
Component-Based Design Environment for Multiprocessor SoC Platforms

Dr. Wander Cesário
TIMA Laboratory
46, avenue Félix Viallet
38031 Grenoble - France
<http://tima.imag.fr/>



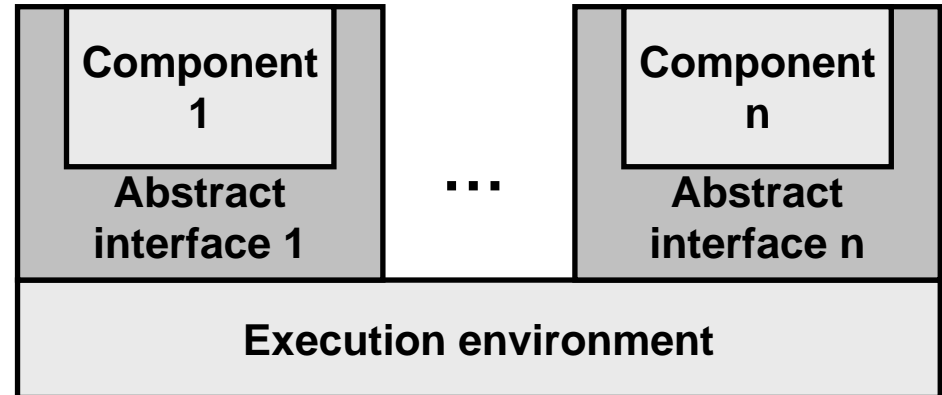
- Introduction
- Virtual Component Model
- ROSES: Component-based design environment
- Case study: MPEG-4 encoder
- Summary

- Reuse is the enabling technology for SoC design
- Global SoC models may include software, hardware and functional components
- Component integration is a source of design bottlenecks (e.g., bridges, I/O drivers, ad-hoc co-simulation setups, etc.)
 - may be automated, the key technologies are:
 - ▲ virtual component model
 - ▲ component-based design environment

- Introduction
- Virtual Component Model
- ROSES: Component-based design environment
- Case study: MPEG-4 encoder
- Summary

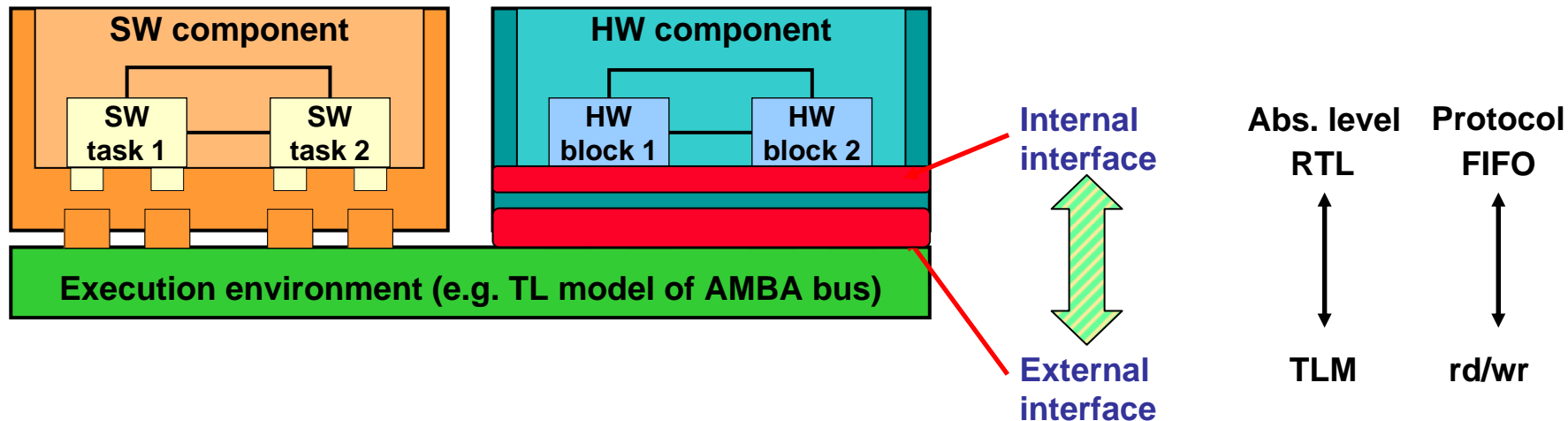
The Virtual Component Model (1/2)

