Simulation and Debug of Mixed Signal Virtual Platforms for Hardware-Software co-development

Vincent Motel, Cadence Design Systems, Inc.
Alexandre Roybier, Cadence Design Systems, Inc.
Serge Imbert, Cadence Design Systems, Inc.
Agenda

• INTRODUCTION

• MIXED SIGNAL VIRTUAL PLATFORMS

• MODELING ASPECT

• TOOL SUPPORT / EXAMPLE

• CONCLUSION
Introduction
INTRODUCTION

MIXED SIGNAL VIRTUAL PLATFORMS

MODELING ASPECT

TOOL SUPPORT / EXAMPLE

CONCLUSION
Virtual Platforms

• High-level software models of the hardware
• Run embedded software, verify hardware/software interactions
• Explore architecture, start software development earlier

- Abstract, behavioral models of relevant components only
- Quick to develop
- Inexpensive to duplicate
- Unique debug capabilities
Mixed signal in virtual platforms

Most systems are mixed signal => emerging demand from architecture and software teams to add analog fidelity

• Development and validation of software depending on analog behavior
  – Smart control of analog
  – Understanding & optimizing system sensitivity,
  – Aggressive power optimizations
  – Reduce the time and cost of development

• Architecture exploration
  – Cost/performance tradeoff
  – Less expensive analog sensors but strong post processing algorithms

• System validation, based on realistic scenarios
• INTRODUCTION

• MIXED SIGNAL VIRTUAL PLATFORMS

• MODELING ASPECT

• TOOL SUPPORT / EXAMPLE

• CONCLUSION
Modeling Aspect

... But abstraction is the key aspect

Abstraction level

Discrete time

Real Number Modeling : SystemVerilog, VHDL, WReal

VHDL-AMS

Verilog-AMS (electrical)

Spice

Verilog-A

Continuous time

Non Conservative

TDF

LSF

SystemC-AMS

ELN

Conservative
Modeling Aspect

• Software: approx. 100Mhz (+/- 10x)

• Hardware: 2 possibilities for analog modeling

Electrical
  – Conservative analog engine
  – Kirchhoff laws to be solved
  – Continuous in time
  – Slow

RNM
  – Non conservative digital engine
  – No convergence problem
  – Continuous in value and discrete in time
  – Fast
Modeling Aspect
example : DAC / ADC with RNM

Example: Study of 14 bit ADC + 14 bit DAC complete transfer
- 14 bit => $2^{14}=16384$ steps to simulate all values
- 3 seconds with RNM. Hours with latest analog multi-core simulators.
Using digital engines for analog modeling

• Event-driven simulation (SystemVerilog, Verilog-AMS, VHDL-AMS...)
  – No predefined time step
  – Models at least the control part, can also model analog

• Timed Data flow (SystemC-AMS)
  – Fixed time step
  – Pre-scheduled => fast simulation
  – Appropriate for continuous, steady signal processing

• Dynamic Timed Data Flow (SystemC-AMS)
  – Dynamic change of frequency (variable time step)
  – Can handle variable data rates and occasionally changing signals
  – May require manual tuning
Reaching speed required for virtual platforms
Abstraction!

- Structural details => functional blocks
- Conservative behavior of electrical nodes => ignored
- Continuous time => discrete time
  avoid the need for differential equation solver
- Electrical values => not represented
  although it is good to know which values are represented, with
  relevant units and absolute values
• INTRODUCTION

• MIXED SIGNAL VIRTUAL PLATFORMS

• MODELING ASPECT

• TOOL SUPPORT / EXAMPLE

• CONCLUSION
Example Mixed Signal Virtual Platform

- Earth magnetic field model
- magnetic perturbation model
- motion model
- field rotation
- magnetic sensor (HMC6052)
- ADC
- sensor interface
- registers
- ESW bare-metal sw-defined compass
- Cortex® A9
- simple TLM router
- LCDC
- TLM memory
- simple timer
- DISPLAY

**physics**  **analog**  **digital and software**
Example: Modeling AMS

- AMS part of the system was implemented twice

- Both models are strictly equivalent
  - same computations
  - at the same frequency
  - from the same inputs

