

Mixed SW/SystemC SoC Emulation Framework



Màrius Montón , Antoni Portero, Marc Moreno, Borja Martínez, Jordi Carabina
 {marius.monton, antoni.portero, marc.morenob, borja.martinez, jordi.carrabina}@uab.cat
 CEPHIS. Dpt. Microelectronics and Electronics Systems. **Universitat Autònoma de Barcelona**



Motivation

The Design of HW/SW Cores (IPs) for specific applications that have to be connected to Standard Platforms (PCs, PDAs, etc.) is a complex multi-disciplinary process that requires plenty of time and mixed design teams

These IPs require OS Drivers to connect to the Platform

Functional Code Testing is required at early design stages

But:

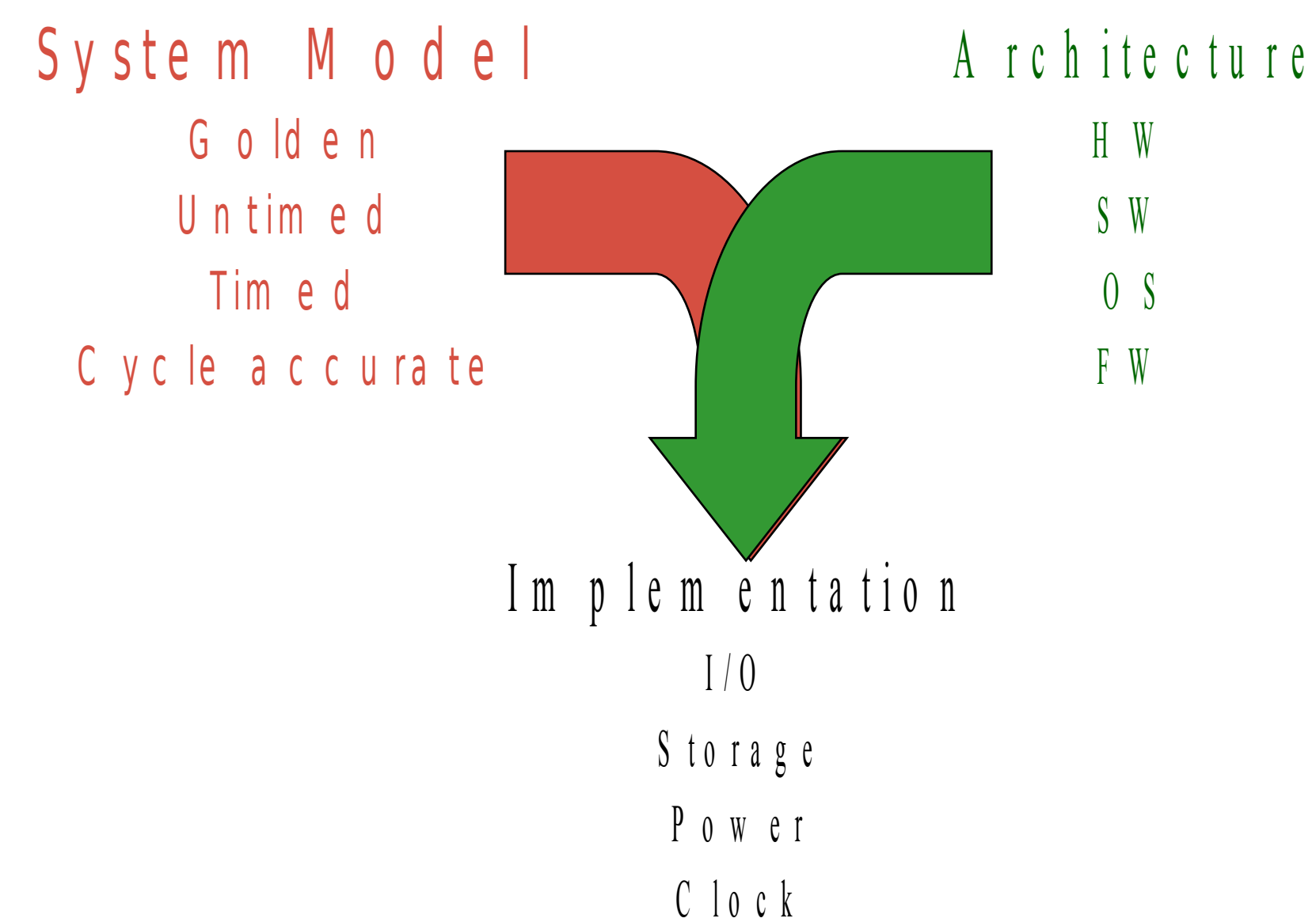
- Cycle-accurate System Level Model Verification with a OS running on top is a nightmare
- OS Driver development is tough
- HW Redesign affects the whole system and requires often driver redesign