- Virtual component
 - Component
 - ▲ Hardware
 - ▲ Software
 - ▲ Functional
 - Abstract interfaces
 - ▲ Required **services**
 - ▲ Provided **services**
 - ▲ Control **services**
 - ▲ Synchronization **services**
 - ▲ **Parameters**
- Execution environment
 - Abstract platform (e.g. NoC, cosimulation backplane, ...)
 - Heterogeneous component thanks to adaptation



The Virtual Component Model (2/2)

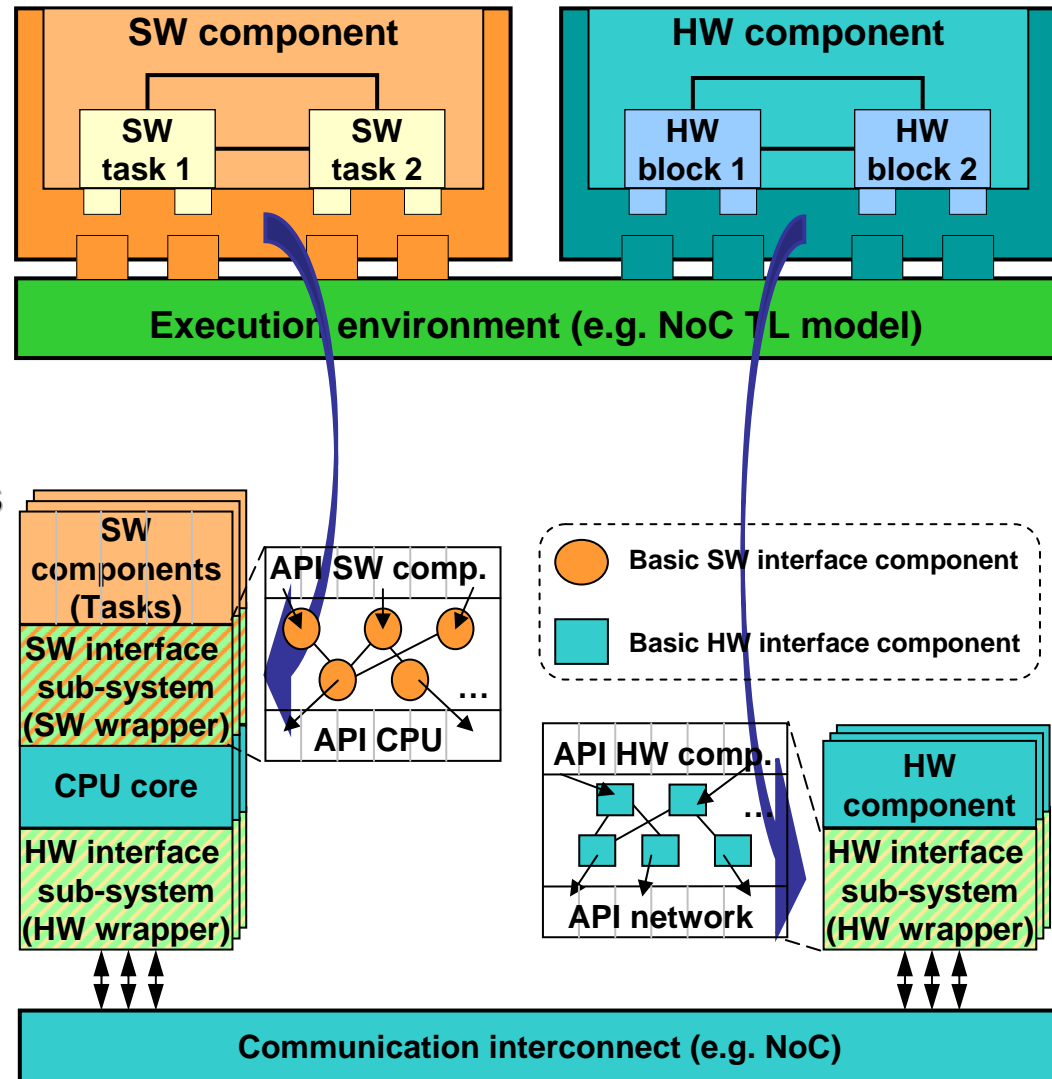
- Basic model: a set of hierarchically interconnected **virtual modules** and an execution environment
- Virtual Module:
 - **Content**: Tasks/Instances + communication channels)
 - **Abstract interface**: set of virtual ports (**internal**, **external**)