- Only the digital engine of the simulator is used
- Sample rate can be configured to experiment with the simulation performance aspects

```verilog
module magnetic_field_sensor(strength, direction, outa, outb);
    input strength, direction;
    wreal strength, direction;
    output outa, outb;
    wreal outa, outb;
    parameter real zfo = 1.5;
    parameter real sensitivity = 0.5 * 1e4;
    real outa_val, outb_val;
    always @(strength, direction)
        begin
            outa_val = zfo + strength * sensitivity * cos(direction);
            outb_val = zfo + strength * sensitivity * sin(direction);
        end
    assign outa = outa_val;
    assign outb = outb_val;
endmodule
```

```c++
#include "systemc.h"
#include "systemc-ams.h"

SCA_TDF_MODULE(magnetic_field_sensor) {
    public:
        sca_tdf::sca_in<double> strength; // in T
        sca_tdf::sca_in<double> direction; // in rad
        sca_tdf::sca_out<double> OUTA; // in V
        sca_tdf::sca_out<double> OUTB; // in V
    private:
        double sensitivity; // in V/T
        double zfo; // zero field output, in V
    void initialize() { }
    void processing() {
        OUTA.write(zfo + strength.read() * sensitivity * cos(direction));
        OUTB.write(zfo + strength.read() * sensitivity * sin(direction));
    }
    public:
        magnetic_field_sensor(sc_module_name name_,
            double sensitivity_ = 0.5 * 1e4, double zfo_ = 1.5)
            : sca_tdf::sca_module(name_),
            sensitivity(sensitivity_), zfo(zfo_) {
            cout << "construction of " << this->name() << endl;
        }
};
```
Example : Modeling digital

• Fast processor models
  – Trade-off between speed and accuracy
  – Typically instruction-accurate but not cycle-accurate
  – Bus protocols abstracted by transactions
  – Direct Memory Interface for even higher speed gain
  – Wrapped in SystemC modules with TLM interfaces

• Other models in SystemC / TLM 2.0

• Peripherals with registers automatically generated from register description
Tool support: languages

• Key simulator requirements
  – Support of appropriate analog and digital languages
  – Mixed language simulation

• Cadence Incisive Enterprise Simulator / AMS-Designer
  – Native support of VHDL-AMS, Verilog-AMS, SystemVerilog ...
  – Native support of SystemC allows to run SystemC-AMS
    • Open source implementation from Fraunhofer IIS
    • Recompiled for SystemC implementation of Incisive
    • Linked with models compiled with Incisive SystemC headers files and SystemC-AMS header files

=> Simple and efficient use of SystemC-AMS models
Visualization, debug and analysis: Multiple domains

- Analog

- Digital hardware
  - SystemC debug, process-aware
  - TLM 2.0 transactions

- Hardware-software interface
  - Registers
  - Interrupt signals
  - Memory maps, memory contents

- Embedded software
  - Source level symbolic debug, breakpoints, stepping
  - OS-aware debug capabilities
Visualization, debug and analysis:
All domains linked together
Visualization, debug and analysis : Structural view
Performance considerations: profiling

- High simulation performance comes primarily from appropriate modeling level selection
- Then profiling can be used
  - To optimize further
  - To chase performance bugs
- Accurate multi-language, multi-level profiler is instrumental

Model behavior

Transaction counts
- Exact counts
- Not related to actual time spent

Check granularity

Model implementation

Check coding

Function profiling
- Actual CPU usage
- Some variation between runs

Model behavior

Transaction counts
- Exact counts
- Not related to actual time spent

Check granularity

Model implementation

Check coding

Function profiling
- Actual CPU usage
- Some variation between runs
Example magnetic sensor design

Coarse-grained profiling
Profiling example magnetic sensor design

Closer examination

- Digital SystemC + Verilog-AMS Implementation (10µs sampling period)
Integration in a larger virtual platform
• INTRODUCTION

• MIXED SIGNAL VIRTUAL PLATFORMS

• MODELING ASPECT

• TOOL SUPPORT / EXAMPLE

• CONCLUSION
Conclusion

- Virtual platforms with AMS can bring valuable benefits
  - Development of embedded software for real world systems
  - Architecture exploration
  - System validation

- To make it a reality
  - Use existing languages, at appropriate abstraction level
  - Leverage existing tool capabilities: multi-level debug, advanced multi-language profiling

- Challenges
  - Multi-domain expertise needed
  - Model availability for complex analog IPs
Questions