Automatic Generation of HdS System Models for System Simulation using IP-XACT

DATE 2011 Workshop "Hardware Dependent Software Solutions for SoC Design"

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

Microelectronics Engineering Group

University of Cantabria

SOFTSOC

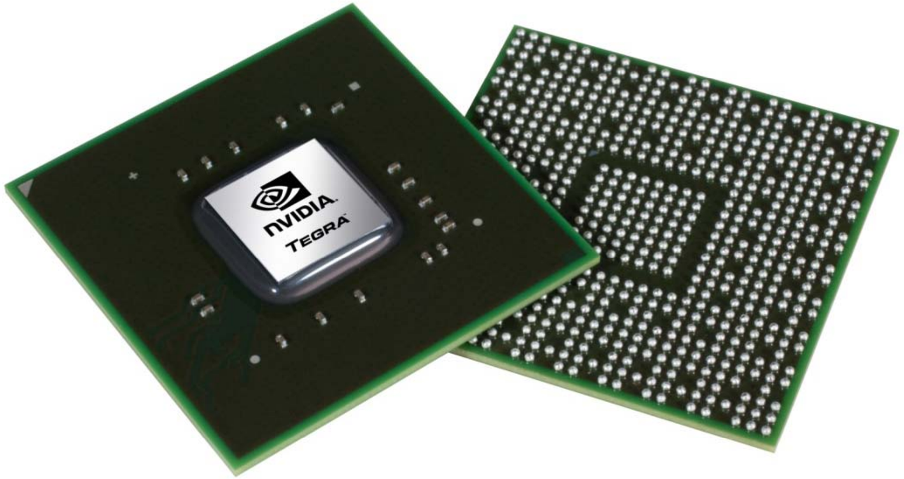
Talk Outline

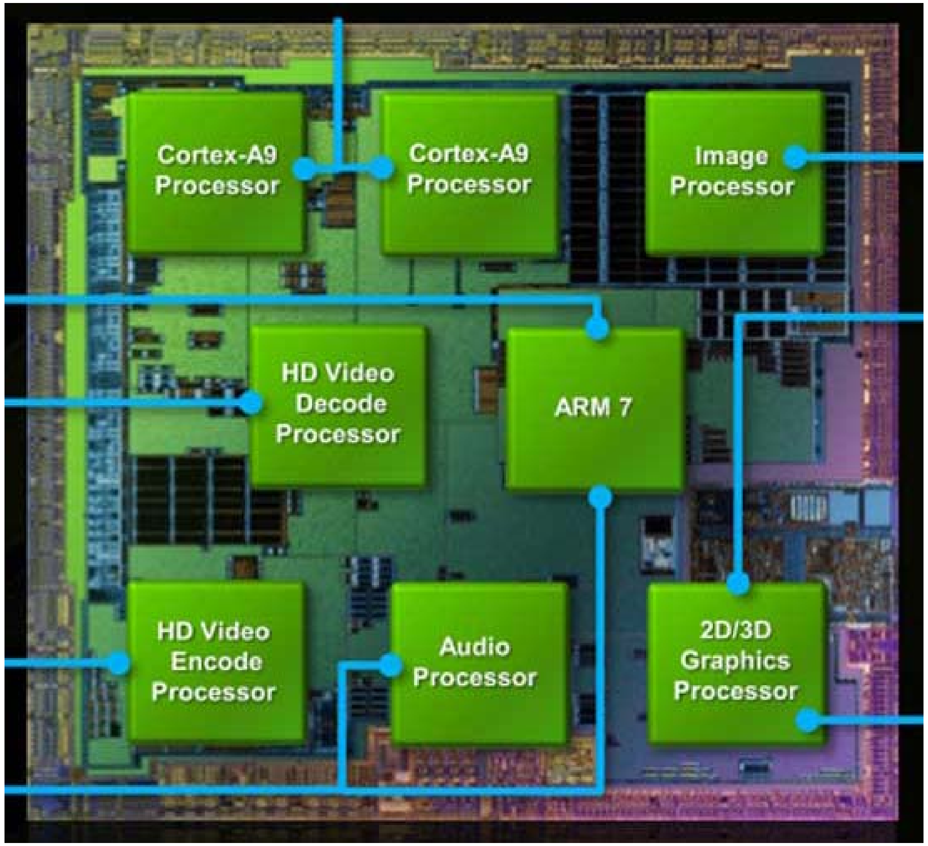
- Introduction
  - Context
  - Motivation
  - Objectives
- Technologies for HdS System Simulation
- IP-XACT System Description Methodology
- Automatic Generation of HdS System Models
  - Application to SCoPE Native Simulator
- Conclusions

Introduction: Context

- **Modern MPSoCs: Entire Systems integrated on a single die**

[Nvidia Tegra 2]