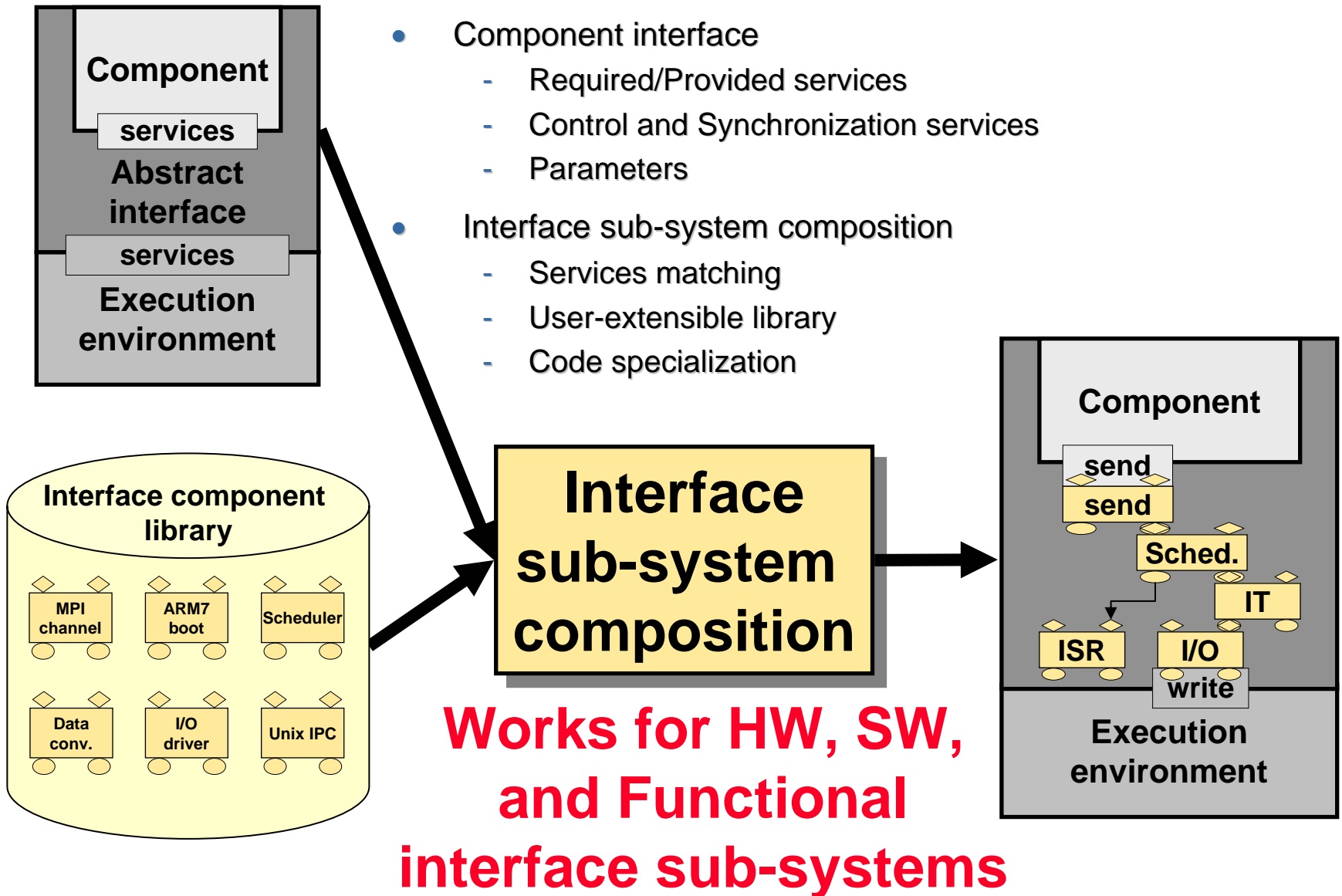
- Introduction
- Virtual Component Model
- ROSES: Component-based design environment
- Case study: MPEG-4 encoder
- Summary