Bad News

Proposal

Use HW/SW System-Level Languages (i. e. SystemC) for modeling HW using C++ classes

Plug the SystemC “domain” (HW model + simulator) into the SW platform emulator (i.e. QEMU)

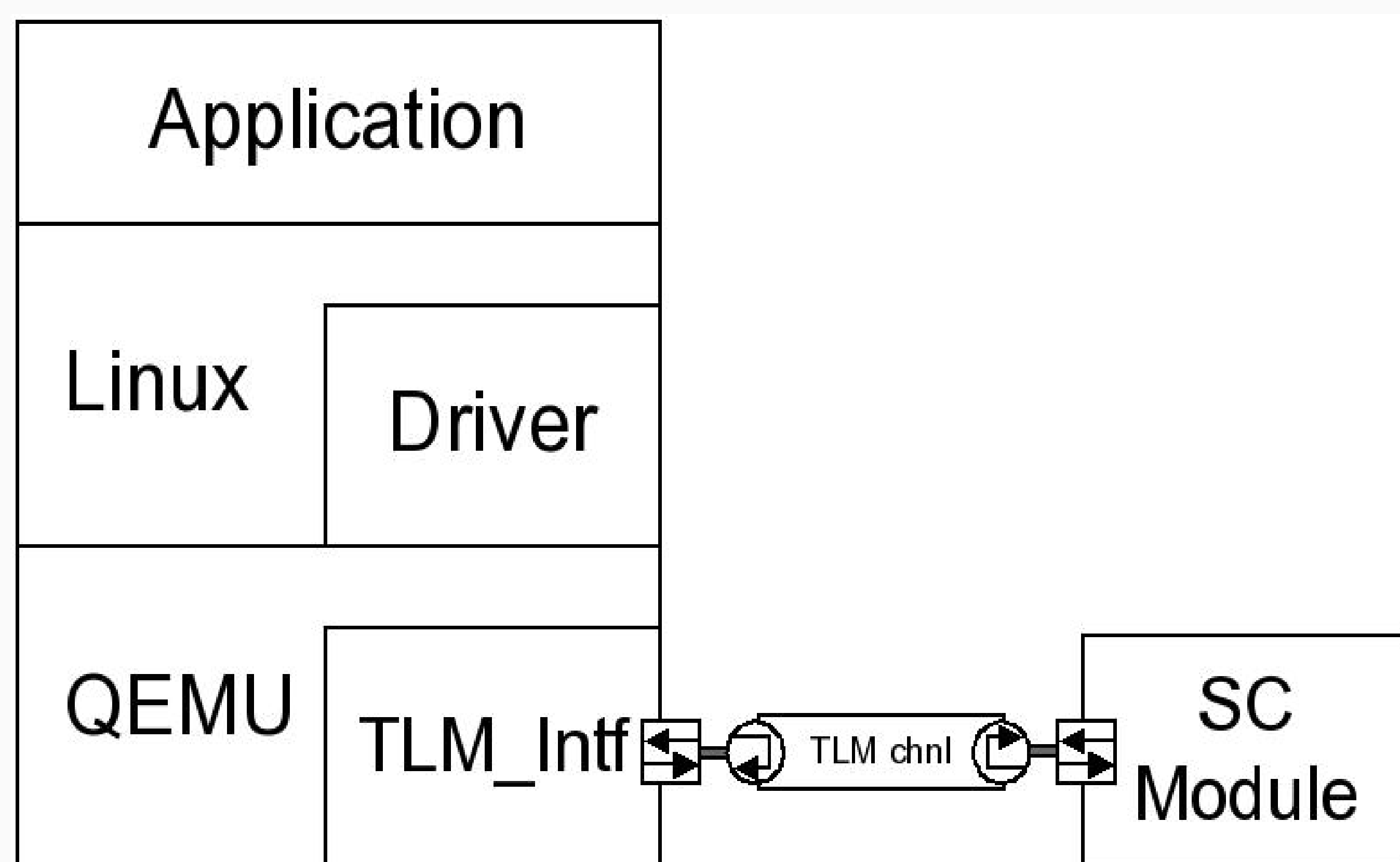


Goals

- Set-up a new SystemC-oriented emulation framework for HW/SW co-verification
- Virtual machines (architectures): x86, x86-64, SPARC, ARM, MIPS
- Allows Standard OS & SW: Linux, RTLinux, Windows

