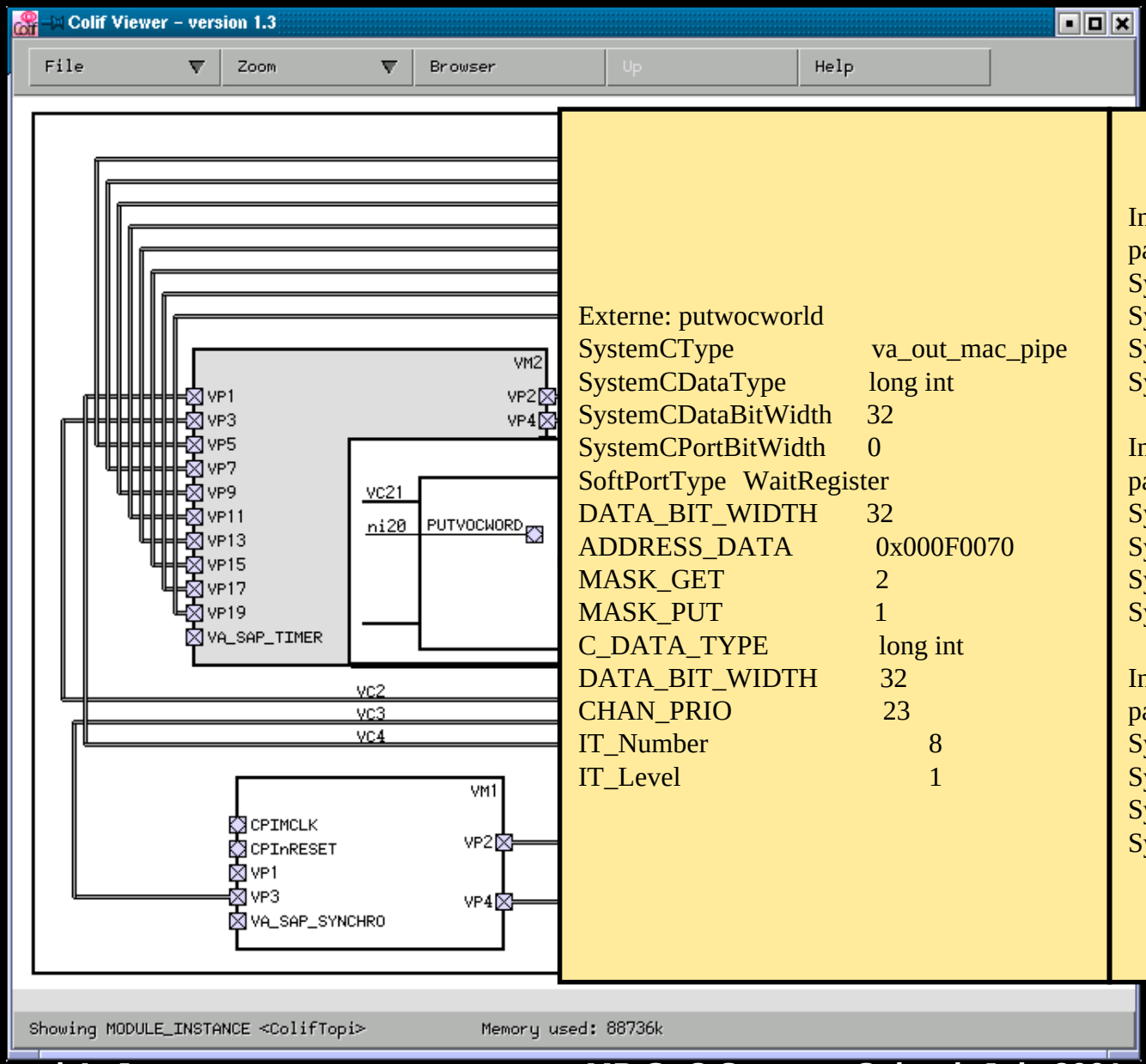
- SystemC specification using virtual component concept
- Abstract communication
- Parameter annotations within specification
- Early co-simulation