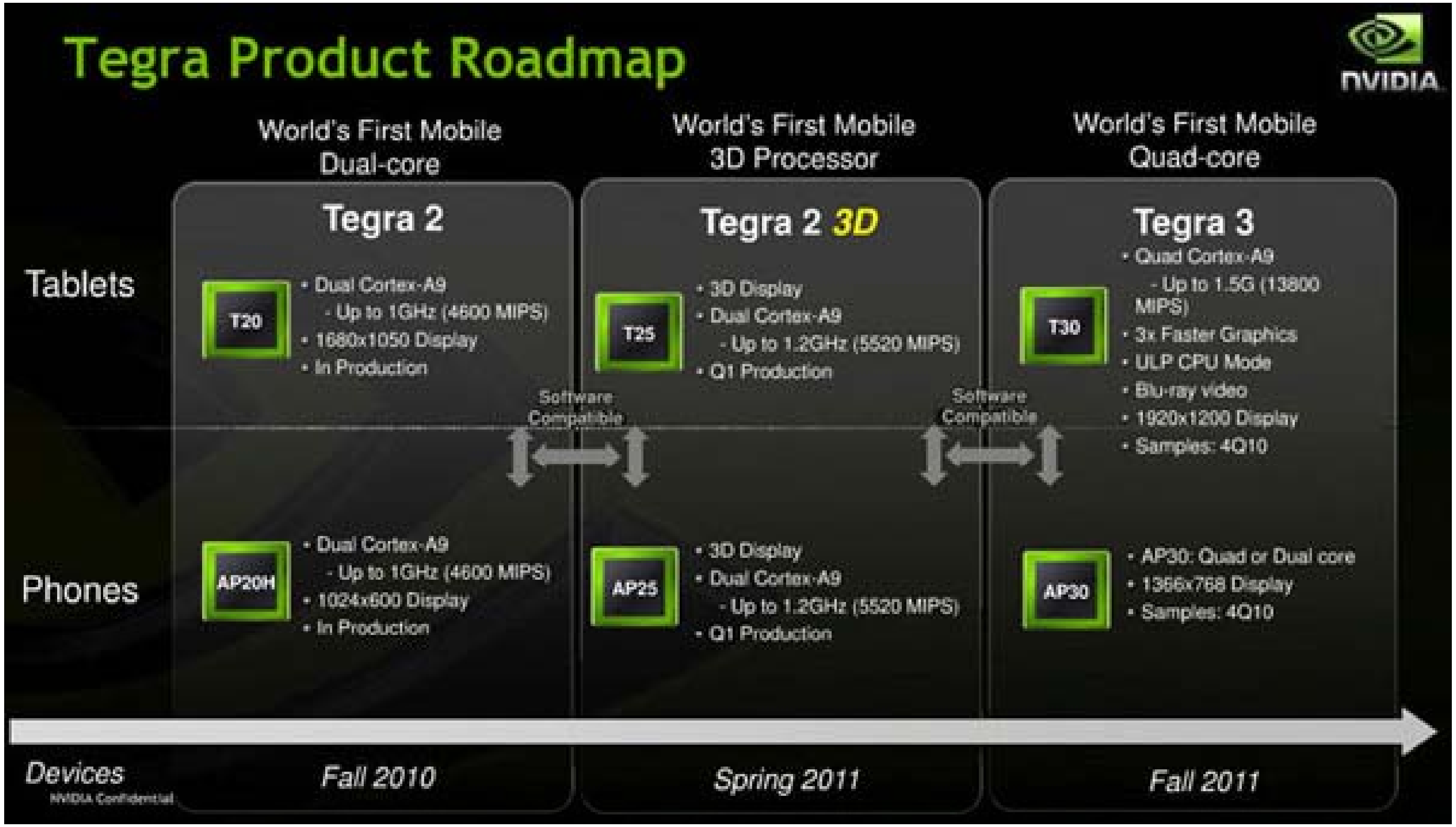




Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

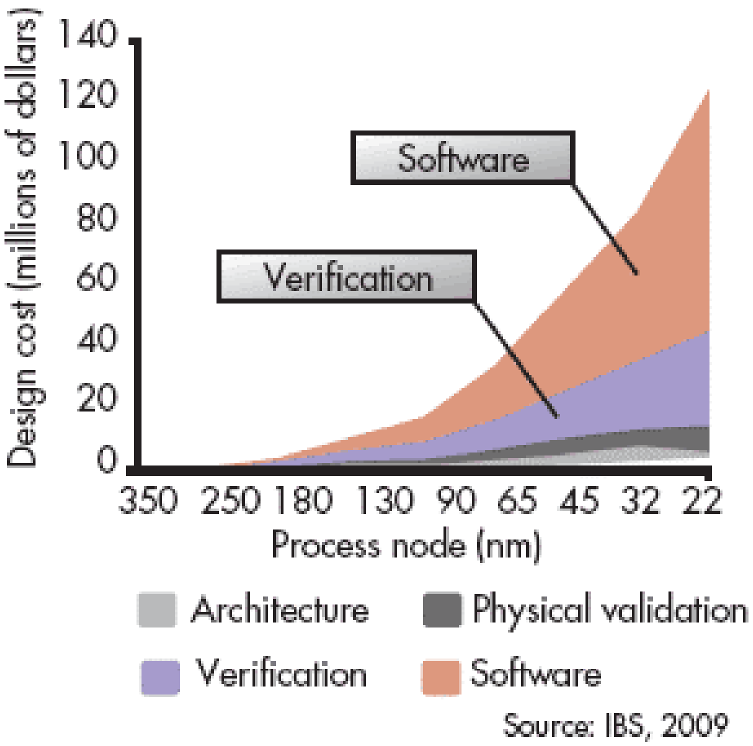
Introduction: Context

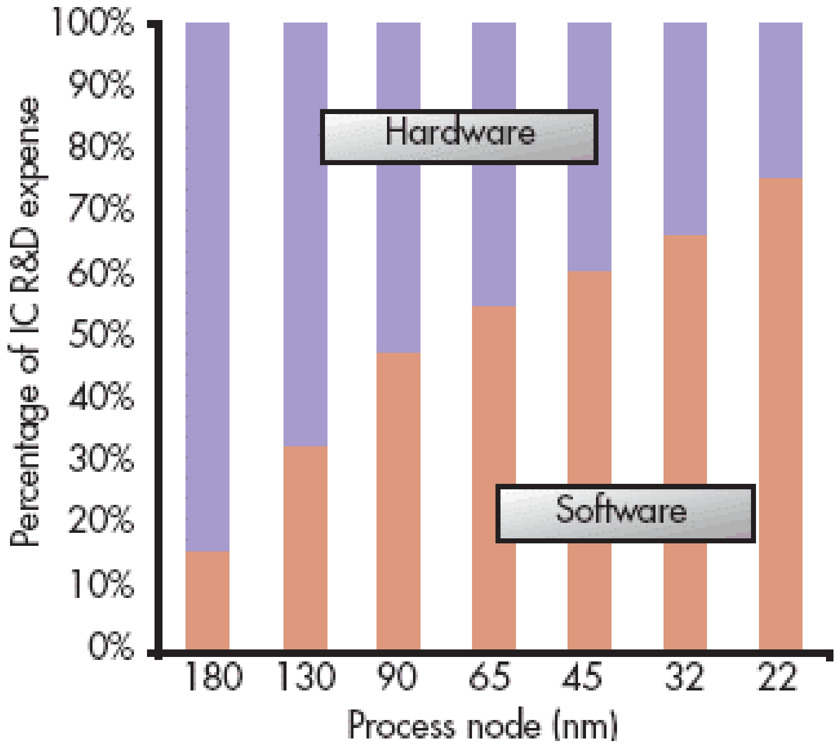
- **Modern MPSoCs: Entire Systems integrated on a single die**



Introduction: Context

- **Software complexity increases event faster than hardware**





Introduction: Context

- **Software complexity increases event faster than hardware**

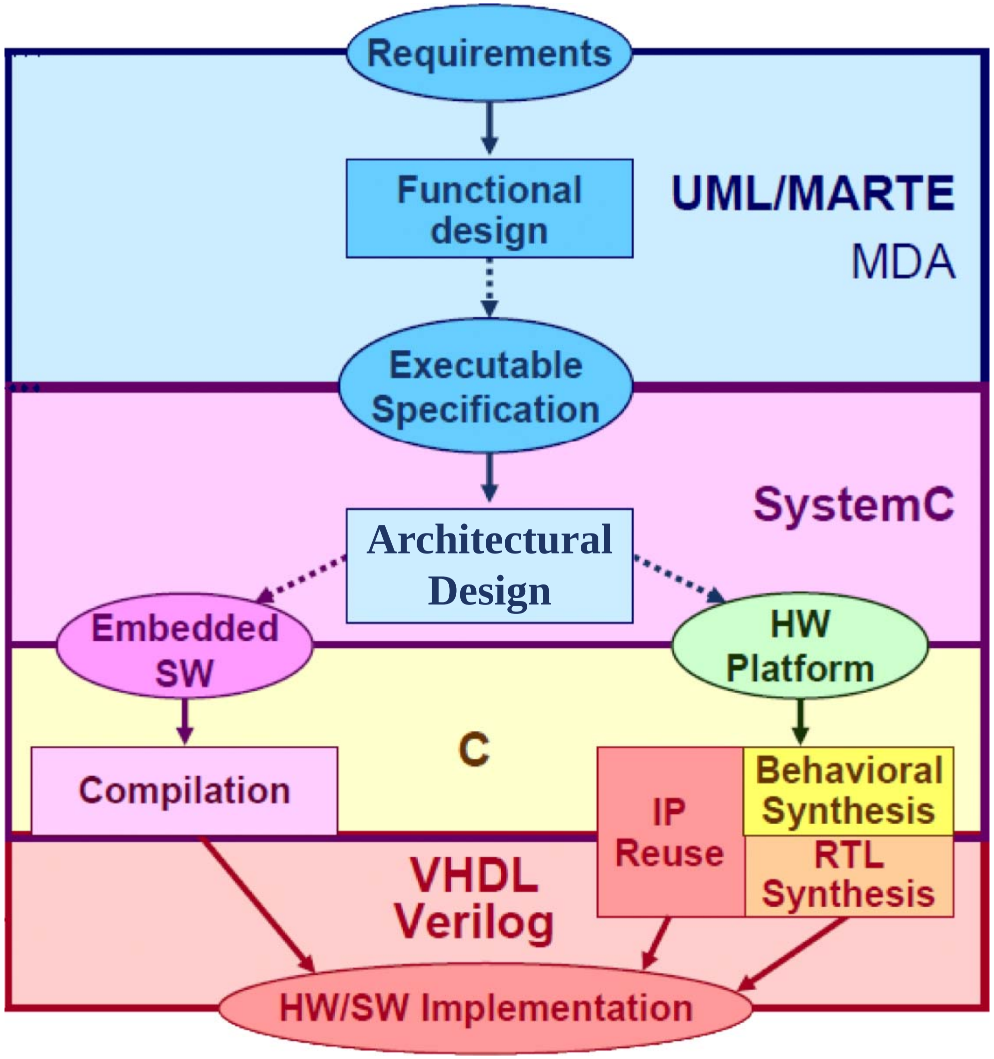
- **F-4**
  - **$20 Milion**
  - **No firmware**

- **F-22**
  - **$257 Milion**
  - **Half of costs due to Embedded SW.**
  - **Reliability problems.**

