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
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
Introduction: Context

- Modern MPSoCs: Entire Systems integrated on a single die
Introduction: Context

- Software complexity increases even faster than hardware
Introduction: Context

- Software complexity increases event faster than hardware

- F-4
  - $20 Million
  - No firmware

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

Jack Ganssle, Subtract 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

Design Space Exploration

Estimation

Design Space Exploration

System Simulation

Design Flow

Speed

Detail
Introduction: Motivation

- Validation of System Simulation to solve HdS problems.

- HW IP designers.

- MPSoC designers.

- SW designers.

Nvidia Tegra 2
Introduction: Motivation

- MPSoC System Simulation Technologies

System representation?
Introduction: Objectives

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

Valid for any design stage and simulation technology

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
Technologies for HdS System Simulation

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

Embedded System Architecture

HDL Model

VHDL

Verilog

Node i

CPU1 model

CPU-Up model

Cache models

Cache models

Bus model

DMA

NoC if

ASHW

memory

Compilation

Node i

Application Code

Task 1

... Task n

OS API

HdS API

OS API

HdS API

OS

HdS

OS

HdS

CPU1

CPU-Up

caches

caches

Bus

memory

NoC if

ASHW

DMA

Other Nodes

NoC

Other Nodes

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
Technologies for HdS System Simulation

- ISS Simulation
  - Cicle accuracy.
  - High simulation times.
  - Real SW and HdS.

ISS Model

Embedded System Architecture

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
Technologies for HdS System Simulation

- Virtualization
  - Performance estimation is still an open line of research.
  - Faster than ISS.
  - Real SW and HdS.
Technologies for HdS System Simulation

• Native Simulation based on HAL API

• Good accuracy.

• Faster than ISS or Virtualization.

• Real SW and HdS. 

Virtual Model

Embedded System Architecture

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
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.

Virtual Model

Embedded System Architecture

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
Technologies for HdS System Simulation

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

Virtual Model

Embedded System Architecture

---

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
• 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

- SW Applications
- SW Platform
- Mappings
- Configurability

XML Format

- HW platform
- HdS

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
IP-XACT System Description Methodology

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

  ```xml
  <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>
  ```

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
IP-XACT System Description Methodology

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

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

<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>
</spirit:SW_services>
```

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
IP-XACT System Description Methodology

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

```xml
<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:HdS>
  <spirit:filessets>
    <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:fileset>
      <spirit:fileset>
        <spirit:name>io_driver.h</spirit:name>
        <spirit:fileType>cSource</spirit:fileType>
      </spirit:fileset>
      <spirit:fileset>
        <spirit:name>io_driver.ko</spirit:name>
        <spirit:fileType>KernelModule</spirit:fileType>
      </spirit:fileset>
    </spirit:filesets>
  </spirit:filessets>
</spirit:HdS>
```

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
Automatic Generation of HdS System Models

Model Generator

System Model

Reports

IPXACT Library

Component Models

IP-XACT System Description

Communication Infrastructure

CPU

CPU

CPU

CPU

RTOS

RTOS

Application SW Code

Application SW Code

Application SW Code

Application SW Code

Hw Driver

Hw Driver

Hw Driver

Hw Driver

SMP 1

SMP 2

Memory

HW IPs

Daniel Calvo, Patricia Botella, Héctor Posadas, Pablo Sánchez, Eugenio Villar
© Microelectronics Engineering Group, University of Cantabria
• **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 HsS 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

Generic Components
Specific Application HW, peripherals...

SCoPE HW Abstract Models
SystemC model referenced in the IP-XACT component description
• **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

![Graph showing Power (W) vs Time (sec) with CPU Info table]
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): 48.1852 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: 24233780632 ns
Use of cpu: 40.3896%
Instructions executed: 24233780632
Instruction cache misses: 942876
Data cache hits: 0
Data cache misses: 0
Data cache write backs: 0
Core Energy: 7.47012e+00 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: 5996
Total instruction miss transfers: 9084
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