# HW-SW Wrapper Generation Steps




Colif Viewer - version 1.3
File
Zoom
Browser
Up
Help
ColifTop1
VC6
VC8
VC10
VC12
VC14
VC16
VC18
VC20
VM2
VP1
VP3
VP5
VP7
VP9
VP11
VP13
VP15
VP17
VP19
VA_SAP_TIMER
VP2
VP4
VP6
VP8
VP10
VP12
VP14
VP16
VP18
VP20
VA_SAP_SYNCHRO
VC5
VC7
VC9
VC11
VC13
VC15
VC17
VC19
VC21
VM3
VP1
VP3
VP5
VP7
VP9
VP11
VP13
VP15
VP17
VP2
VP4
VP6
VP8
VP10
VP12
VP14
VP16
VA_SAP_SYNCHRO
VC2
VC3
VC4
VM1
CPIMCLK
CPInRESET
VP1
VP3
VA_SAP_SYNCHRO
VP2
VP4
Showing MODULE_INSTANCE <ColifTop1>
Memory used: 88664k




Externe: putwocworld

| | |
|---|---|
| SystemCType | va_out_mac_pipe |
| SystemCDataType | long int |
| SystemCDataBitWidth | 32 |
| SystemCPortBitWidth | 0 |
| SoftPortType | WaitRegister |
| DATA_BIT_WIDTH | 32 |
| ADDRESS_DATA | 0x000F0070 |
| MASK_GET | 2 |
| MASK_PUT | 1 |
| C_DATA_TYPE | long int |
| DATA_BIT_WIDTH | 32 |
| CHAN_PRIO | 23 |
| IT_Number | 8 |
| IT_Level | 1 |

Interne: voc_available

| parametre | value |
|---|---|
| SystemCType | sc_in |
| SystemCDataType | bool |
| SystemCDataBitWidth | 1 |
| SystemCPortBitWidth | 0 |