Jack Ganssle, Substract software costs by adding CPUS. EE Times, April 2005

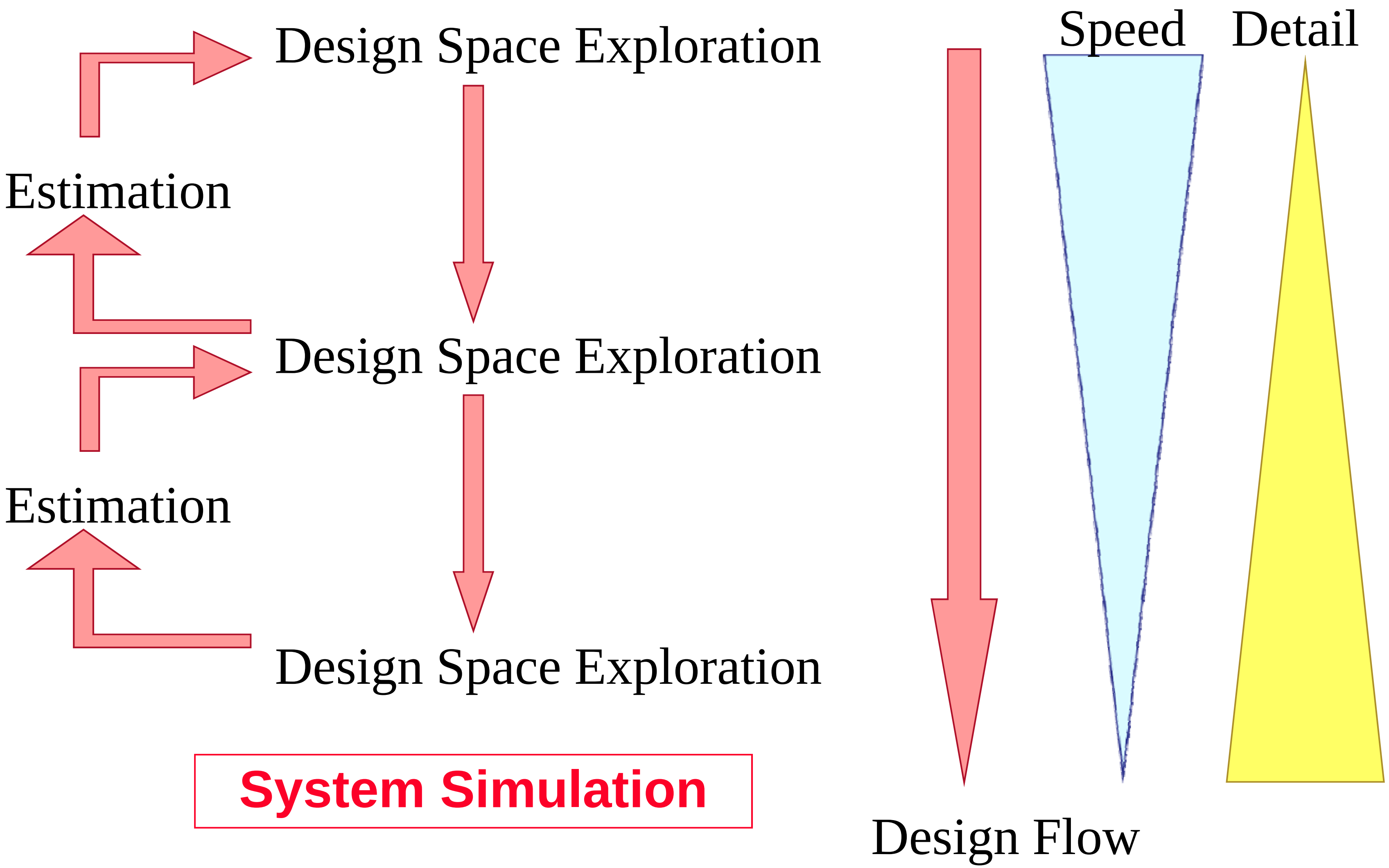
Introduction: Context

- **HW/SW Embedded Systems Co-Design Flow**



Introduction: Motivation

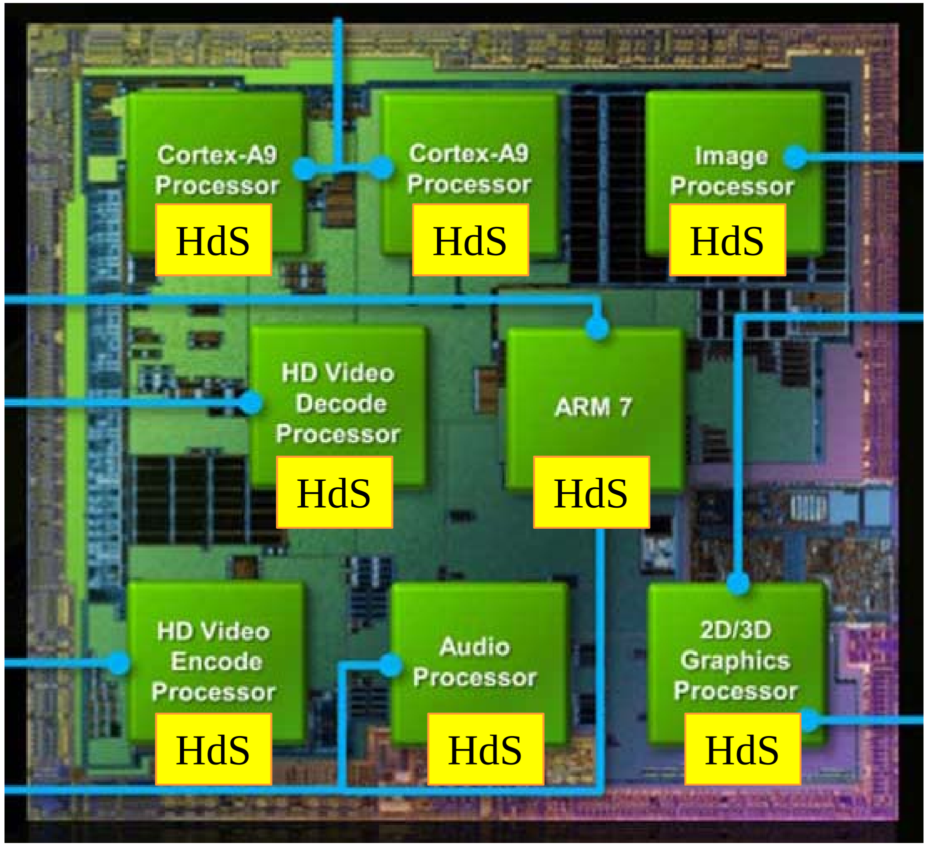
- **HW/SW Embedded Systems Co-Design Flow**