System-level SoC Design Flow

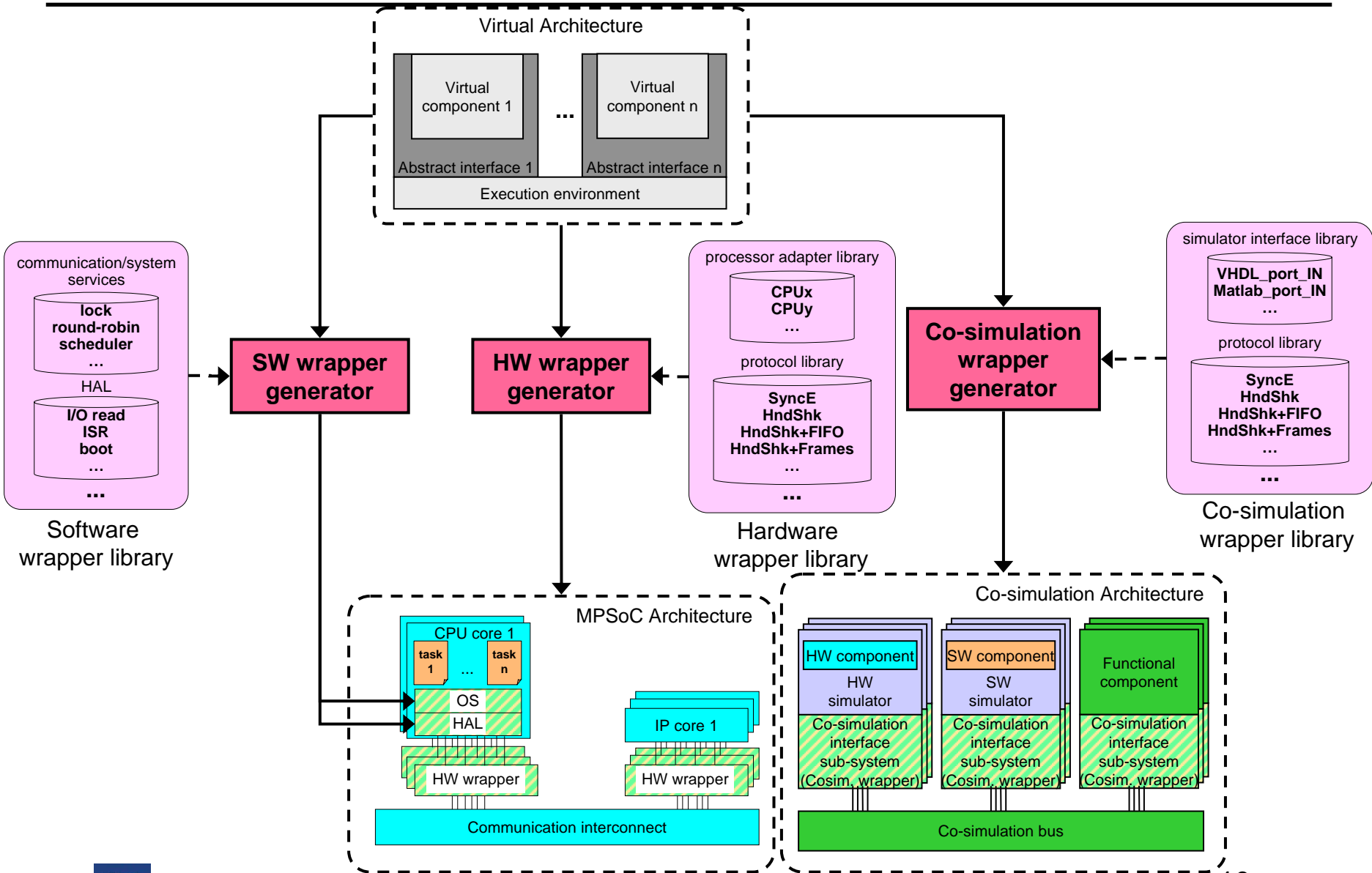
- System specification is a virtual architecture: virtual modules use abstract interfaces for HW/SW communication
- Architecture implementation: heterogeneous components and sophisticated interconnect linked through HW and SW interface sub-systems
- Automatic generation of application-specific HW/SW interface sub-systems from basic interface components