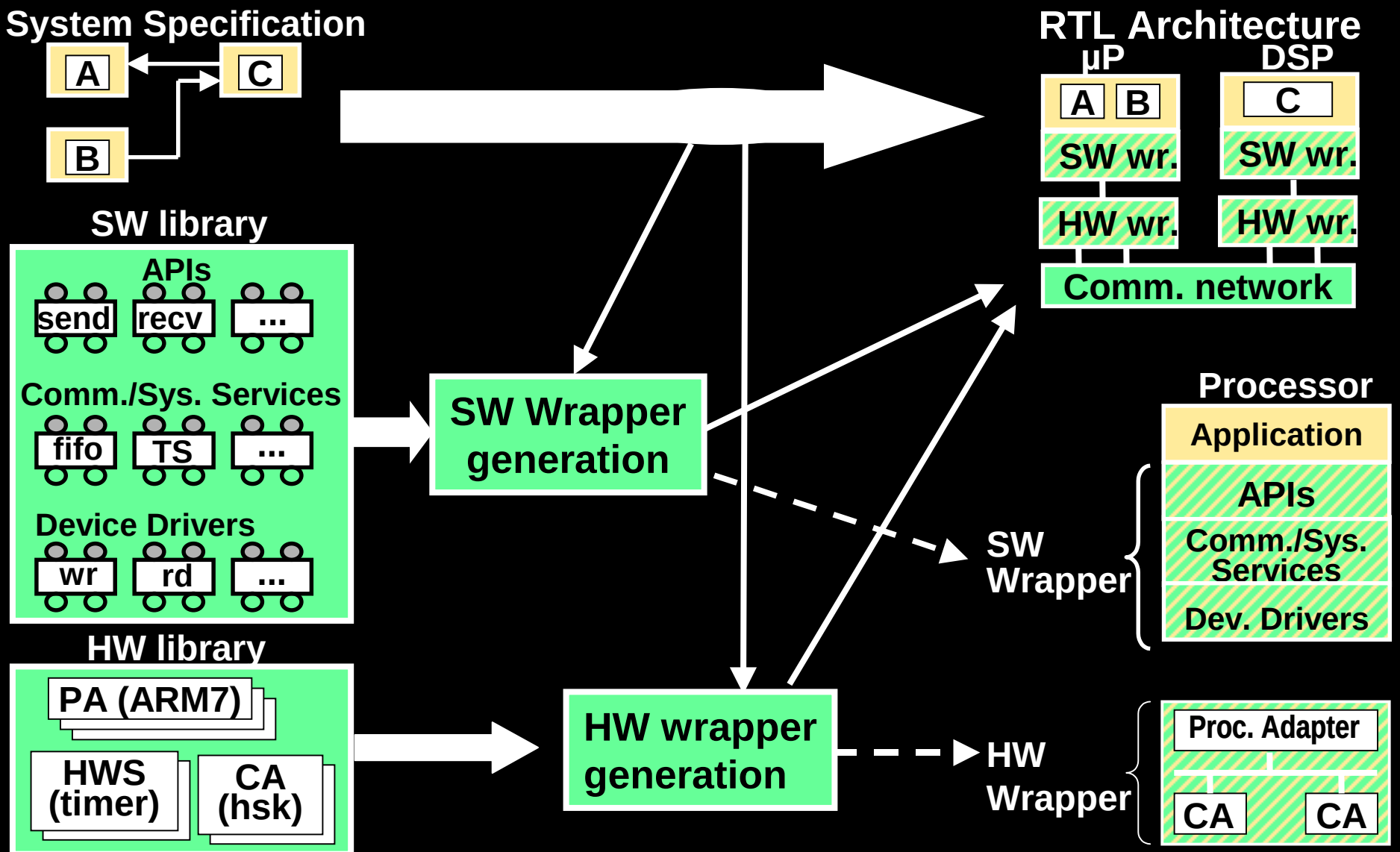
Interne: en_voc_byte

| parametre | value |
|---|---|
| SystemCType | sc_out |
| SystemCDataType | bool |
| SystemCDataBitWidth | 1 |
| SystemCPortBitWidth | 0 |