Introduction: Motivation

- **Validation of System Simulation to solve HdS problems.**

Nvidia Tegra 2

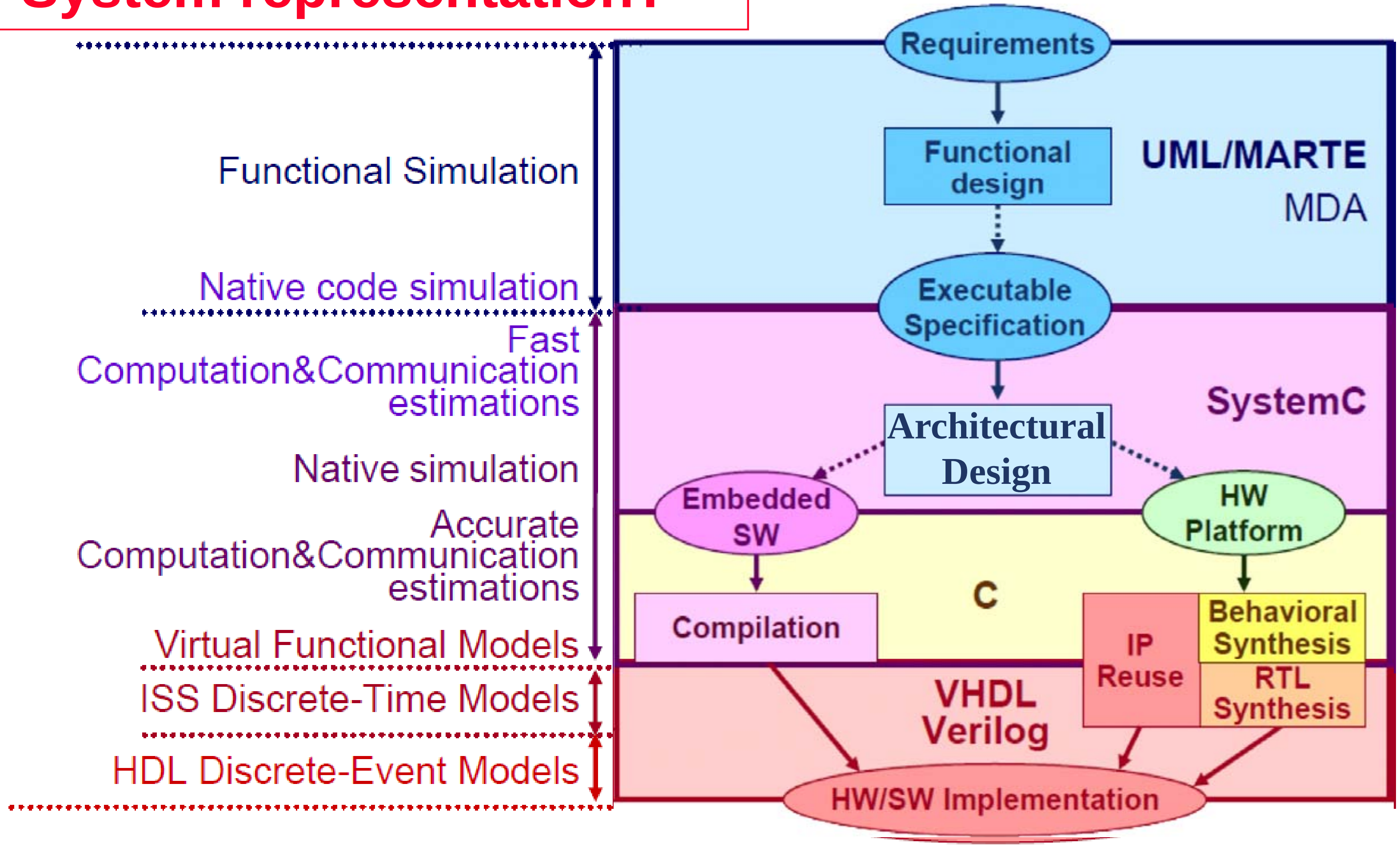


- **HW IP designers.**
- **MPSoC designers.**
- **SW designers.**

Introduction: Motivation

- **MPSoC System Simulation Technologies**

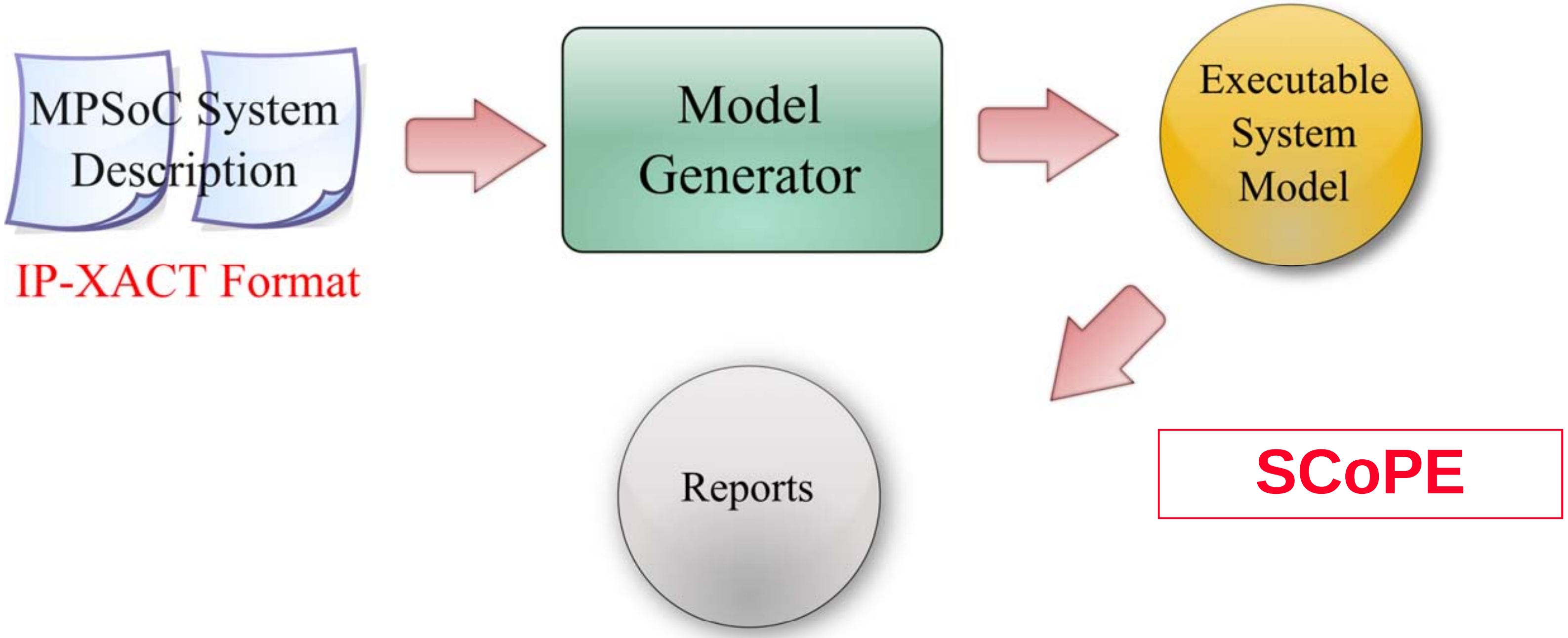
System representation?



Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

Introduction: Objectives

Automatic Generation of HdS System Models for System Simulation using IP-XACT

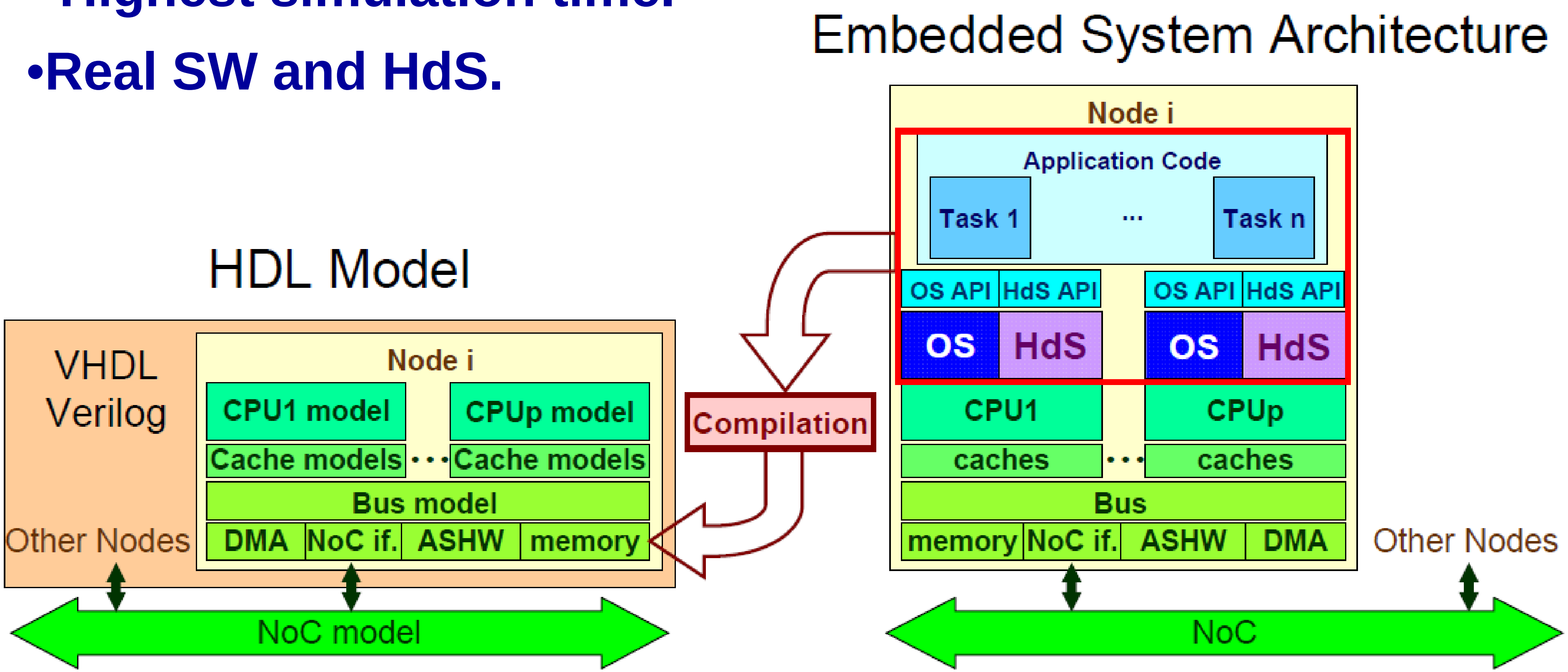


Valid for any design stage and simulation technology

Technologies for HdS System Simulation

- **HDL Simulation**

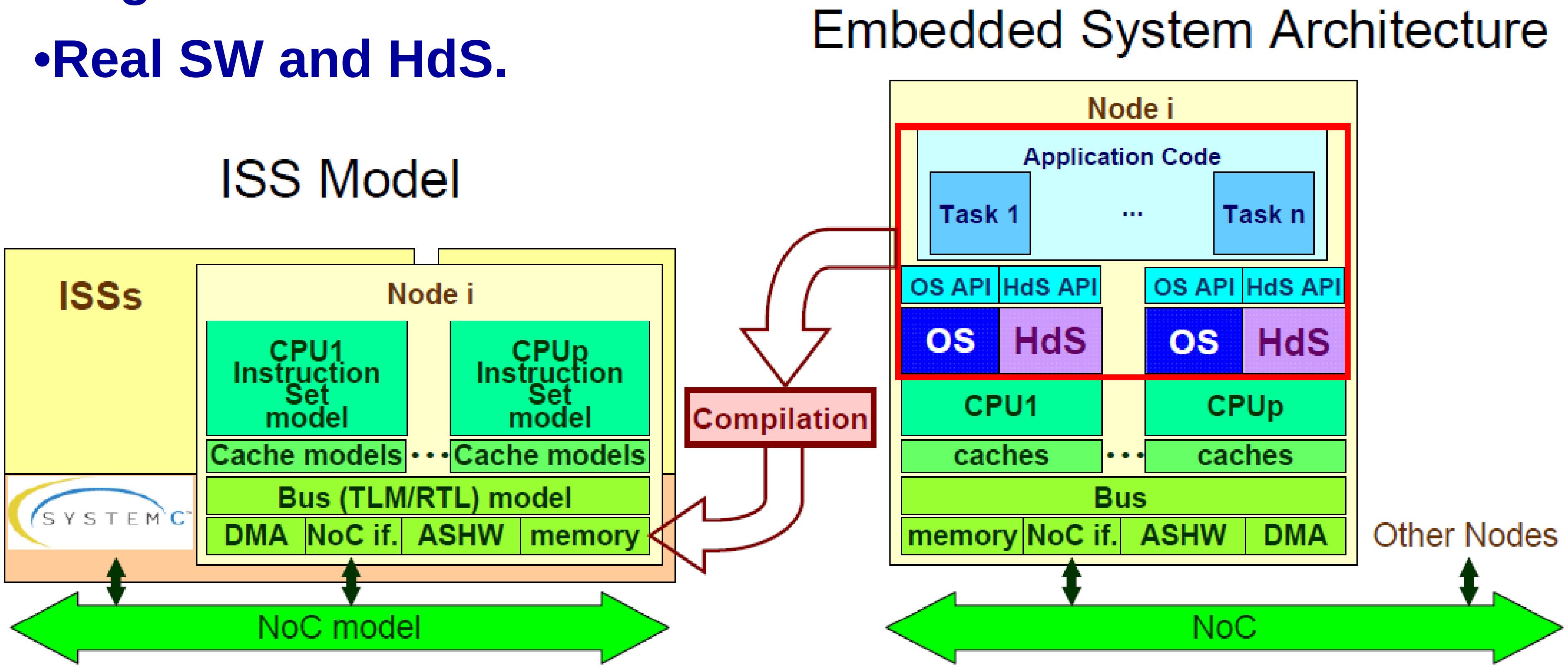
- **Highest accuracy.**
- **Highest simulation time.**
- **Real SW and HdS.**