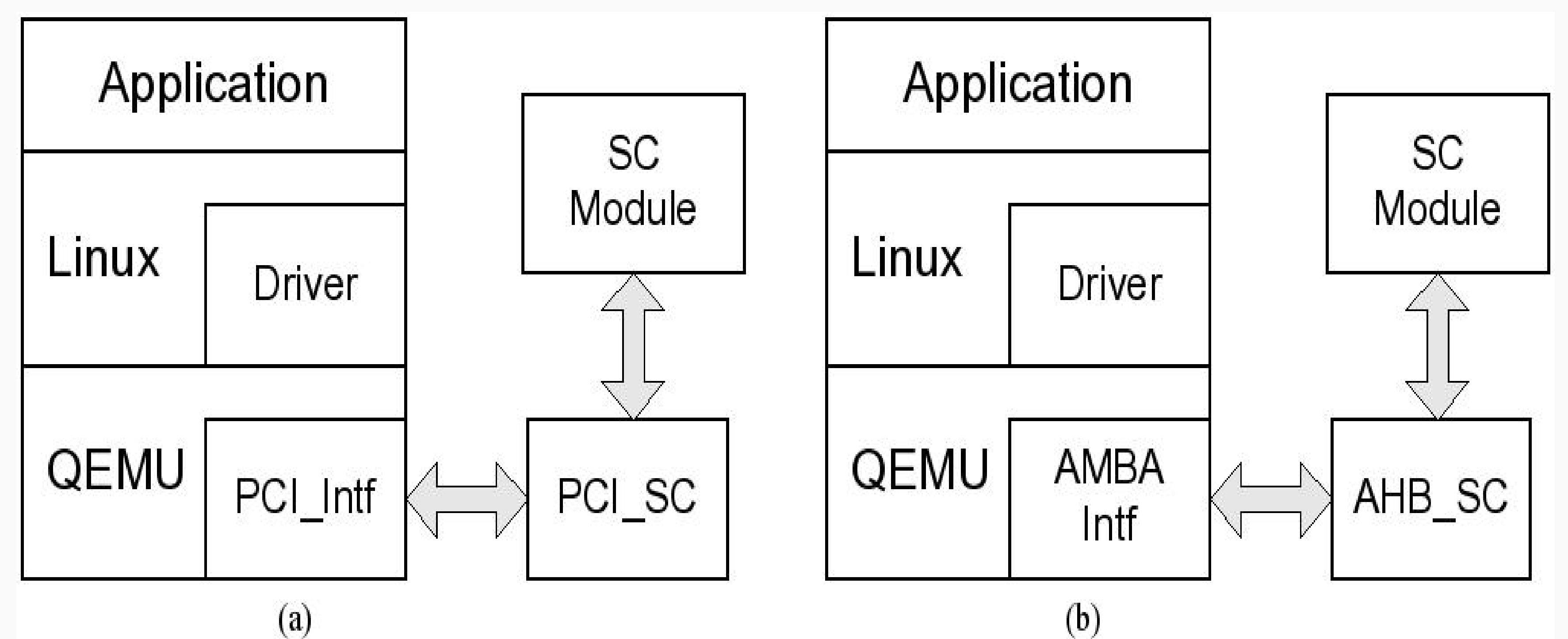
TLM Level

For generic non-platform dependent implementation



RTL Level

For platform dependent implementation
 (a) PCI on PC-ix86 (b) AMBA on Versatile



Experiences with Complex Systems

Functional NoC

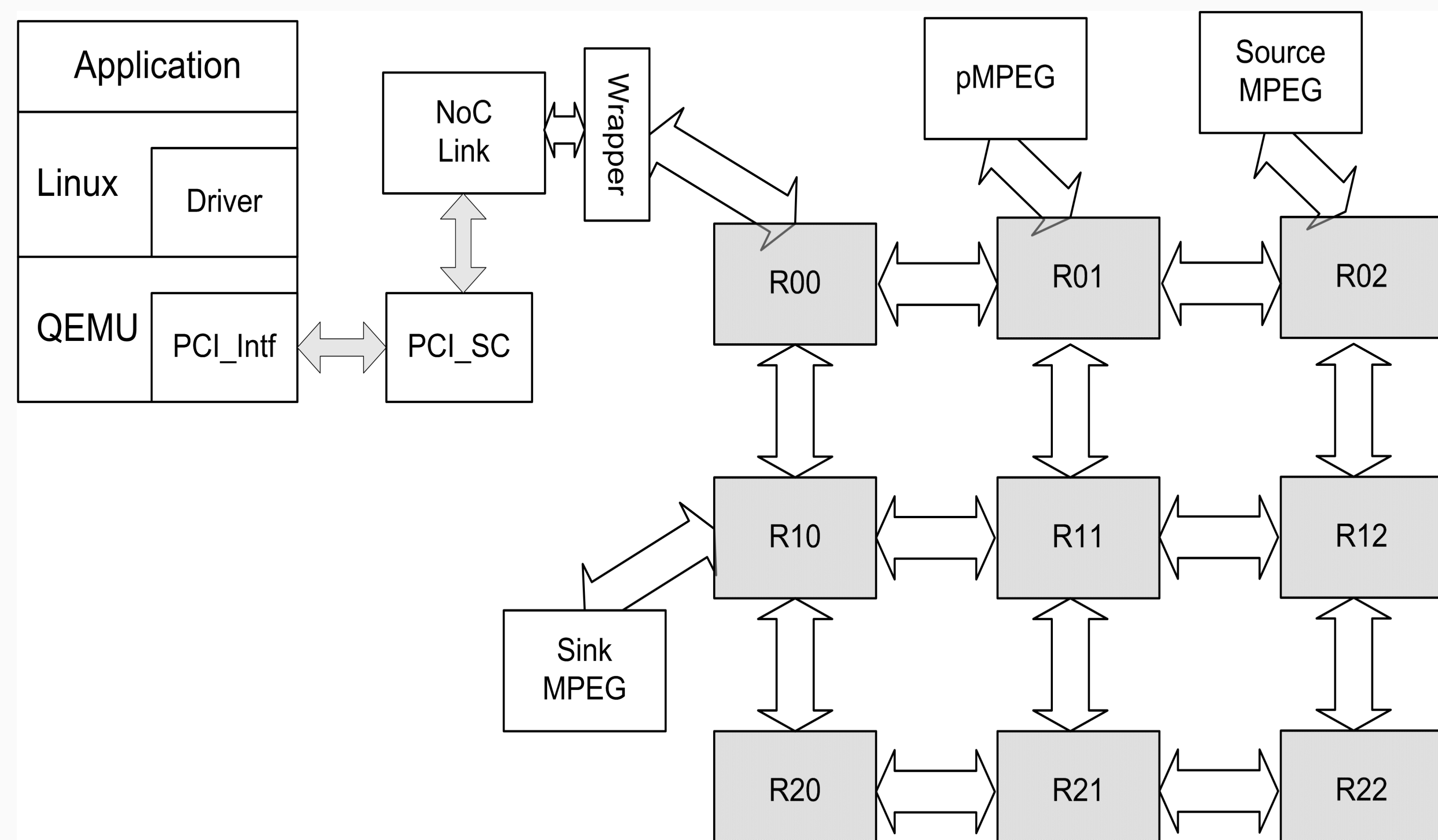
developed in SystemC

MPEG-4 Encoder

Multiple Tasks connected to the NoC processing tiles

System Level Test

- Application: MPEG-4 with QoS
- O.S. Driver: /dev/mpeg /proc/mpeg
- RTL HW: Synthesizable SystemC code (ForteDS Synthesizer)



QEMU overhead

We evaluated QEMU overhead using a simple MPEG-2 decoder example:

Either running on real platform vs. same platform emulated by QEMU

Platform	Execution time	OverHead
Real i386 PC	0,088	7
QEMU i386 PC	0,638	
Real ARM-9	1,06	1,4
QEMU ARM-9	1,5	

- x7 Overhead for intel-x86
- Only 1.4 overhead for ARM Systems
- Fast enough to emulate entire system, O.S. boot and running applications

Conclusions

- QEMU-SC is very useful for SoC simulation and verification when developing custom HW modules
- Can simulate bus transfers for most common SoC buses
- Can generate bus transfer traces to be used for realistic testbenches
- Use of sockets enables distributed computation

Future Work

- Enabling use of time for performance measurements
- Emulate partial reconfiguration management

<http://cephis.uab.es/proj/public/qemu/>