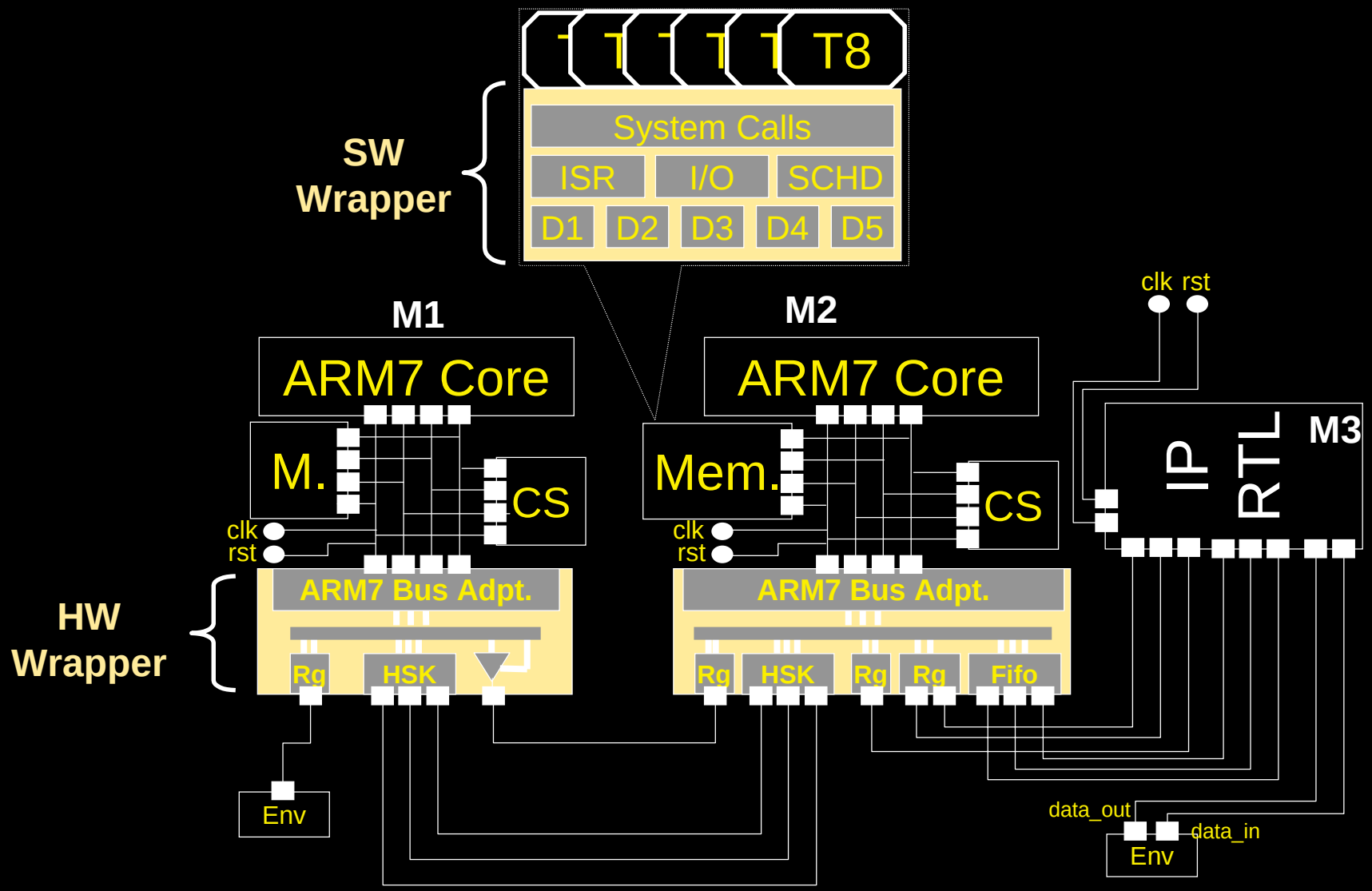
Interne: voc_byte

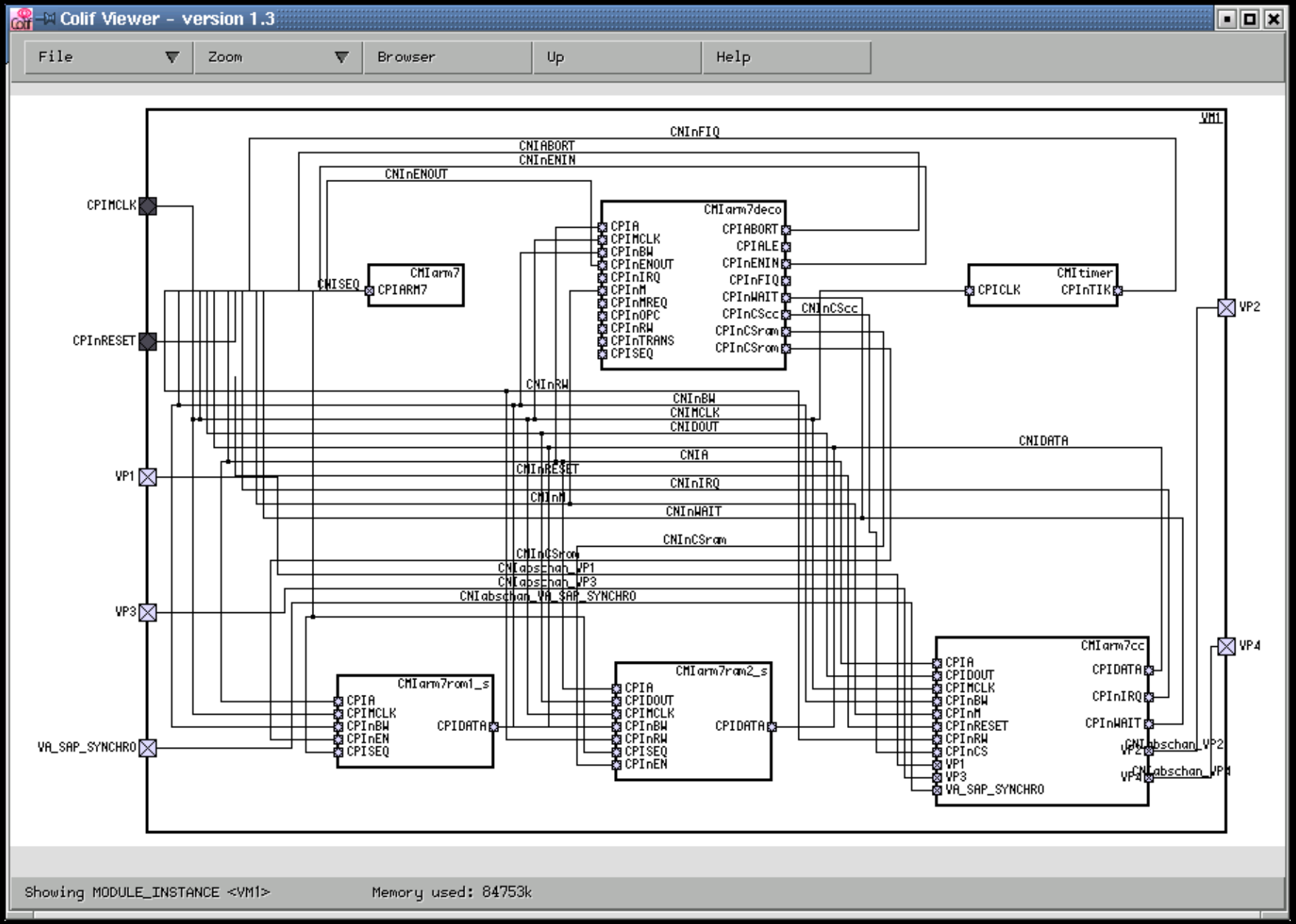
| parametre | value |
|---|---|
| SystemCType | sc_in |
| SystemCDataType | bit8 |
| SystemCDataBitWidth | 8 |
| SystemCPortBitWidth | 0 |