Technologies for HdS System Simulation

- **ISS Simulation**

- **Cicle accuracy.**
- **High simulation times.**
- **Real SW and HdS.**

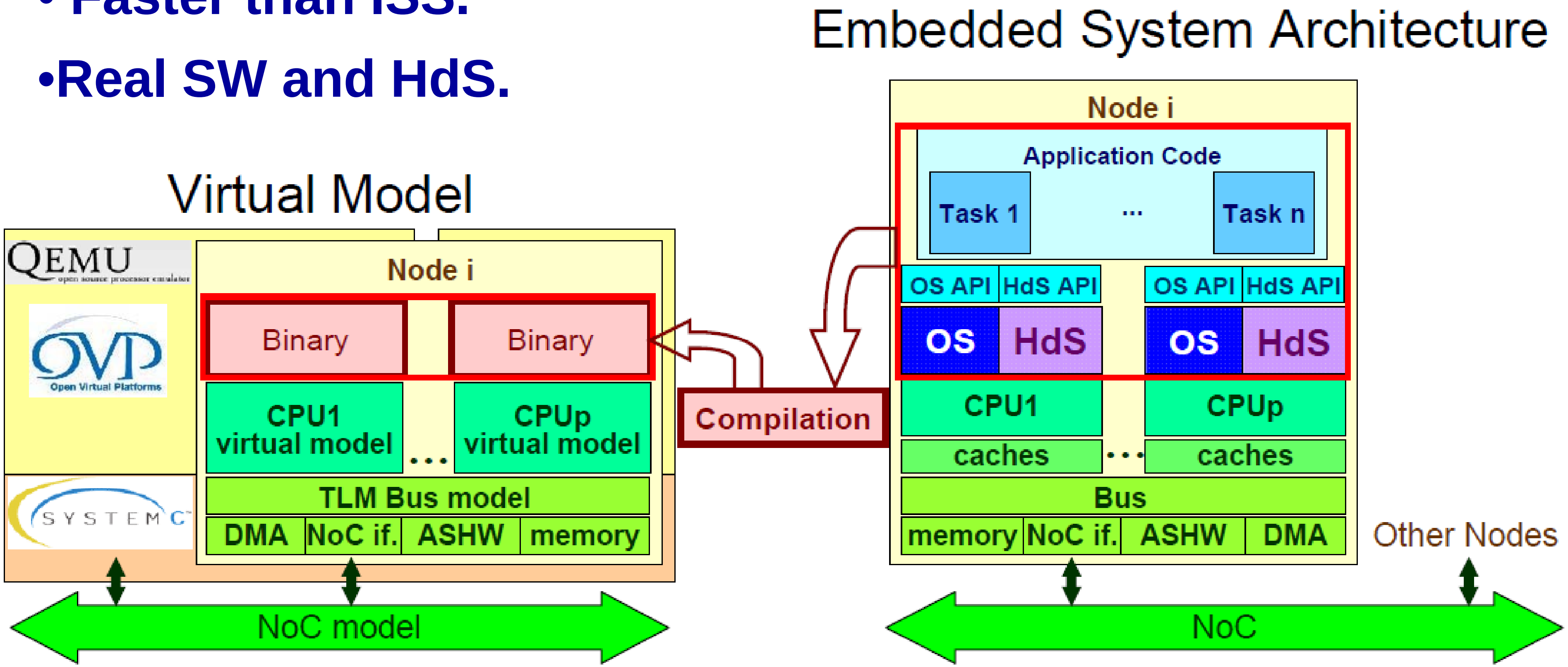


Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

Technologies for HdS System Simulation

- **Virtualization**

- **Performance estimation is still an open line of research.**
- **Faster than ISS.**
- **Real SW and HdS.**

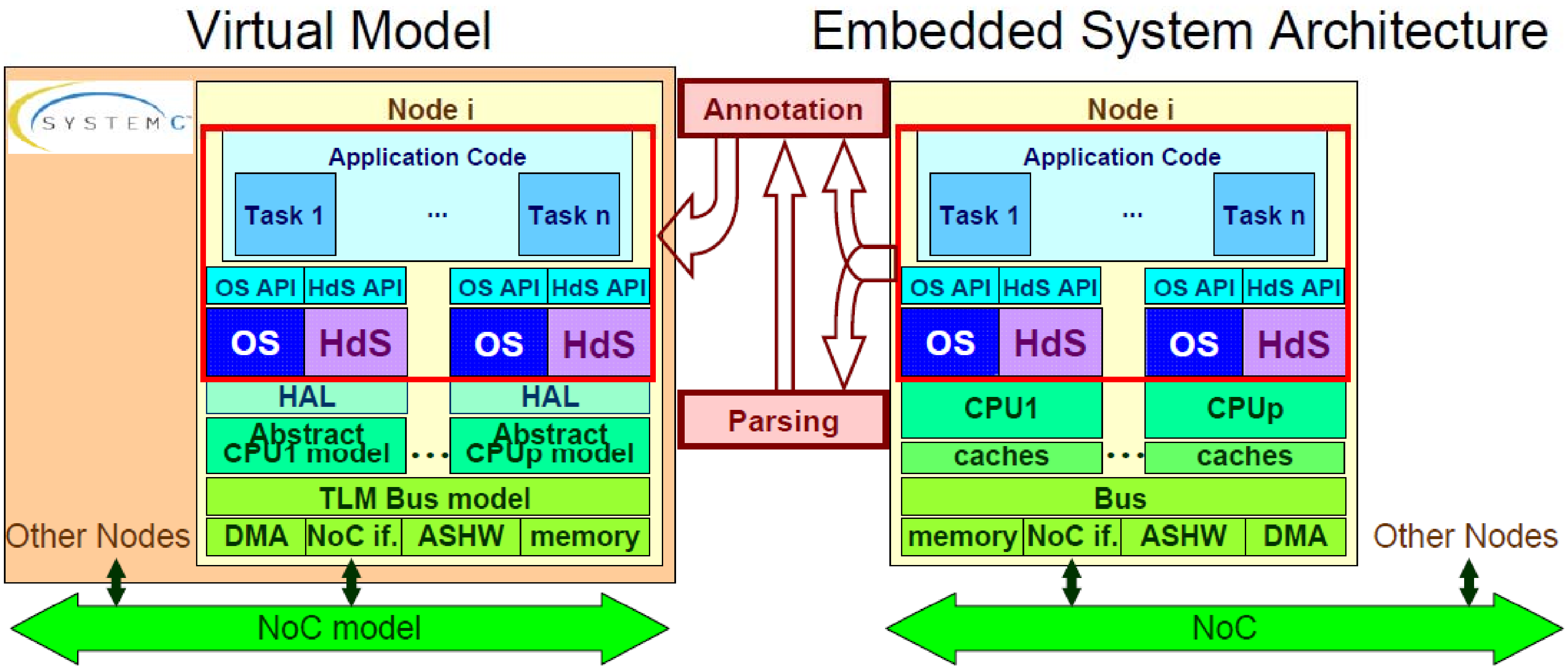


Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

Technologies for HdS System Simulation

- **Native Simulation based on HAL API**

- **Good accuracy.**
- **Real SW and HdS.**
- **Faster than ISS or Virtualization.**