Key Technology: composing Interfaces



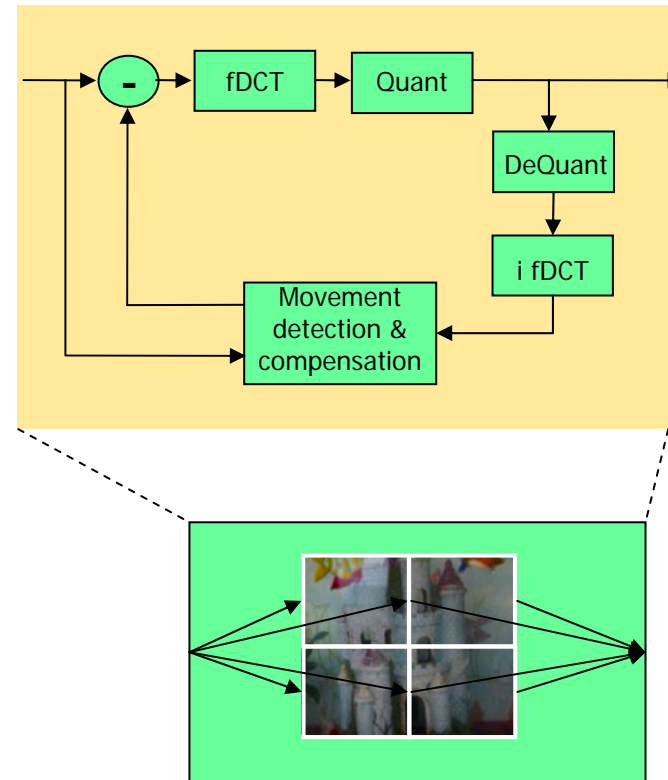
ROSES: Component-based Design Environment



- Introduction
- Virtual Component Model
- ROSES: Component-based design environment
- Case study: MPEG-4 encoder
- Summary

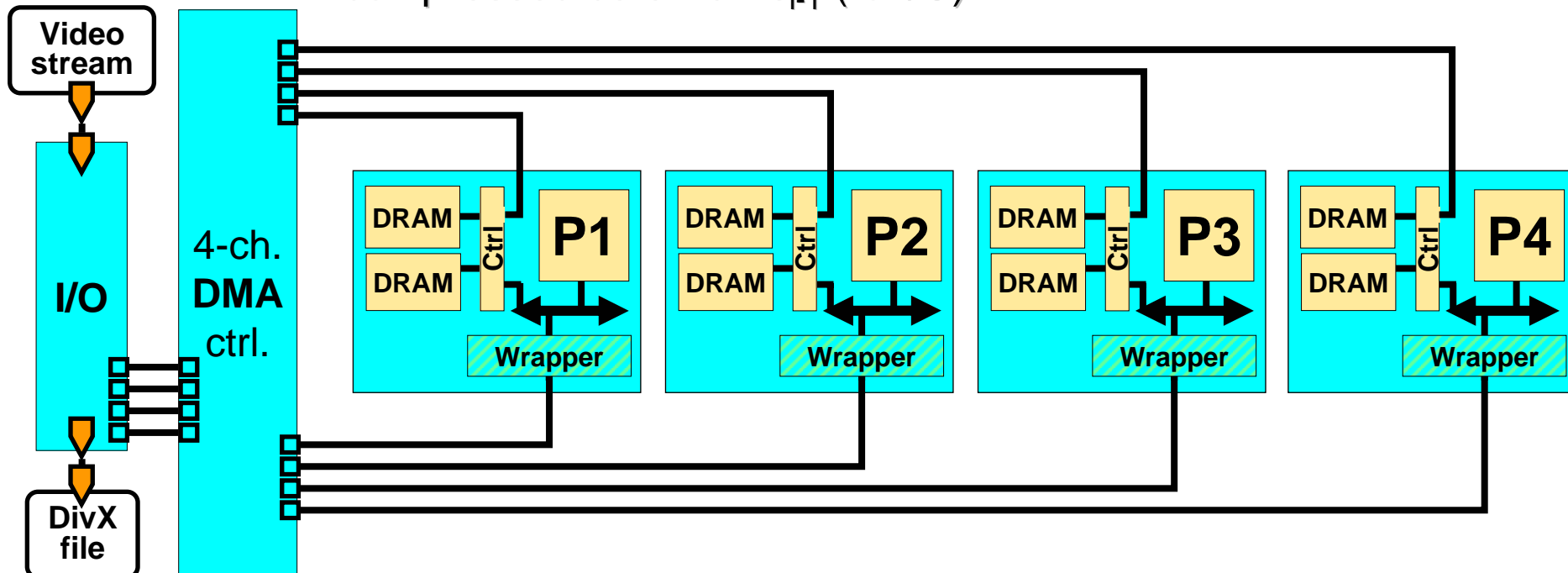
MPSoC design of a DivX encoder

- OpenDivX
 - Open source MPEG-4 encoder/decoder
 - Modified to work concurrently on 1/4th of each frame



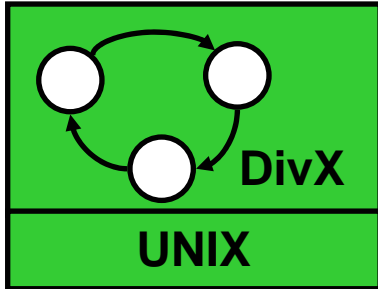
DivX encoder target architecture

- Nexperia-like 4-processors architecture
- Point-to-point communication networks
- Application-specific 4-channel DMA controller
- Use of a double banc DRAM for local memories:
 - CPU: $1/4^{\text{th}}$ frame_{*i*} + compressed data frame_{*i*}
 - DMA: $1/4^{\text{th}}$ frame_{*i+1*} (from I/O)
+ compressed data frame_{*i-1*} (to I/O)

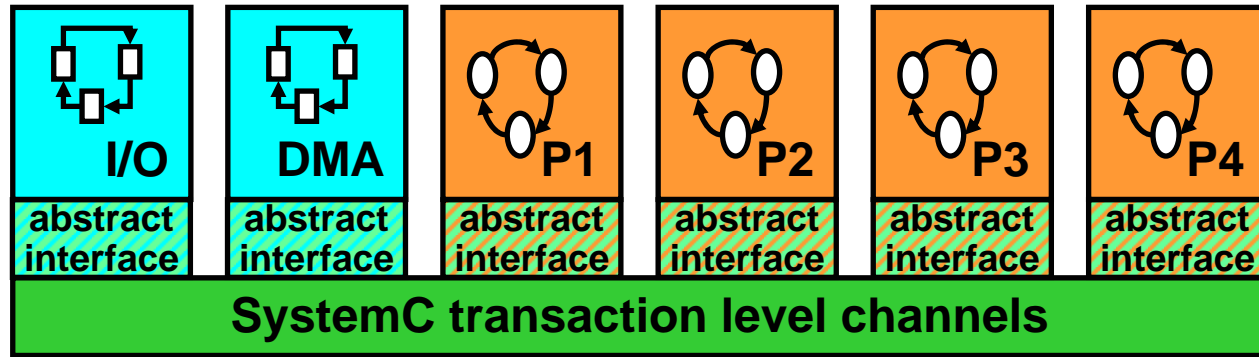


DivX main design steps

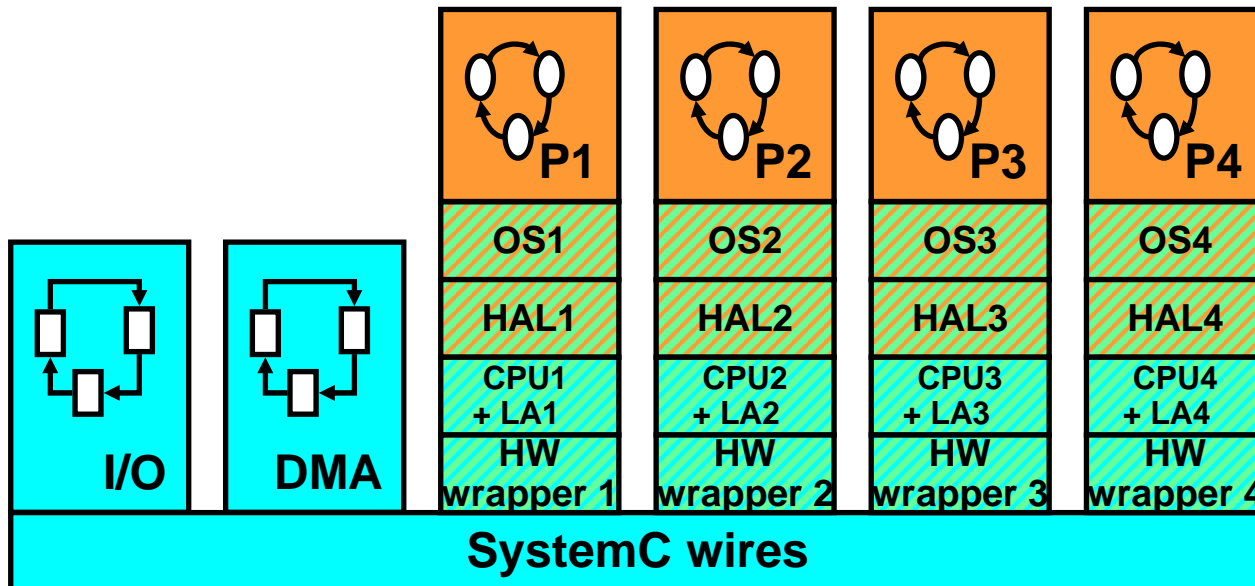
1. Algorithm



2. Transaction/RT-level mixed model in SystemC



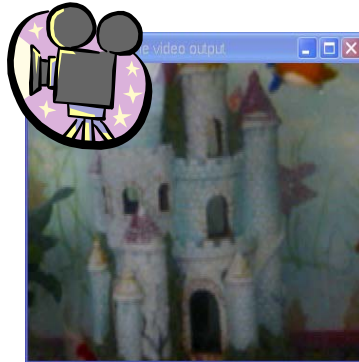
3. After refinement of HW/SW interfaces



4. Implementation

- RTL design flow
- ARM integrator

DivX MPEG-4 encoder demonstrator



```
sim <4>
sim <3>
sim <2>
sim
Program received signal SIGUSR2, User defined signal 2.
VProc1: working SHM banc: 1

Program received signal SIGUSR2, User defined signal 2.
VProc1: data to compress: 0x4034f000
VProc1: memory in frame: 6

68650019 60697170 6d72605e 5d5c5b60 6365605d
67605f60 6c63666b 777b7871 84848078 84838282
8f95988d 75778088 82818480 97949289 a9a5a29f
a7a5abab 9f9fa1a6 8ba6a49e 4446556d 3d3a3b40
VProc1: compressed data: 0x40351534
VProc1: memory out size of bitstream: 3be

4a0f0000 d50b5549 5a5a61e9 6a945698 429dab17
7ae1cd72 31213cb5 984427e0 421cab17 adb095b4
552f6020 a8ea05ae e852bd70 13c09411 c90851af
a6c30946 842f6084 f101a1e1 668043b 9085ea05

Program received signal SIGUSR2, User defined signal 2.
