Heterogeneous Virtual Prototyping for IoT Applications

Paul Ehrlich, paul.ehrlich@coseda-tech.com
Karsten Einwich, karsten.einwich@coseda-tech.com
COSEDA Technologies GmbH
Mark Burton, mark.burton@greensocs.com
GreenSoCs
Luc Michel, luc.michel@antfield.fr
Antfield S.A.S
Agenda

• Introduction and Motivation
• Digital Platform integration
• Digital Analog Interface Modelling
• Physical domain Modelling
• Application Example
Heterogeneous Hard-Software System
Challenges for Virtual Prototyping of heterogeneous Systems

- Simulation Performance
- Highly configurable systems
- Combining different Modelling Domains
- Numerous application scenarios
- Hard-Software Codesign
- Closed loops
Digital Platform Modelling Requirements

- Complex Multi Core / Multi Processor Networks
- Complex Bus Architectures and Arbitration Algorithm
- Tight Hardware/Software dependencies

- High simulation performance requirements regardless of complex processor architectures
Qemu SystemC Integration

• QEMU is a generic and open source machine emulator and virtualizer
• A wide variety of processor architectures and micro-controllers and boards are available (e.g. ARM, PPC, x86, Mips, MPCxxxx, etc...)
• Qbox uses Qemu to provide SystemC TLM 2.0 based CPU models
• Runs extremely fast: MULTI Thread Qemu
  – A massive speed improvement for Qemu to take advantage of multi-core hosts
  – Multiple CPU cores can run in separate threads
• Full SystemC/TLM integration (including DMI and Quantum support)
• Flexibly allow different parts of the system in Qemu or SystemC.
The problem multi-thread simulation in SystemC

SystemC runs out of events…
(even though model 1 and 2 are still active)

Simulation dies.
Synchronization of CPU core instances

Thread 1
Model 1

Coming in SystemC 2.3.2 (look for a public announcement here at DVCon)

Thread 0
SystemC Simulation

Thread 2
Model 2

Single shared semaphore.
Triggered from async_request_update
Digital – Analog Interface

- Loosely coupled interaction
SystemC AMS – TLM Interaction

- Synchronization via method process
Physical Domain Modelling

• Using ELN elements as base classes
• ELN provides basis to describe any linear DAE system
• Equation setup based on Kirchoff’s law
• Applying analogy relations
Analogy Relation

\[ F_m + F_r + F_n = F \]
\[ F_m = m \cdot \text{vel} \]
\[ F_n = k \cdot s = k \cdot \int \text{vel} \cdot dt \]
\[ F_r = m \cdot \text{vel} \]

\[ i = C \cdot \dot{v} \]
\[ i = \frac{1}{L} \int v \cdot dt \]
\[ i = \frac{1}{R} \cdot v \]
Analogy Relation

- Direct mapping of physical components to electrical components possible
- Different analogy relations are possible
- It should be prevented to connect nodes of different physical properties
- The analogy relation should be selected in a way, that the wiring of the components remains
Analogy Relation – Realization within SystemC AMS

```cpp
#include <systemc-ams>

class spring : public sca_eln::sca_l
{
public:
    spring(const char* nm, double k) : sca_eln::sca_l(nm, 1.0/k)
    {}
};

namespace sca_ln
{
    class sca_node_base : public sca_eln::sca_node
    {...};

    template<class PDOMAIN>
    class sca_node : public sca_ln::sca_node_base,
    public sca_ln::sca_node_if<PDOMAIN>
    {...};
}

sca_ln::sca_node<sca_translational> mech_node;

class spring : public sca_eln::sca_l
{
public:
    sc_ln::sca_terminal<sca_translational> t1;
    ...
};
```
Demonstrator

- Thermal house model
- Embraces digital hard- and software, electrical as well as thermic components
- Control unit is realized by an ARM CPU
- The dynamics of the electrical and thermal components are modelled
Demonstrator
Demonstrator - Results

3 days are simulated in 81 sec with simplified uC program
Summary

• More and more systems require embracing different domains
• Understanding the tight interaction is essential
• Qemu based processor integration enables fast hard-software debugging
• SystemC AMS can be used as a basis to create physical domain libraries
• Abstract modelling is the enabler to create virtual platforms which permit the simulation of application scenarios
Questions