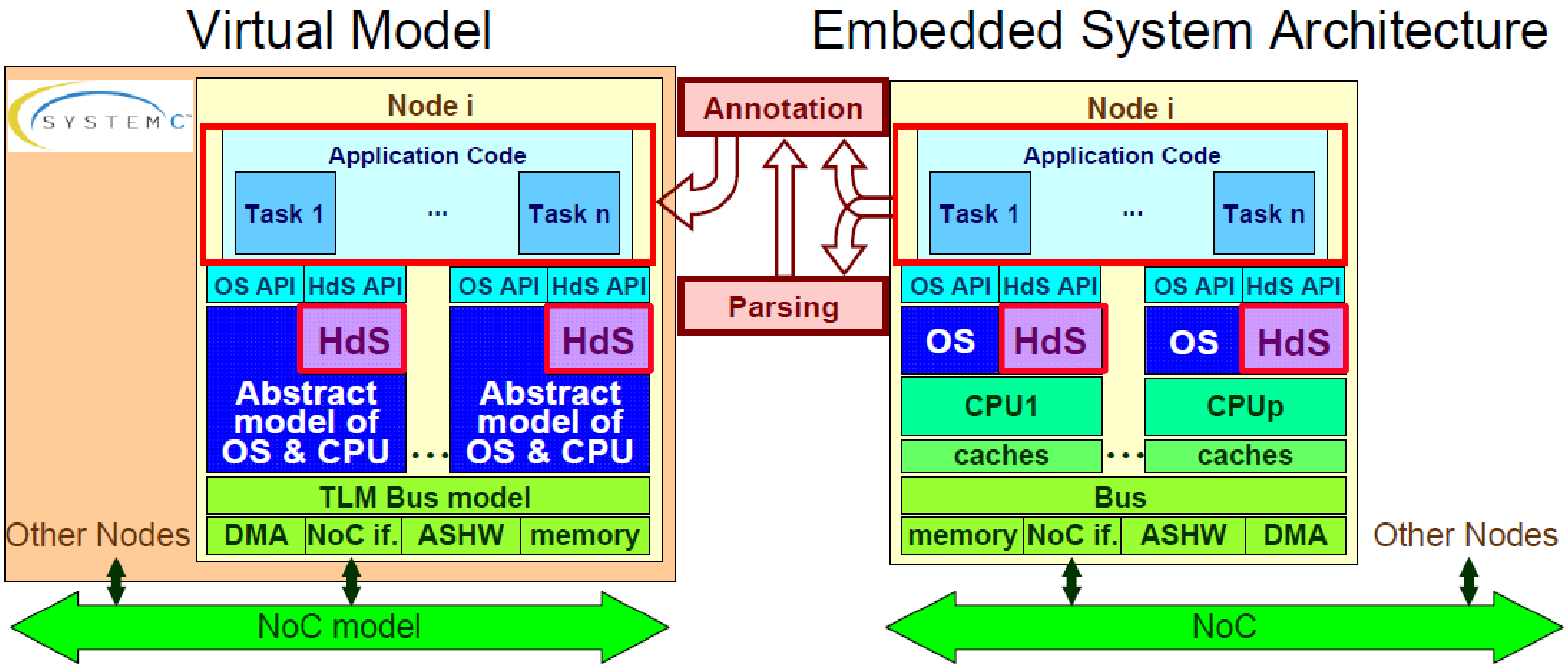


Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

Technologies for HdS System Simulation

- **Native Simulation based on OS API**

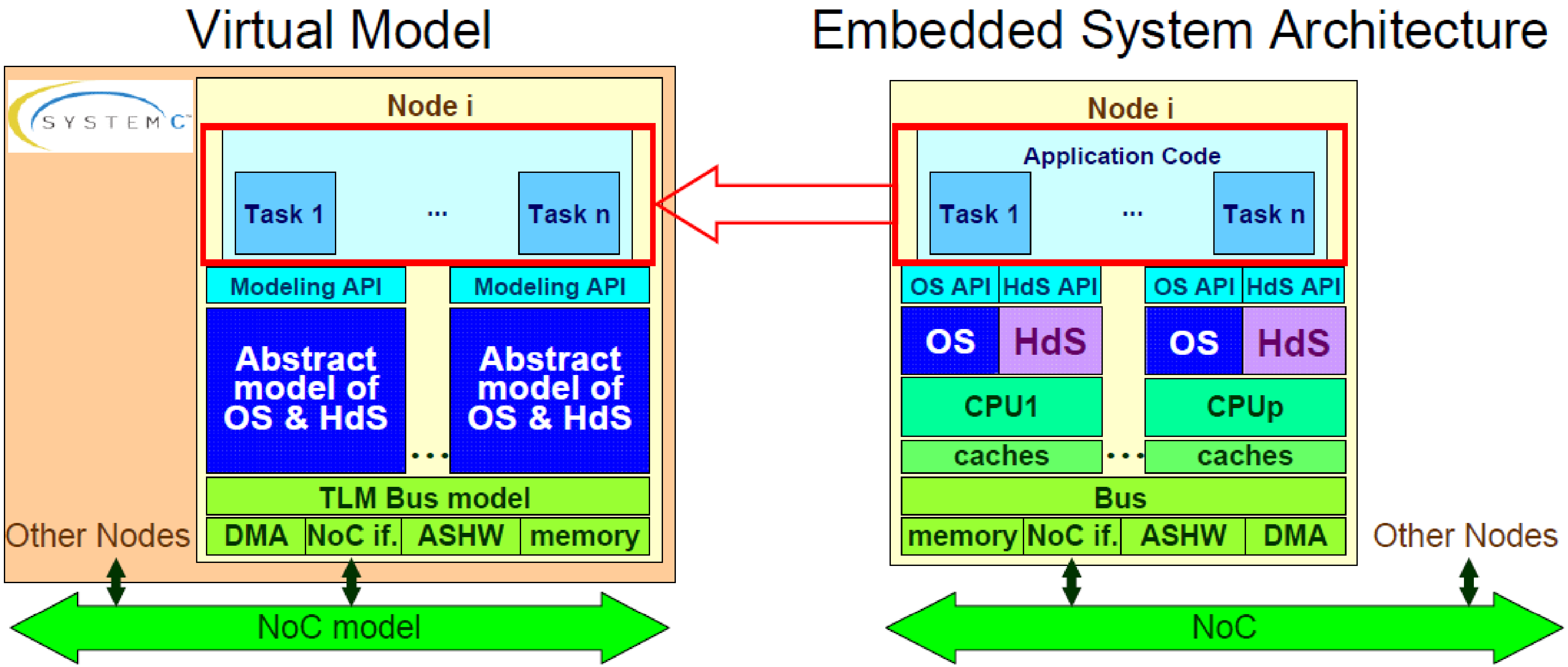
- **Good accuracy.**
- **Faster than ISS or Virtualization.**
- **Real SW app and HdS.**
- **Abstract OS model.**



Technologies for HdS System Simulation

- **Functional Simulation**

- **No performance estimation.**
- **HdS is not considered.**



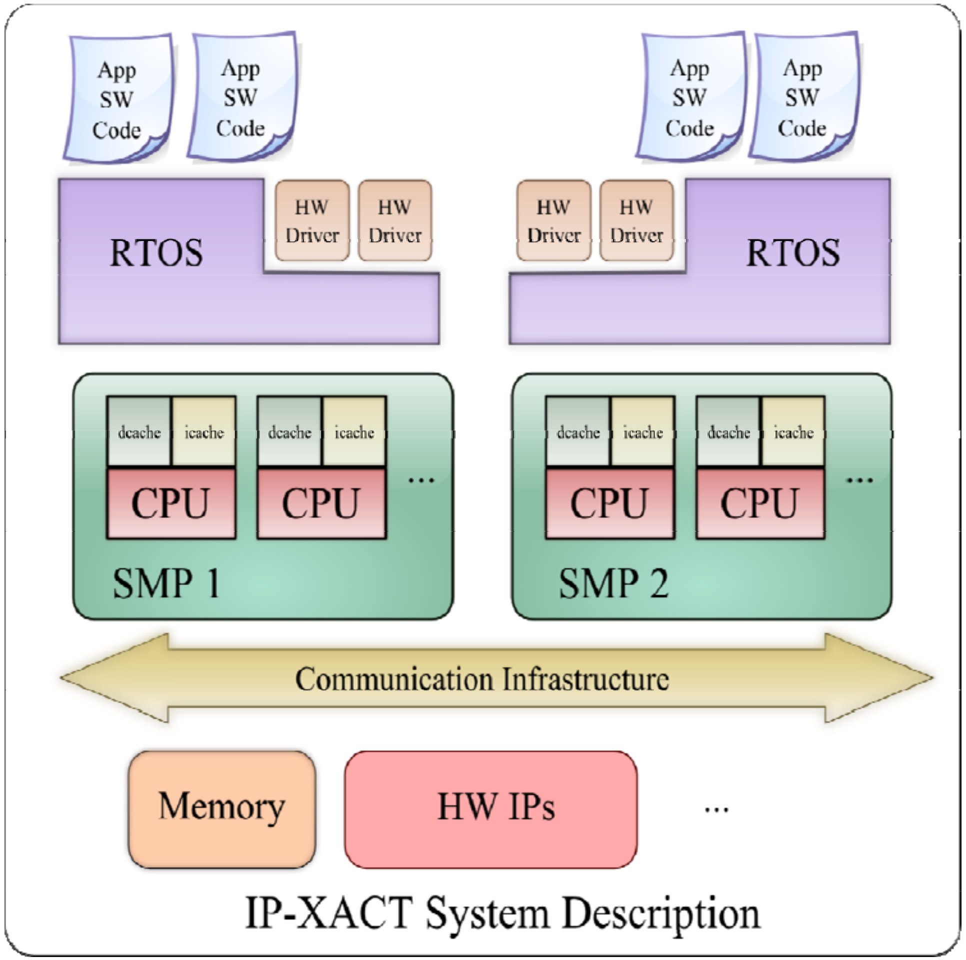
Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

Technologies for HdS System Simulation

- Each simulation technology is appropriate for a different design stage.
- In almost all the approaches real HdS can be integrated in the system models.
  - No HW/SW partition in functional simulation