VProc1: Task1 -Frame Calculated

Program received signal SIGUSR2, User defined signal 2.
```

```
[divxman] @ colorado.imag.fr/home/divxman/DivX/MemStrv2/Cosim/OSUnix
IO: setting IO_DONE
--- CMMemCtrl_VProc1 MEMCTRL banc changed: 1
--- CMMemCtrl_VProc2 MEMCTRL banc changed: 1
--- CMMemCtrl_VProc3 MEMCTRL banc changed: 1
--- CMMemCtrl_VProc4 MEMCTRL banc changed: 1
Memserv::automata = Going out from INPUTPROC1STATE
Memserv::automata = Going out from OUTPUTPROC1STATE
Memserv::automata = Going out from INPUTPROC2STATE
Memserv::automata = Going out from OUTPUTPROC2STATE
Memserv::automata = Going out from INPUTPROC3STATE
Memserv::automata = Going out from OUTPUTPROC3STATE
Memserv::automata = Going out from INPUTPROC4STATE
Memserv::automata = Going out from OUTPUTPROC4STATE
IO: frameCount = 0
New frame read
IO: setting IO_DONE
--- CMMemCtrl_VProc1 MEMCTRL banc changed: 0
--- CMMemCtrl_VProc2 MEMCTRL banc changed: 0
--- CMMemCtrl_VProc3 MEMCTRL banc changed: 0
--- CMMemCtrl_VProc4 MEMCTRL banc changed: 0
Memserv::automata = Going out from INPUTPROC1STATE
Memserv::automata = Going out from OUTPUTPROC1STATE
Memserv::automata = Going out from INPUTPROC2STATE
```

Summary

- Component-based design environment starting from a virtual component model specification
- Eases integration of hardware, software and functional components for global MPSoC design and validation
- Automates the integration of complex interface sub-systems by composing components from an user-extensible open-library