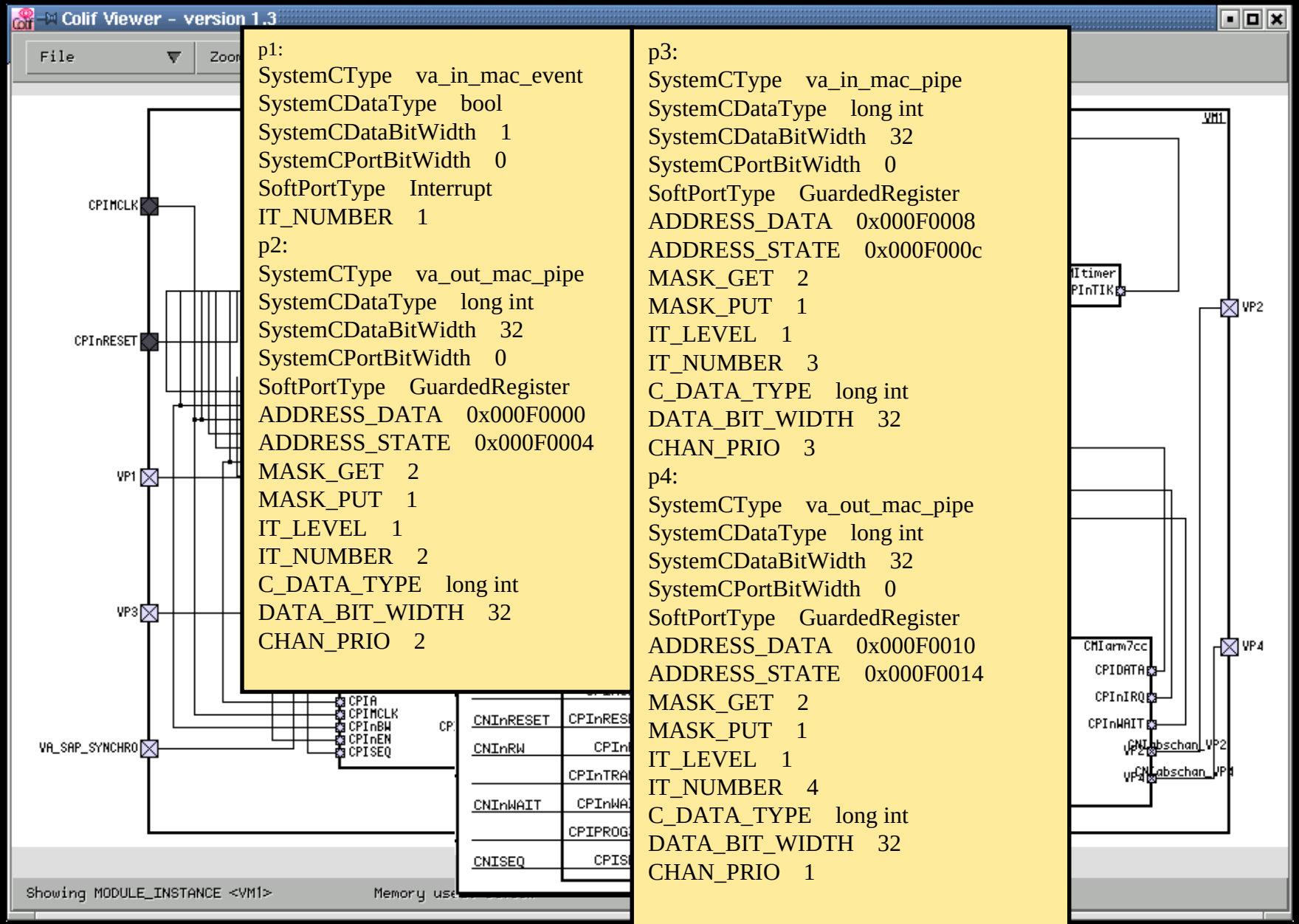
# Architecture generation through systematic HW-SW assembly



# HW-SW Wrapper Generation




Colif Viewer - version 1.3
File
Zoom
Browser
Up
Help
VM1
CNInFIQ
CNIABORT
CNInEMIN
CNInENOUT
CPIMCLK
CPInRESET
VP1
VP3
VA_SAP_SYNCHRO
CNIarm7deco
CPIA
CPIMCLK
CPInBW
CPInENOUT
CPInIRQ
CPInM
CPInMREQ
CPInOPC
CPInRW
CPInTRANS
CPISEQ
CPIABORT
CPIALE
CPInENIN
CPInFIQ
CPInWAIT
CPInCScc
CPInCSram
CPInCSrom
CNISEQ
CNIarm7
CPIARM7
CNItimer
CPICLK
CPInTIK
CNInCScc
VP2
VP4
CNInRW
CNInBW
CNIMCLK
CNIDOUT
CNIDATA
CNIA
CNInRESET
CNInIRQ
CNInM
CNInWAIT
CNInCSram
CNInCSrom
CNIabschan_VP1
CNIabschan_VP3
CNIabschan_VA_SAP_SYNCHRO
CNIarm7ram1_s
CPIA
CPIMCLK
CPInBW
CPInEN
CPISEQ
CPIDATA
CNIarm7ram2_s
CPIA
CPIDOUT
CPIMCLK
CPInBW
CPInRW
CPISEQ
CPInEN
CPIDATA
CNIarm7cc
CPIA
CPIDOUT
CPIMCLK
CPInBW
CPInM
CPInRESET
CPInRW
CPInCS
VP1
VP3
VA_SAP_SYNCHRO
CPIDATA
CPInIRQ
CPInWAIT
CNIabschan_VP2
CNIabschan_VP4
Showing MODULE_INSTANCE <VM1>
Memory used: 84753k




p1:
SystemCType va_in_mac_event
SystemCDataType bool
SystemCDataBitWidth 1
SystemCPortBitWidth 0
SoftPortType Interrupt
IT_NUMBER 1

p2:
SystemCType va_out_mac_pipe
SystemCDataType long int
SystemCDataBitWidth 32
SystemCPortBitWidth 0
SoftPortType GuardedRegister
ADDRESS_DATA 0x000F0000
ADDRESS_STATE 0x000F0004
MASK_GET 2
MASK_PUT 1
IT_LEVEL 1
IT_NUMBER 2
C_DATA_TYPE long int
DATA_BIT_WIDTH 32
CHAN_PRIO 2

p3:
SystemCType va_in_mac_pipe
SystemCDataType long int
SystemCDataBitWidth 32
SystemCPortBitWidth 0
SoftPortType GuardedRegister
ADDRESS_DATA 0x000F0008
ADDRESS_STATE 0x000F000c
MASK_GET 2
MASK_PUT 1
IT_LEVEL 1
IT_NUMBER 3
C_DATA_TYPE long int
DATA_BIT_WIDTH 32
CHAN_PRIO 3

p4:
SystemCType va_out_mac_pipe
SystemCDataType long int
SystemCDataBitWidth 32
SystemCPortBitWidth 0
SoftPortType GuardedRegister
ADDRESS_DATA 0x000F0010
ADDRESS_STATE 0x000F0014
MASK_GET 2
MASK_PUT 1
IT_LEVEL 1
IT_NUMBER 4
C_DATA_TYPE long int
DATA_BIT_WIDTH 32
CHAN_PRIO 1

# VDSL design summary

- System specification/Validation
  - VADeL model: Lines/Nets 150/21
  - Simulation model Lines/Nets 393/60
  - Simulation model generation time: 90s
- Architecture design
  - HW Wrapper generation
    - VM1: 3284 Gates, area 818 µm², maxF ck 168 MHz
    - VM2: 3795 Gates, area , maxF ck 162 MHz
    - Latence : Write 2 clk-cycles, Read 6 clk-cycles
    - VHDL code 2168 lines of VHDL RTL
  - SW Wrapper generation
    - VM1 : 1249 lines (281 ASM), Mem Code/data 1484/450 bytes
    - VM1 : 2153 lines (281 ASM), Mem Code/data 2624/1020 Bytes

# Conclusions

- Multiprocessor SoC: already a reality and main future driver
- System Specifications: Few Concepts, Too many languages
- Key issues for MP SoC specification/validation:
  - On chip communication network abstraction,
  - Execution and refinement  for multi and mixed level model
- COLIF: A Design Model for MP SoCs Specification
  - Abstract wrappers to connect heterogeneous components
  - Architecture generation through systematic HW-SW assembly

# Reading about system specification


1. A. Lee and A. Sangiovanni-vicentelli, A Denotational Framework for Comparing Models of Computation, ERL Memorandum UCB/ERL-M97/11, University of California, Berkley, CA 94720, January 1997.
2. A. Jantsch, S. Kumar, A. Hemani, « The Rugby Model: A Metamodel for Studying Concepts in Electronic System Design », *IEEE Design & Test of Computers*, 2000, p. 78-85.
3. D. D. Gajski, J. Zhu, R. Zömer, A. Gerstlauer, S. Zhao, *SpecC Specification Language and Methodology,* Kluwer Academic Publishers, Boston, MA, ISBN 0-7923-7822-9, March 2000.
4. M. Sgroi, L. Lavagno, A.S. Vicentelli, « Formal Models for Embeded System Design », IEEE Design & Test of Computers, vol. 17, no. 12, April-June 2000.
5. SystemC, available at http://www.systemc.org
6. R. Ernst, D. Ziegenbein, K. Richter, L. Teich, "Hardware/Software Co-Design of Embedded Systems - The SPI Workbench," Proc. IEEE Workshop on VLSI'99, pp. 9-17, Orlando, 1999.

# Reading about system co-simulation


1. J.A. Rowson « Hardware/Software Co-simulation », proceeding Design Automation Conference, 1994.
2. C.A. Valderrama, A. Changuel, P.V. Vijaya-Raghavan, M. Abid, T. Ben Ismail, A.A. Jerraya, "A unified model for co-simulation and co-synthesis of mixed hardware/software systems", European Design and Test Conference (EDAC-ETC-EUROASIC'95), Paris, France, March 1995.
3. L. Séméria and A. Ghosh, “Methodology for Hardware/Software Co-verification in C/ C++” , Proceeding of ASPDAC, 2001.
4. Seamless CVE, available at http://www.mentorg.com
5. C. Passerone, L. Lavagno, M. Chiodo, A. Sangiovanni-Vincentelli “Fast hardware/software co-simulation for virtual prototyping and trade-off analysis”, in Proceedings of Design Automation Conference, June, 1997
6. Coware, Inc. “N2C” , available at http : // coware.com/cowareN2C.html

# Reading about COLIF and the Architecture design environment


1. S. Yoo, al. « A Generic Wrapper Architecture for Multi-Processor SoC Cosimulation and Design », *CODES*, 2001.
3. W.O. Cesario, al. "Colif: a Multilevel Design Representation for Application-Specific Multiprocessor System-on-Chip Design", IEEE Design & Test, Sept, 2001.
4. A. Baghdadi, al. « An Efficient Architecture Model for Systematic Design of Application-Specific Multi-processor SoC », *Design Automation and Test in Europe*, March, 2001.
5. L. Gauthier, al. « Automatic Generation and Targeting of Application specific Operating Systems and Embedded systems software », *Design Automation and Test in Europe*, March, 2001.
6. D. Lyonnard, al. "Automatic Generation of Application-Specific Architecture for Heterogeneous Multiprocessor System-on-Chip", DAC, June, 2001.
7. S. Meftali, al. « An Optimal Memory Allocation for Application-Specific Multiprocessor System-on-Chip », soumis pour publication ISSS, octobre 2001.
8. P. Gerin, al. « Scalable and Flexible Co-simulation of SoC Design with heterogeneous Multi-processor Target Architecture », *Asia South Pacific Design Automation conference*, January, 2001.
9. G. Nicolescu, S. Yoo, A.A. Jerraya, « Mixed-Level Cosimulation for Fine Gradual Refinement of Communication in SoC Design », *Design Automation and Test in Europe*, mars, 2001.