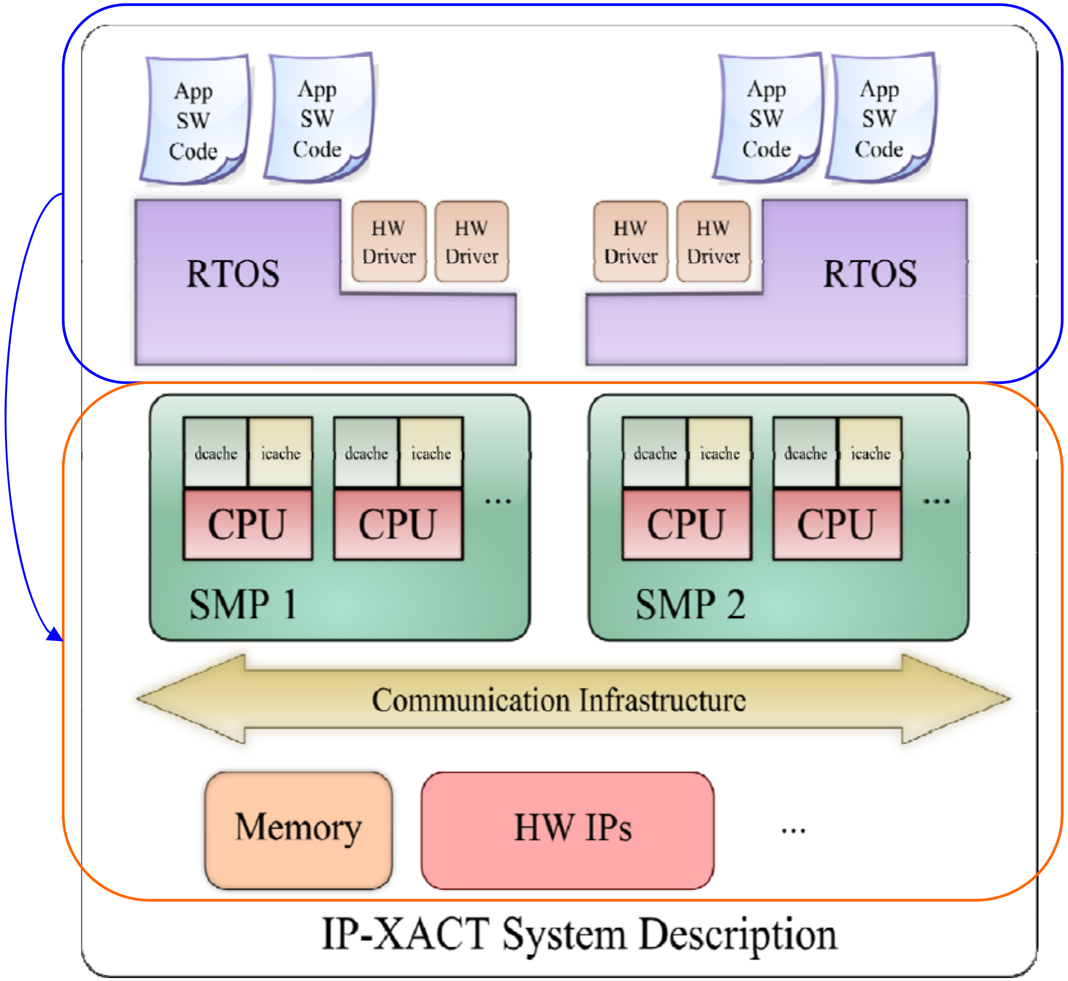
IP-XACT System Description Methodology

- **IP-XACT System Description Methodology**



IP-XACT System Description Methodology

- **IP-XACT System Description Methodology**



XML Format
- SW Applications
- SW Platform
- Mappings
- Configurability

IP-XACT
- HW platform
- HdS

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

IP-XACT System Description Methodology

- IP-XACT Extensions to describe HdS
  - In spirit:component files:

```
<spirit:vendorExtensions>

        <spirit:HdSComponents>

            <spirit:HdSComponent>
                <spirit:HdSRef spirit:vendor="uca" spirit:library="date_workshop"
spirit:name="thermal_driver" spirit:version="1.0"/>
                <spirit:node>/dev/iomodule0</spirit:node>
                <spirit:active>true</spirit:active>
            <spirit:HdSComponent>

            <spirit:HdSComponent>
                <spirit:HdSRef spirit:vendor="uca" spirit:library=" date_workshop "
spirit:name=" thermal_driver " spirit:version="0.9"/>
                <spirit:node>/dev/iomodule0</spirit:node>
                <spirit:active>false</spirit:active>
            </spirit:HdSComponent>

          </spirit:HdSComponents>

</spirit:vendorExtensions>
```

IP-XACT System Description Methodology

- IP-XACT Extensions to describe HdS
  - New **spirit:HdS** file type:

```
<spirit:HdS>
    <spirit:vendor>uca</spirit:vendor>
    <spirit:library>date_workshop</spirit:library>
    <spirit:name>thermal_driver</spirit:name>
    <spirit:version>1.0</spirit:version>

    <spirit:HdSType>Char</spirit:HdSType>

    <spirit:OS>
           <spirit:name>Linux</spirit:name>
           <spirit:version>2.6.27</spirit:version>
           <spirit:major>249</spirit:major>
           <spirit:minor>1</spirit:minor>
           <spirit:magic>0xFE</spirit:magic>
           <spirit:irq_number>13</spirit:irq_number>
    </spirit:OS>

    <spirit:HW_comm>iomemory</spirit:HW_comm>
```

```
<spirit:SW_services>
   <spirit:SW_service>
           <spirit:entryPoint>read</spirit:entryPoint>
           <spirit:returnType>int</spirit:returnType>
           <spirit:disable>false</spirit:disable>
           <spirit:arguments>
              <spirit:argument spirit:dataType="int">
                <spirit:name>fd</spirit:name>
              </spirit:argument>

              <spirit:argument spirit:dataType="void">
                <spirit:name>buf</spirit:name>
              </spirit:argument>

              <spirit:argument spirit:dataType="size_t">
                <spirit:name>count</spirit:name>
              </spirit:argument>

           </spirit:arguments>
    </spirit:SW_service>
```

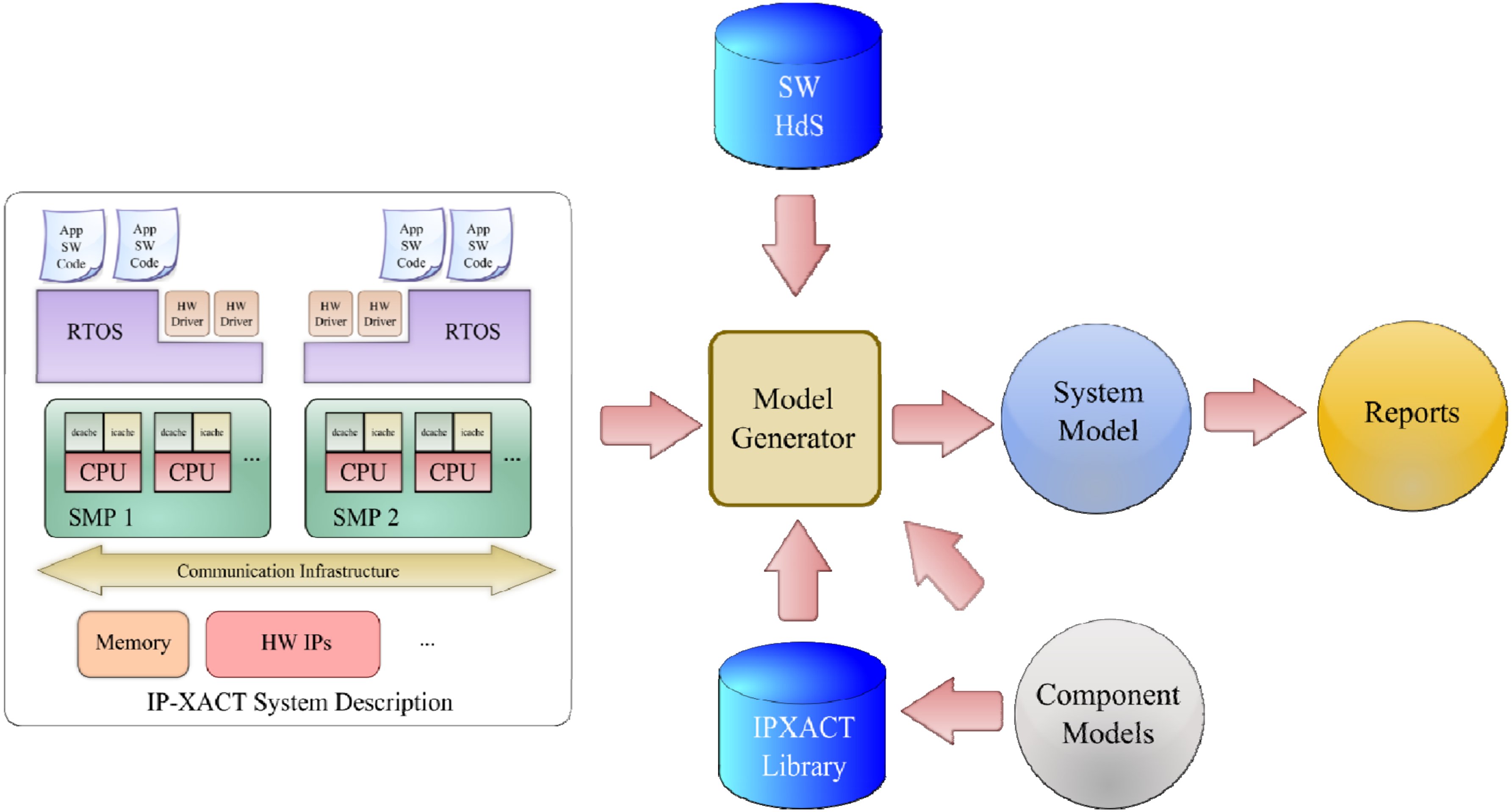
IP-XACT System Description Methodology

- **IP-XACT Extensions to describe HdS**
  - **New spirit:HdS file type:**

```
 <spirit:SW_service>


<spirit:entryPoint>ioctl</spirit:entryPoint>
   <spirit:returnType>int</spirit:returnType>
   <spirit:disable>false</spirit:disable>
   <spirit:arguments>
       <spirit:argument spirit:dataType="int">
          <spirit:name>cmd</spirit:name>

          <spirit:possible_values>
             <spirit:possible_value>
               <spirit:alias>IOCSHLIMIT
                      </spirit:alias>
               <spirit:value>1</spirit:value>
             </spirit:possible_value>
          </spirit:possible_values>
       </spirit:argument>
                                  …
    </spirit:arguments>

 </spirit:SW_service>
                                  …
</spirit:SW_services>
```

```
<spirit:filesets>
    <spirit:fileset>
        <spirit:name>sourceCode</spirit:name>
        <spirit:file>
            <spirit:name>io_driver.c</spirit:name>
            <spirit:fileType>cSource</spirit:fileType>
          </spirit:file>
          <spirit:file>
            <spirit:name>io_driver.h</spirit:name>
            <spirit:fileType>cSource</spirit:fileType>
          </spirit:file>
        <spirit:file>
            <spirit:name>io_driver.ko</spirit:name>
               <spirit:fileType>
                  KernelModule
               </spirit:fileType>
          </spirit:file>
      </spirit:fileset>
  </spirit:filesets>

</spirit:HdS>
```

Automatic Generation of HdS System Models



Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

Automatic Generation of HdS System Models

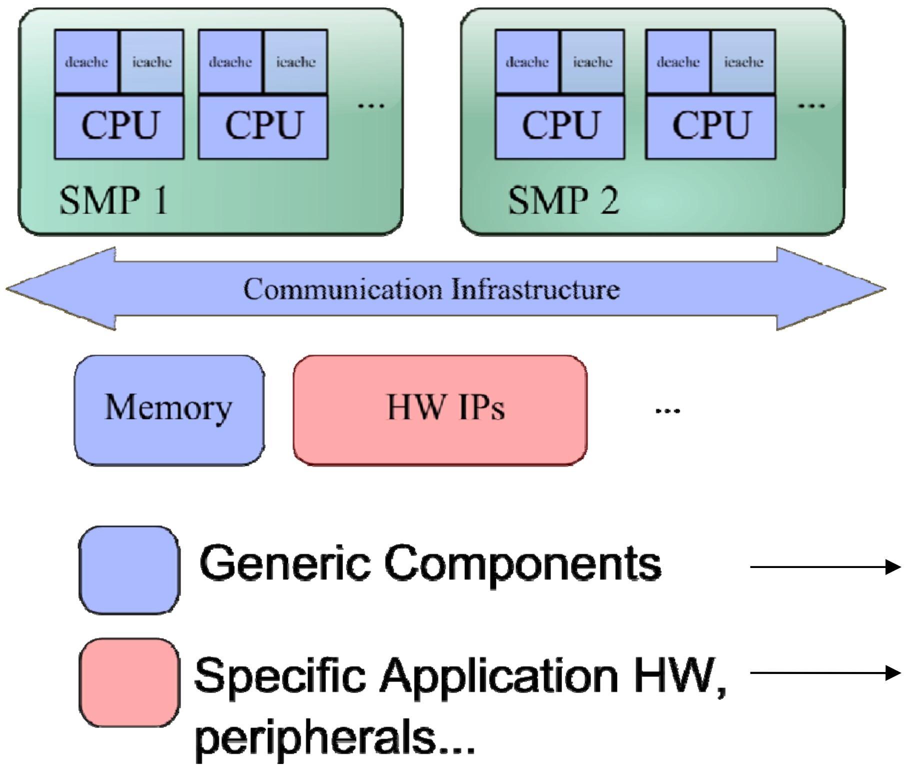
- **SCoPE: SystemC Cosimulation and Performance Estimation**
  - **Native Simulation based on OS API.**
  - **Provide sufficient accurate metrics (less than 5% error in representative testcases.)**
    - **Time and power estimation**
    - **Consider SW & HW effects.**
    - **Data and instruction caches fast and accurate modeling.**
    - **L2 cache modeling.**

Automatic Generation of HdS System Models

- **Automatic Generation of SystemC System Models using SCoPE**
  - **Software platform model generation**
  - **Hardware platform model generation**

RTOS Abstract Model

Linux Drivers



SCIPIV TAGS

SCoPE HW Abstract Models

SystemC model referenced in the IP-XACT component description

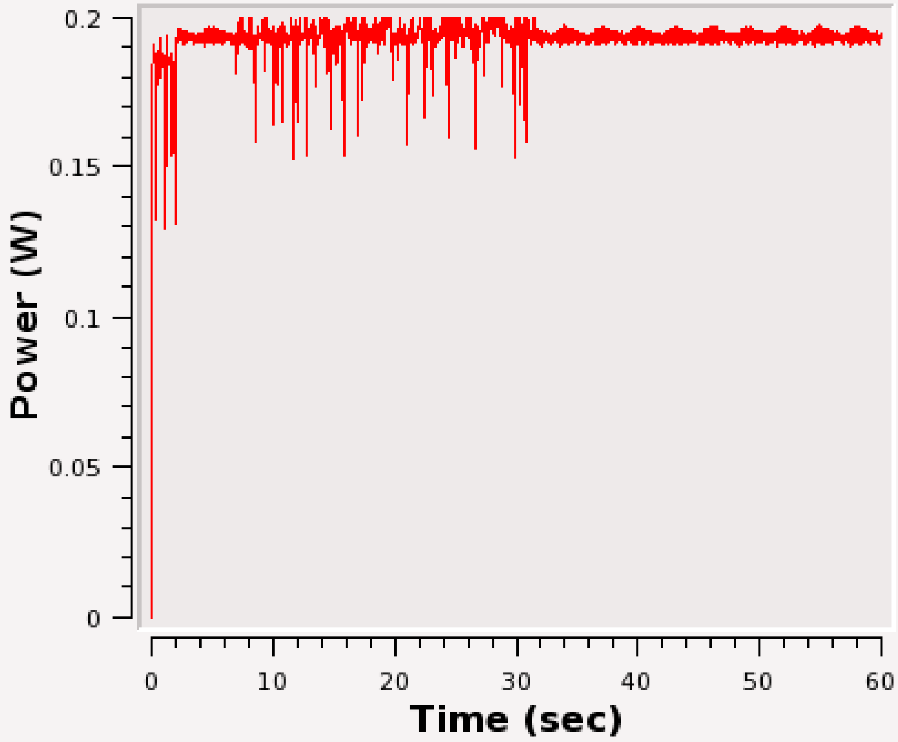
Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar

Automatic Generation of HdS System Models

- **Modeling Linux Drivers in SCoPE**
  - **Linux drivers are inserted without modification.**
  - **The impact of the driver source code on the performance and power consumption is neglected.**
  - **Hardware access functions access bus model.**
    - **The impact on the performance and power consumption is modeled.**

Automatic Generation of HdS System Models

- **SCoPE Reports**



CPU Info

Power (W)	0.192903
Core Energy (nJ)	1.15645e+10
I-Cache misses	2176546
I-Cache Energy (nJ)	17433850132
D-Cache misses	597
D-Cache write backs	0
D-Cache Energy (nJ)	87061840

BUS 0

Bytes	0

Automatic Generation of HdS System Models

```
Simulated time: 60 s
RTOS:  0
      Number of m_processes created:  2
      Number of m_processes destroyed:  1
      Mean process duration (process start - process end):  48.1853 sec
      Last SW execution time:  60 sec
      Process PID: 4
            Thread TID: 5, name: h264_coder, User time: 36320294392 ns
      Total User time:  84.6021 sec
      Total Kernel time:  0.428415 sec
processor_0_rtos_0_0
      Number of thread switches:  8502
      Number of context switches: 1
      Running time: 24233780832 ns
      Use of cpu: 40.3896%
      Instructions executed: 2423375885
      Instruction cache misses: 942876
      Data cache hits: 0
      Data cache misses: 0
      Data cache write backs: 0
      Core Energy: 7.27013e+09 nJ
      Core Power: 121.169 mW
      Instruction Cache Energy: 1.21216e+09 nJ
      Data Cache Energy: 0 nJ
      Instruction Cache Power: 20.2027 mW
      Data Cache Power: 0 mW
      Bus access time:  796300000 ns
      Bus transfers:  31852000 bytes
      Idle time:  34959919168 ns
      Stall time:  0 ns
      Number of interrupts:  5999
      Total instruction miss transfers: 9884
      Total data miss transfers: 0
```

Conclusions

- IP-XACT standard can be used to automatically integrate hardware components in system models
  - Extensions are needed for HdS.
  - Eventually for SW and mappings.
- IP-XACT System Description methodology.
  - Independent of any language and vendor.
- Functional identification requires using SCIPIV tags.
- Automatic Generation of HdS System Models from IP-XACT "Extended" descriptions has been developed using SCoPE native simulator.

Thank you for Your Attention

We value your opinion
and questions

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar