OPEN SYSTEMC INITIATIVE

SYSTEMC AMS DAY 2011

INDUSTRY ADOPTION OF THE SYSTEMC AMS STANDARD

www.systemc.org

May 12, 2011

OPEN SYSTEMC
INITIATIVE

SystemC AMS Day 2011

Industry Adoption of the SystemC AMS Standard

May 12, 2011
Dresden

The SystemC AMS Day is sponsored by:

ARM
cadence
FORTE DESIGN SYSTEMS
Mentor Graphics
SYNOPSYS
XtremeEDA

SystemC AMS Day 2011
May 12, 2011
Hotel Königshof
Dresden, Germany

Chair:
Martin Barnasconi, NXP Semiconductors
OSCI AMS Working Group Chair

Organization and Program Committee:
Karsten Einwich, Fraunhofer IIS, Division EAS
Serge Scotti, STMicroelectronics
Wolfgang Scherr, Infineon Technologies

Published by:
The Open SystemC Initiative and
Fraunhofer Institute for Integrated Circuits IIS, Design Automation Division EAS
Printed by:
Maxroi Graphics GmbH
Copyright © The Open SystemC Initiative, 2011

Preamble

A year after the release of OSCI's AMS 1.0 Standard in March 2010, the SystemC Analog/Mixed-signal (AMS) extensions continue to emerge in semiconductor and system integrator industries as the new system-level modeling language supporting mixed-signal ESL design and verification methodologies. The interest from these industries is to create mixed-signal virtual prototypes using SystemC, Transaction-level modeling (TLM) and the SystemC AMS extensions, to address the design challenges when designing, verifying and integrating complex heterogeneous systems. In such systems, the digital HW/SW design is an important element, often modeled in SystemC and TLM, but only seen as a subsystem within the entire heterogeneous system. The interface to the outside –analog– world enforces interaction with AMS subsystems, which nowadays do not work autonomously, but often require digital steering and control for the most optimal system performance. In addition, the modeling of the application context, such as the air interface of a wireless network, the car infrastructure in case of automotive, or analog sensors in an imaging application are essential to define the overall system functionality and behavior. This requires an integral mixed-signal system-level modeling and simulation strategy where all these system elements can be combined easily, allowing instances from various domains at different levels of abstraction. The use of the SystemC AMS extensions have become a valuable and essential part in the creation of these mixed-signal virtual prototypes, to facilitate uses cases such as software development, architecture exploration or system validation, but now including AMS subsystems.

At this SystemC AMS Day, a wide variety of industrial applications will demonstrate the adoption of SystemC AMS in these industries, and also give an outlook on the standardization plans to further extend and advance the SystemC AMS extensions into other application domains. Both system integrators as well as semiconductors industries will present their applications and show the relevance and benefits of applying the SystemC AMS extensions for their system design or verification tasks.

Not only in the industry, but also the academic world and research institutes embrace the SystemC AMS extensions. Renowned universities have recently established the "Academic Connection Program", which has the objective to exchange knowledge for the introduction of the SystemC AMS standard into educational and teaching programs. They have already adopted SystemC AMS into their curriculum and educational program and offer a supporting hand for other organizations who would like to do the same.

We hope you enjoy the SystemC AMS Day, where we invite system integrators, modeling experts, EDA vendors and system-level design and verification architects to share and exchange knowledge on the industrial application and benefits of using the emerging SystemC AMS 1.0 standard. And in case you would like to stay informed on SystemC AMS after this event, please visit www.systemc.org or sign-up to the AMS discussion forum at this website.

Martin Barnasconi
Chair SystemC AMS Day 2011

Content

Synthesis-supported
Design of Analog Circuits

BMBF label: 01 M 30 86

Northrop Grumman LITEF GmbH

Modelling and Simulation of a Fibre Optical Gyro System with SystemC AMS

12.5.2011

Dr. Stefan Rieke

"This work is supported by the German Ministry of Education and Research BMBF under grant number 01M3086

The author is responsible for the content of this publication."

© Northrop Grumman LITEF GmbH, Freiburg, Germany

Principle of Fibre optic gyroscopes

Sagnac effect:

- Measure optical path length difference of counterpropagating light beams along a circular path as a phase difference
- The time difference between the counterpropagating light beams is measured as phase difference (interference)

Phase shift: [redacted] 2 [redacted] D [redacted]

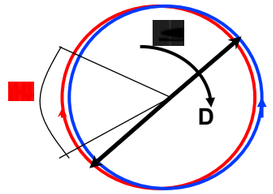
L: length of the optical path

D: diameter of the circle

[redacted]otation rate

[redacted]optical wave length

c: speed of light



© Northrop Grumman LITEF GmbH, Freiburg, Germany

Vetraulich/Propiertery & confidential - subject for NDA

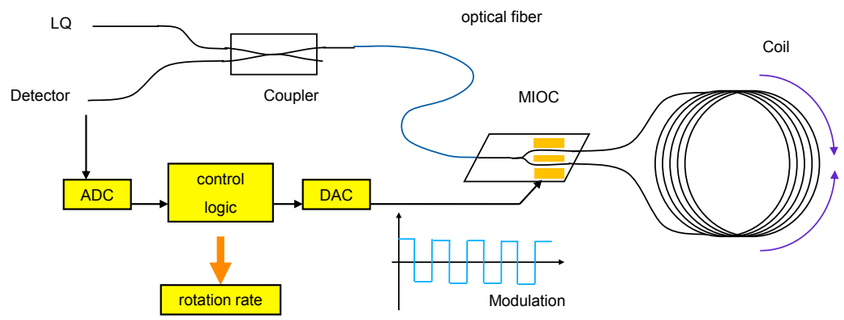
Motivation

- **Development, General Goals**
  - Through simulation, to improve the understanding of the fundamental physical behaviour of the gyro as a sensor
  - Use simulation to explore the interaction of the different sensor components with each other, especially analogue electronics
  - To evaluate design changes using simulation during the development phase with the main objective to reduce costs and development time
  - Use simulation to find potential sources of errors and improvements
- **Expected Economical Benefit**
  - Significant increase in quality and productivity
  - New opportunities for developing innovative complex systems
  - Saving costs for development and hardware

© Northrop Grumman LITEF GmbH, Freiburg, Germany

Vetraulich/Propiertery & confidential - subject for NDA

FOG signal processing principle



Rotation of the coil → phase difference → variation of intensity

- Closed-Loop control
- MIOC: multi functional integrated optics chip, needed for phase shifting
- Operating point stabilisation → MIOC-Signal modulation

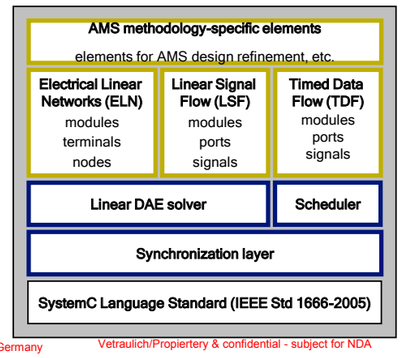
© Northrop Grumman LITEF GmbH, Freiburg, Germany

Vetraulich/Propiertery & confidential - subject for NDA

Why SystemC AMS

- Existing Simulation makes a lot of assumptions about the interfaces between
  - optical path and analogue components
  - analogue components and digital signal processing
- New Simulation need
  - C++ API
  - no assumption about the interaction between optical, analogue, mixed signal and digital part
  - different time domains available
  - change time intervals for each module during the simulation (event based)
  - easy implementation of different abstraction levels (grade of details)

SystemC AMS extensions 1.0



© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Modelling: level of abstraction

minimal system simulation time ←→ exact system description

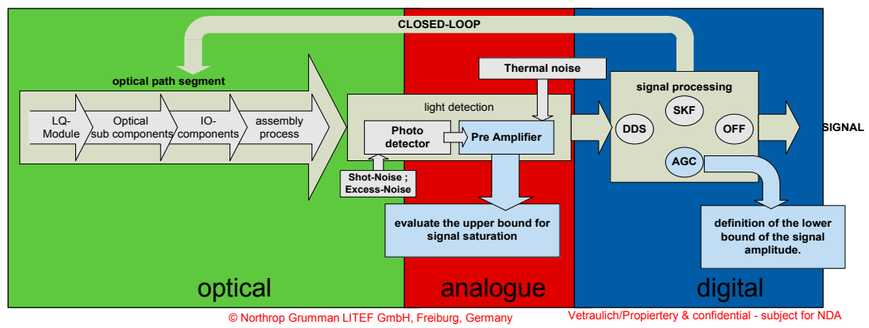
- Keys to define the abstraction level of a functional unit
  - functional unit: optical path, digital, pure analogue, mixed signal, detector, auxiliary controller
  - criticality for the FOG performance
  - sources of errors relevance
  - time of simulation
- level of abstraction: Question of intention (simulation task)
  → A functional block could be implemented in different levels of abstractions to optimize the simulation time
- For the virtual FOG, 3 different abstraction levels where defined

© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Simulation concept

Simulation shall help us to understand the interaction between:
optical path ←→ analogue components and
analogue components ←→ digital signal processing

Important aspects
- dynamic and timing resolution
- specific behaviour of electronic components and/or circuits i.e. Slew-Rate, finite signal slopes, overshooting, ...
- environmental conditions influence on electronic components (i.e. ADC vs. temperature)



© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Using three abstraction levels

- **High precision models, described with schematics**
  - Where block behaviour has a strong influence on the system
  - Where easy, accurate descriptions of the anal. functions are not possible
  - \+ The process includes non-negligible, hardware parasitics
  - − But simulation is complex and time-consuming
- **High precision models, described with complex algorithms**
  - Where block behaviour has a strong influence on the system
  - Where accurate math. descriptions can be used with only minor assumptions
  - \+ The process includes non-negligible, hardware parasitics
  - − But simulation is complex and time-consuming
- **Low complexity models**
  - Where a block has a low or minor influence on the system
  - Hardware parasitics are not included
  - Complex algorithms are avoided
  - \+ Simulation is very fast
  - Not applicable if studies about sources of errors and the implication to the system are done

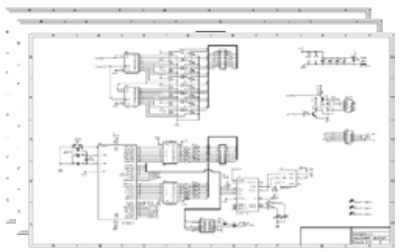
© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Example: optical receiver board

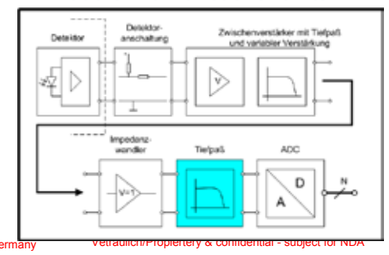
- Steps to define the functional units within a Module
  1. Extract functional units from schematics
  2. Figure out
     - implication to the performance
     - implication to the sources of errors and therefore to the stability of the system
     - Measureable parameter (optical or mechanical modules more complicated)
  3. Implementation

     Depending on Step 2 a unit could be realised in more than one abstraction level
  4. Test the module:
     - Software: different test benches
     - Hardware: Measurement and comparison

Schematics



module with functional units



© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Few Implementation

- Simulation time
  - highest grade of detail: 1000x real time
  - lowest grade of detail: 100x real time
- Functional units with different abstraction levels
  - MIOC
  - optical receiver module
- Input sources
  - rand generator
  - functional generator
  - data files

Modules	API	Time *t*
LQ module	SystemC-AMS	$5*10^5$
OR module	SystemC-AMS	1
MIOC+Coil	SystemC-AMS	$2,5*10^2$
digital data path	SystemC	$2,5*10^2$
Auxiliary controller (3)	SystemC	$2,6*10^5$
Control and reset logic	SystemC	./.

t: minimal time constant

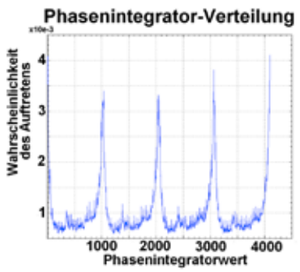
© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Capability of the simulation

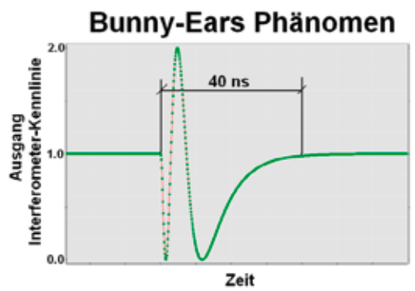
Examples of two effects, that must be handled correctly in the simulation

Distribution of the rotation rate

- **Rotation of the earth**
- **MIOC Phase-bleed effects uncompensated**



Bunny Ears: parasitic optical and electrical effects

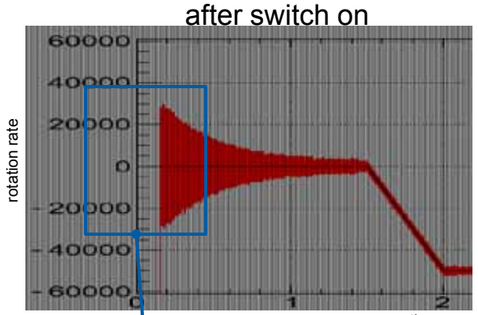


© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Simulation: cross check

- Switch on behaviour: a good cross check for
  - Does the closed loop system work correctly
    → transient oscillation ok
    → range of the parameter values ok.
- Response to an applied rotation rate
- Check the random walk (Noise)

Response of the rotation rate (red) after switch on

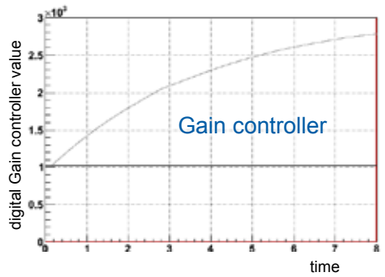


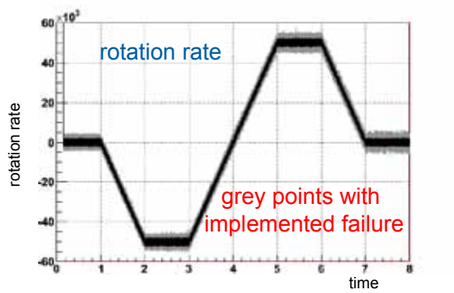
Transient oscillation with a flat rotation rate

© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Simulation: Sources of errors and their implication

- Implication of a failure in a OR module
- Important for
  - Acceptable tolerance of elec. and optical components
  - Effective root cause analysis
- Impact (i.e. R doesn't work)
  - Gain controller (not stable)
  - Response on the rotation rate → RW increase with time (grey points)

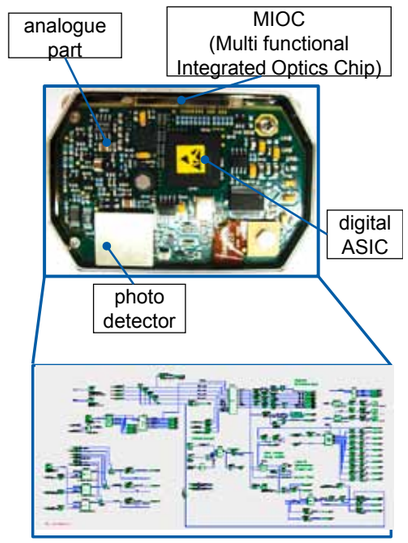




© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Verification & Validation

- To have a good prediction for the next FOG system:
  → The results of the simulation are exactly the same as in the HW for all parameters (ideal)
  Compare simulation results with real FOG
- But some parameters are not accessible because of
  - Use of ASICs and FPGAs
  - optimal choice of CPU power in the FOG
- Special HW (digital Test Board) needed to have access to all important
  - signals
  - parameters (Software and FPGA)

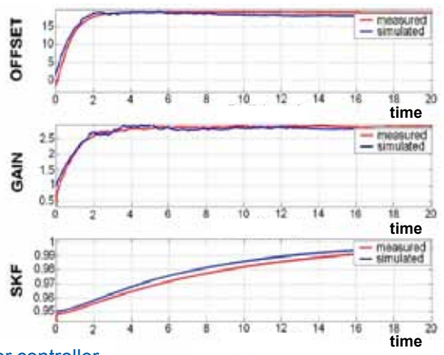


© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

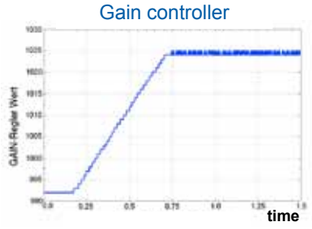
Virtual FOG demonstrator: transient oscillation

- Auxiliary controllers are a part of the digital data path
- Reuse existing C++ code for the virtual FOG.
- Comparison C++ Simulation and hardware (right side)
- Because of the more complex SystemC AMS simulation the „old" parameters must be adjusted

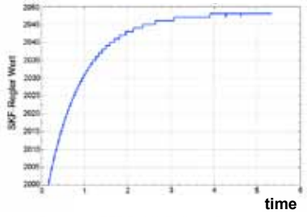
C++ Simulation vs. real world



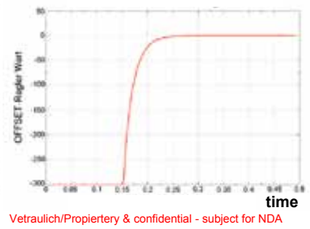
Gain controller



Scale-Factor controller



Offset controller



© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

Summary

- Virtual FOG
  - Evaluate design changes during the development phase with the main objective to reduce costs and development time
  - Improve the understanding of physical basics of the FOG
  - Find potential sources of errors and improvements
  - Significant increase in quality and productivity
- Preferred Simulation environment: SystemC AMS
  - Integrate different domains in one simulation (optic, analogue, digital, mixed-signal)
  - Easy implementation of different abstraction levels for one module
  - Change time resolution for each module during the simulation (event based)
  - Use different (asynchronous) time domains
- Verification and Validation
  - To evaluate design change a good description of an existing HW system must exist.
  - A special test board is provided to have access to all important internal parameters and values.

© Northrop Grumman LITEF GmbH, Freiburg, Germany
Vetraulich/Propiertery & confidential - subject for NDA

SystemC AMS based Virtual Platform for Automotive Electronic Systems Development & Verification

Ingmar Neumann

Continental Corporation: Divisions and Business Units

Continental Corporation

Automotive Group

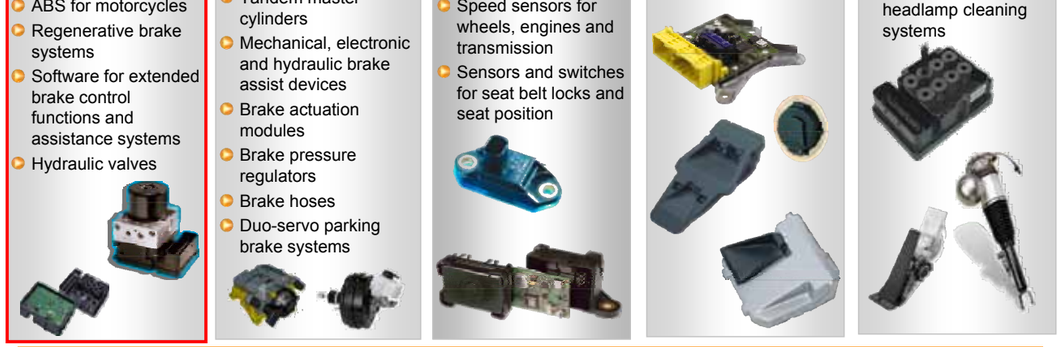
Chassis & Safety	Powertrain	Interior
Electronic Brake Systems	Engine Systems	Instrumentation & Driver HMI
Hydraulic Brake Systems	Transmissions	Infotainment & Connectivity
Sensorics	Hybrid & Electric Vehicle	Body & Security
Passive Safety & ADAS	Sensors & Actuators	Commercial Vehicles & Aftermarket
Chassis Components	Fuel Supply	

Rubber Group

Passenger and Light Truck Tires	Commercial Vehicle Tires	ContiTech
Original Equipment	Truck Tires Europe	Air Spring Systems
Replacement Business EMEA	Truck Tires The Americas	Benecke-Kaliko Group
Replacement Business The Americas	Truck Tires Asia Pacific	Conveyor Belt Group
Replacement Business Asia Pacific	Industrial Tires	Elastomer Coatings
Two-Wheel Tires		Fluid Technology
		Power Trans-mission Group
		Vibration Control
		Other Operations

Chassis & Safety Division – Business Units

Electronic Brake Systems	Hydraulic Brake Systems	Sensorics	Passive Safety & ADAS	Chassis Components
Electronic brake systems, e.g. ABS and ESC	Brake disks	Inertial sensors for stability and ESC applications	Passive safety	Steering systems
Electric-hydraulic combi brake (EHC)	Drum brakes	Sensors for active chassis control	Driver assistance systems	Air suspension systems
Control units for motorcycle brakes	Brake calipers	Steering angle and torque sensors	Occupant classification	Chassis electronics
ABS for motorcycles	Parking brakes	Speed sensors for wheels, engines and transmission	Product integration and validation	Electronic components
Regenerative brake systems	Electric parking brakes	Sensors and switches for seat belt locks and seat position		Windshield and headlamp cleaning systems
Software for extended brake control functions and assistance systems	Brake boosters			
Hydraulic valves	Tandem master cylinders			
	Mechanical, electronic and hydraulic brake assist devices			
	Brake actuation modules			
	Brake pressure regulators			
	Brake hoses			
	Duo-servo parking brake systems			

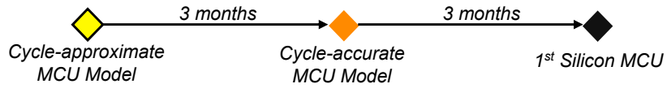


Target System: EBS ECU with Focus on Integrated Circuits

- Target System:
  - Electro-hydraulic Control Unit for Electronic Stability Control (E
  - Car Sensors required to execute software on virtual platform
- Components:
  - Safety Integrity Level 3 (SIL3) certified Chip-Set:
    - Full-Custom leading edge automotive safety MCU
    - Full-Custom leading edge mixed-signal IC
  - Actuators: Valves, Motor
  - Sensors: Acceleration-, Pressure-, Wheel-speed-Sensors

Motivation for C-based IC Modeling: Benefits

- **Reduction of developing time** / time-to-market
  - Evaluation of hardware components to be developed in complex hardware/software environment
  - Parallelizing development of hardware and software



- **Cost Reduction**
  - Simple PC setup instead of physical test equipment (setups of various devices can be easily distributed, installed and maintained)
  - Reduction of Silicon Redesign (e.g. ROM masks)
- **Quality improvement** by verification enhancement
  - Verification of software components
  - Verification of complex analog/mixed signal systems
  - Verification of complex hardware/software systems
- **Enhanced System Visibility and Fault Coverage**
  - Error Injection: Setup of fault conditions which are difficult to provoke in silicon device
  - Higher System Visibility: ECU-level actuators being modeled; Input of sensor data streams supported

From Proprietary to Standard Simulation Platform

- Continental Frankfurt develops Mixed Signal ASIC's and MCU's for fail safe applications since m/o 1990s
- Using Virtual Platforms for Development has a more than 12 years old tradition at Continental

AVSL:

•C++ class library for hardware modeling and simulator coupling

•In-house technology developed at Continental Frankfurt

3 years of development within ***AutoSUN*** research project

Why SystemC AMS

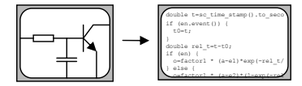
- There exist several analog/mixed-signal (AMS) extensions of hardware description languages:
  - Verilog-AMS
  - VHDL-AMS
  - SystemC-AMS

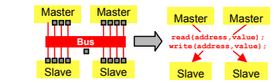
- We have chosen SystemC AMS because:
  - SystemC AMS allows *higher abstraction levels* than Verilog/VHDL-AMS
    ⇒ **increased simulation speed**
  - SystemC AMS is an *open standard*: *open source class library & simulator kernel* available
    ⇒ **simulation technology development, license fees**
  - Simple coupling with other tools offering a C-Interface

Continental's Technical Goals of AutoSUN

Increase of simulation Speed

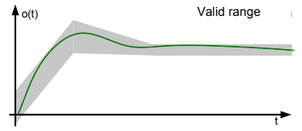
- Automatic generation of behavioral SystemC AMS models from net lists analogue circuit
- TLM modeling for analogue systems





Verification of Analogue Systems

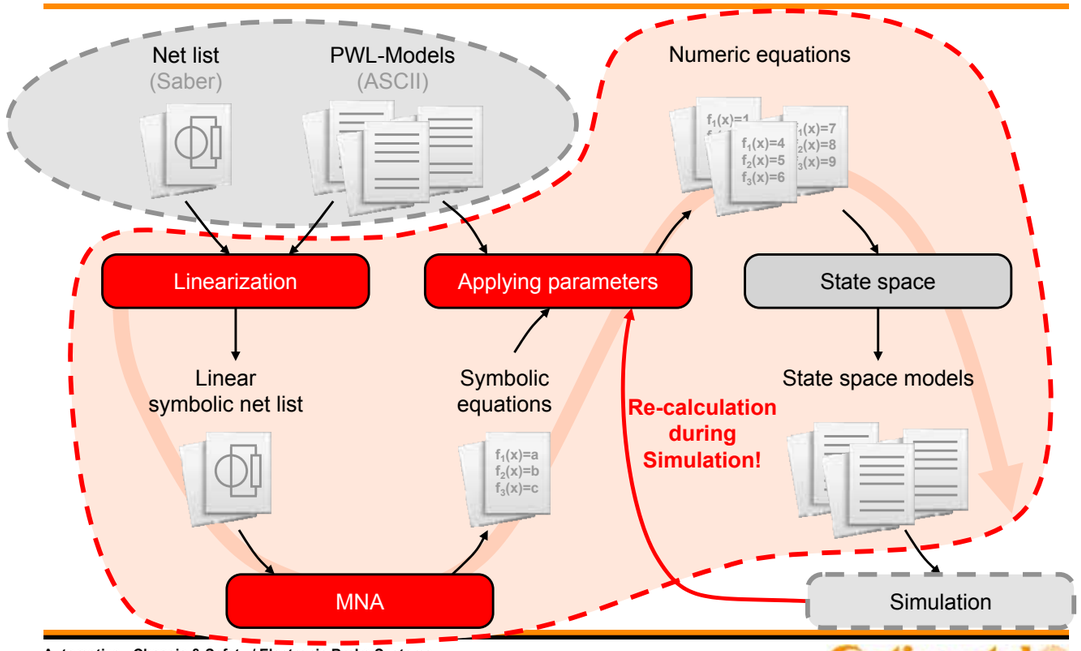
- Analogue property checking
- Regression testing



Simulator coupling

- SystemC co-simulation interfaces to existing commercial simulators in design environment

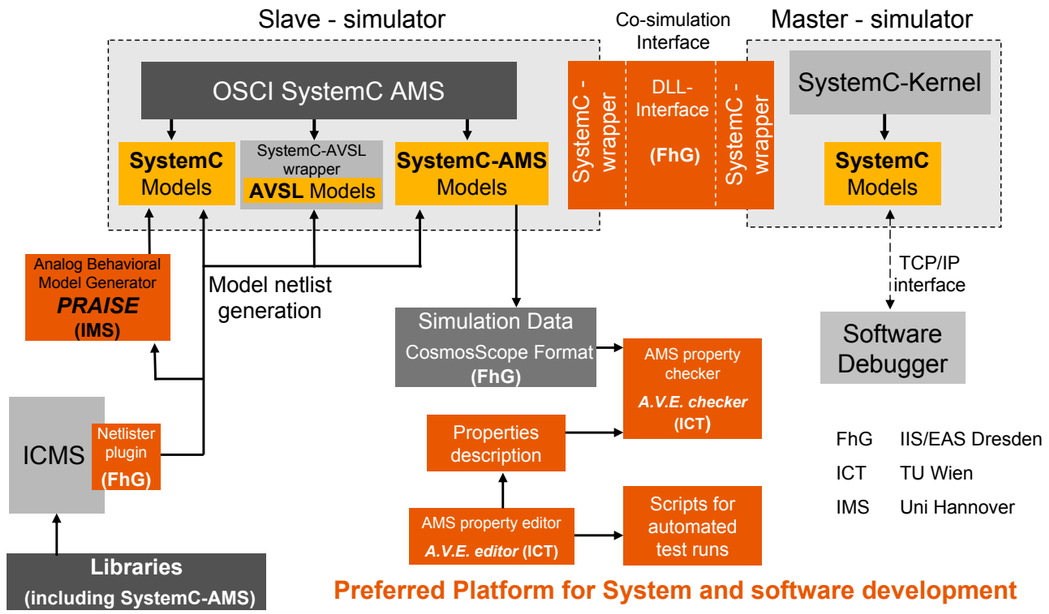
Creation of Behavioral Models of Analogue Coponents



© Continental AG

confidential

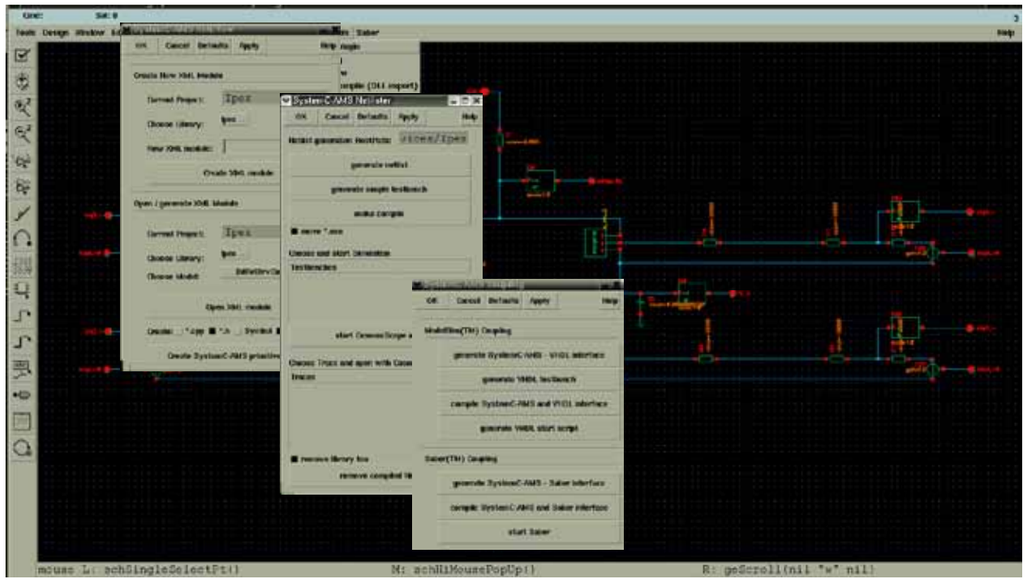
Pure SystemC Simulation Environment



© Continental AG

confidential

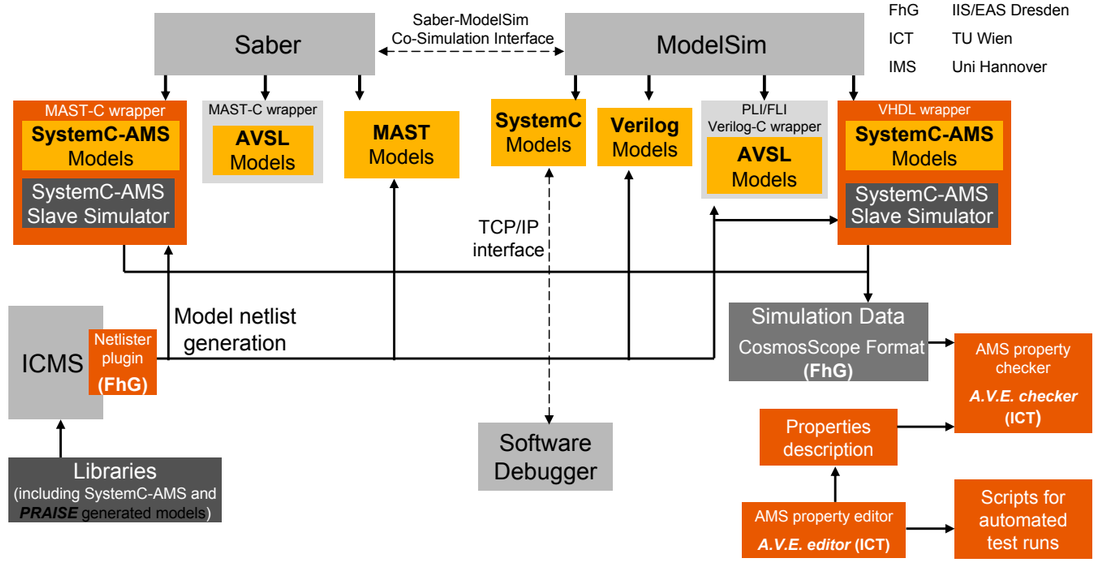
Schematic Entry with XML Based Design Flow



© Continental AG

confidential

Heterogeneous Simulation Platform



Preferred platform for hardware development (IC)

© Continental AG

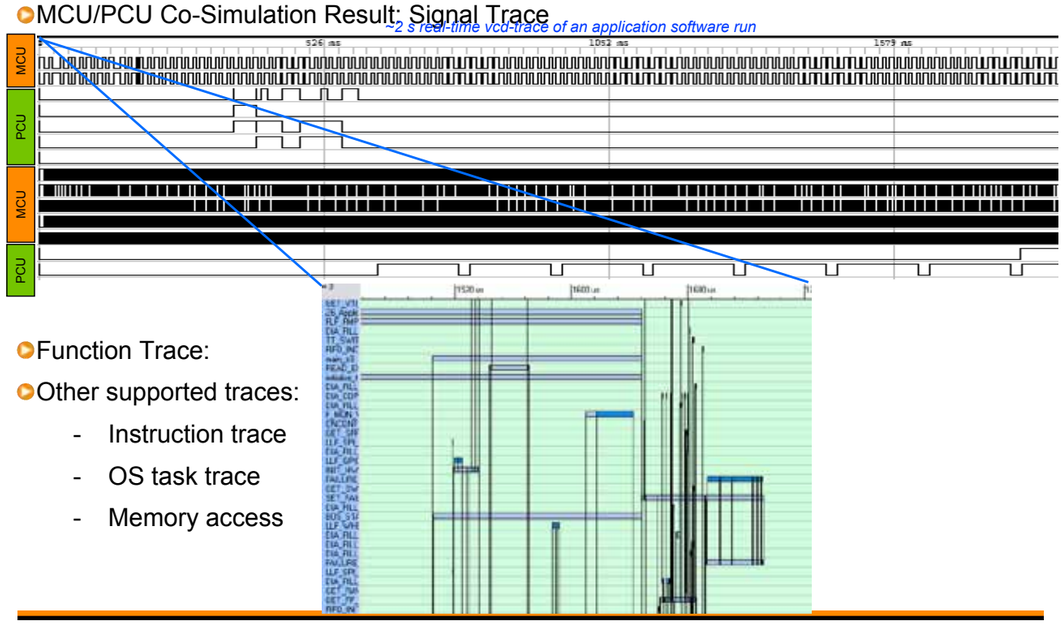
confidential

IC Simulation Use Cases

- Software-related use cases
  - (Pre-silicon) Software Development of hardware-dependant software parts
    ⇒ low-level drivers, AUTOSAR drivers, hardware-dependant S/W-functions
  - (Post-silicon) Software Verification
    - MCU connections on ECU Level (MCU I/O Test)
    - Failsafe concepts Microcontroller / Mixed-signal ASIC
    - Hardware/Software Interface (mutual failsafe requirements, Flash/ROM compatibility)
  - Cross-correlation to silicon; silicon performance evaluation
- Hardware-related use cases
  - ASIC circuit development/design exploration
  - Functional Partitioning Microcontroller/Mixed-Signal IC
  - Functional Partitioning Hardware/Software
  - Top-level Mixed-Signal IC Design Verification
  - Certification of automotive safety chip-set (enabling verification diversity)

S/W Analysis on Virtual Platform

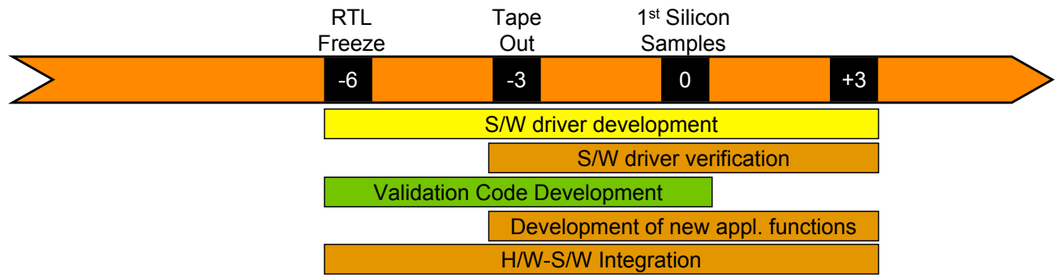
- MCU/PCU Co-Simulation Result: Signal Trace



- Function Trace:
- Other supported traces:
  - Instruction trace
  - OS task trace
  - Memory access

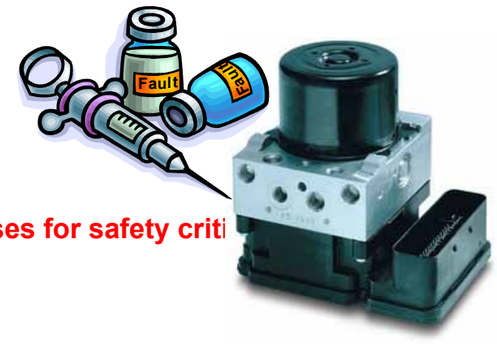
Pre-Silicon Software Development

- S/W driver development at supplier (6 months ahead of 1st silicon)
- S/W driver verification at Conti (3 months ahead of 1st silicon)
- Silicon validation code development (6 months ahead of 1st silicon)
  - 100% complete verification database when silicon samples are available
- S/W development of new application functions (3 months ahead of 1st silicon)
- H/W-S/W integration (3 months ahead of 1st silicon)



Validation of Failsafe Concepts

Safety critical applications:
•Malfunction may lead to accidents
⇒ Mature safety concepts required:
- Malfunction detection
- Fall-Back policy



- **Laws on product liability**
- **ISO 26262: Development processes for safety criti systems**

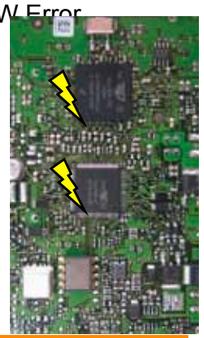
Exemplary techniques:
•Redundant CPU cores
•ECC protected memories
•Monitoring voltages/currents in analogue parts: detection of disconnects and shorts.
•Watchdog signals/data packages

Validation requires testing environments with faulty components:

Error injection

Error Injection

- **Target**:
  - Functional safety software verification: exception handling in case of system fail state
- **Benefit** of simulator approach:
  - Access to failure modes, which are hard to provoke with silicon setup
- **Method**:
  - Injection of errors into IC models by simulation backplane during program execution
- **Examples:**
  - MCU errors:
    - Memory System: Flash data/ECC error, ECC correction error, RWW Error
    - Bus Systems: Data/Address/Control Bus error
    - Clock System: Loss-of lock reset, Loss-of clock reset
    - Failsafe System: Watchdog timer reset, Checkstop reset
    - Reset/Interrupt System
  - Mixed-signal IC errors:
    - Watchdog Failures: Initialization/Command/Redundancy Errors
    - ADC Failures: Data Corruption Errors, Shutdown Errors
    - Actuator Failures: Leakage Errors, Load Driver Failures



 © Continental AG

Summary: Continental's Chassis and Safety Virtual Platform

- Continental EBS IC Development is committed to virtual prototype development
- ECU-level mixed-signal, mixed-level Virtual Platform
- ECU-level Virtual Platform enables pre- and post-silicon use cases for software development, verification and analysis as well as hardware design exploration
- Coupling technology for IP model integration being developed by Continental
- Scripted Tool Flow for model generation mandatory to manage complex system modeling

 © Continental AG

SystemC AMS Day, May 12th, 2011

Automatic Transformation of MATLAB/Simulink Models to SystemC AMS

Nico Bannow Robert Bosch GmbH
Ralph Görgen OFFIS Institute
Wolfgang Nebel University of Oldenburg

Federal Ministry of Education and Research

This work has been developed in the project RapidMPSoC. RapidMPSoC (project label: 01 M 3085) is partly funded by the German ministry of education and research (BMBF) within the Research Program ICT 2020 .

OFFIS Transportation

125 Bosch 1886–2011

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

BOSCH

Overview

Challenge

- Simulation performance for Analogue and Mixed Signal Systems
  - Rising integration and relevance of analogue components

Reason

- Multiple domain-specific modeling environments (Simulink, HDL's, ...)
  - Co-Simulation and coupling of different simulators

Solution

- Automatic transformation of MATLAB/Simulink models to equivalent SystemC AMS modules
  - Easy integration into digital SystemC simulation
  - Reuse of Simulink models in later design steps

OFFIS Transportation

125 Bosch 1886–2011

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

BOSCH

Agenda

- Motivation and Use Case
- Real Time Workshop
- SystemC AMS Generation
- Test Bench Generation
- Results – Evaluation of Industrial Model
- Summary

OFFIS Transportation

125 Bosch 1886–2011

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

BOSCH

Motivation

OFFIS Transportation

125 Bosch 1886–2011

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

BOSCH

Industrial Use Case

Environmental model

Car model & Driver model

Engine model

Simulink Model

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

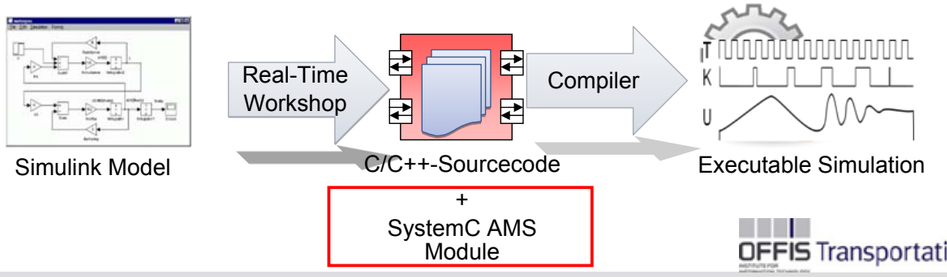
OFFIS Transportation

BOSCH

Real Time Workshop

- Toolbox for Simulink
- Generates equivalent C/C++ source code from Simulink models
  - Customisable for specific targets via TLC
- Good performance of result code
- Supports most Simulink blocks and block sets

- ***Idea:*** Modify Real Time Workshop code generation for SystemC AMS

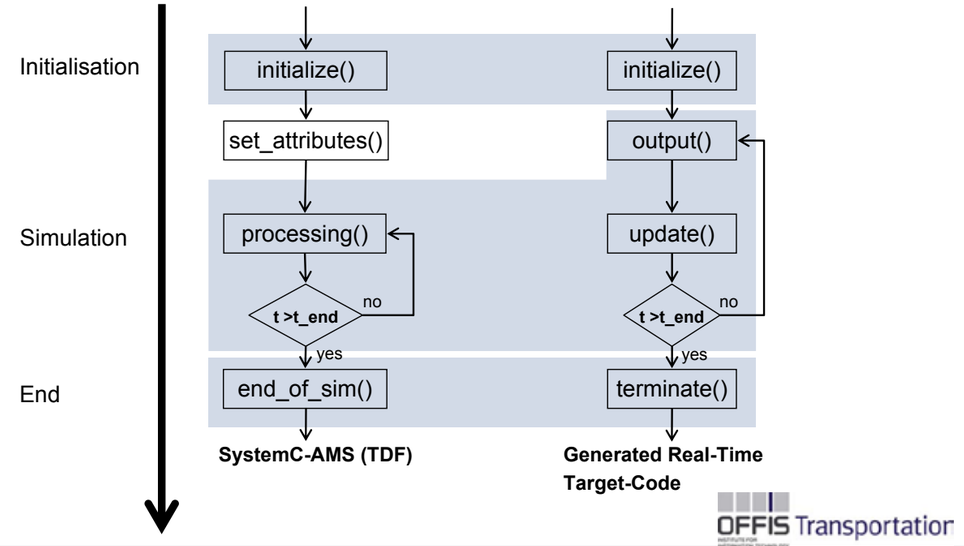


CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

OFFIS Transportation

BOSCH

SystemC AMS vs. Real Time Workshop Code



CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

OFFIS Transportation

BOSCH

Generation of SystemC AMS Module

- Real Time Workshop ‘Compiled Model’ generation
  - Contains code fragments and additional information (ports, structure, ...)
  - Stored in MATLAB workspace

- SystemC AMS generation
  - Implementation of a new target with TLC (Target Language Compiler)
  - Global variables and functions become member attributes and methods
  - Initialization in module constructor
  - Assignment of sample rate to module ports
  - processing() handles I/O and calls output() and update()
  - Multiple instances of same module in one simulation possible

- Java Tooling
  - Accesses ‘Compiled Model’ via MATLAB Engine and Java Native Interface
  - Generation of code files via Java Emitter Templates

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

OFFIS Transportation

BOSCH

Handling Fixed Point Types

- Fixed point types become integer types in generated code
  - Possible misinterpretation
    e.g., 2.0 (0b000010.00) → 8 (0b00001000)

- Solution:
  - Using SystemC fixed point types
  - Extract type information from RTW
  - Extending code generation for specific type conversion
  - Reading and writing values via conversion elements

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Configuration of AMS Model Parameters

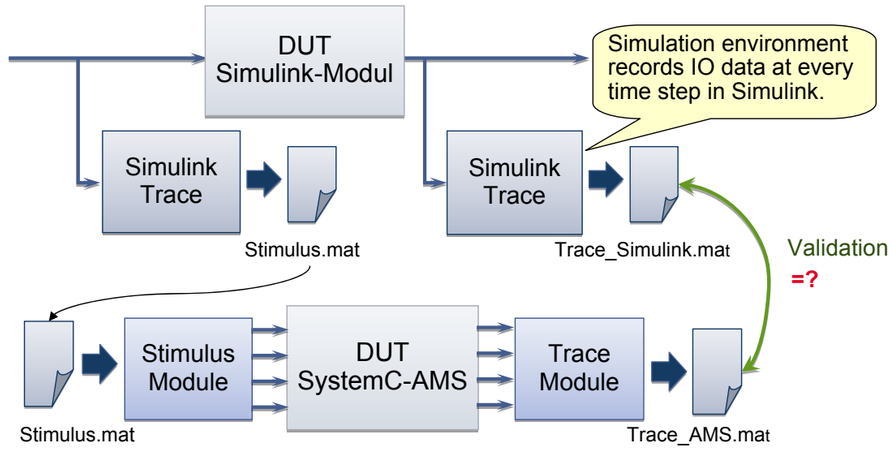
- Modification of parameters in generated SystemC AMS model
  - Initialization of constant values at start of simulation
    e.g. A/D converter quantization
  - Dynamic modification of variable values during simulation
    e.g. gain factor of gain block

- Automated execution of test series with changing parameter values

- Simple and type-safe user interface:
  - `CParameterHandler p = model.getParameter(„my_module/g1/gain“)`
  - `Read:  p.get(my_var)`
  - `Write:  p.set(my_var)`

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Automatic Test-Bench Generation



CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Limitations of this Concept

- Limitations of Real Time Workshop
  - No support for solvers with variable step size
  - Unsupported Simulink blocks
    - M-File S-Functions
    - MATLAB functions
    - Model Reference Subsystems
    - ...

- Access to some parameters not supported
  - e.g., initial value of a signal

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Evaluation Results with Automotive Examples

- Mathworks Example: Clutch
  - Modified Simulink model from Mathworks
  - Automotive clutch for torque submission
- Bosch industrial model:
  - From Automotive Business Unit
  - Vehicle component in a complete environment model
  - Becomes about 100.000 lines of generated C++ code
- Comparison of simulation results with generated testbench
  - Diverse synthetic models
    - Difference always < 1‰, most without differences
  - Bosch industrial model
    - No differences

CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

OFFIS Transportation

BOSCH

Evaluation: Simulation Performance

Model	Simulated Time	Simulink	Sim-A	Sim-RA	SCA-1	SCA-2
Clutch	1000 s	57,6 s	30,3 s	37,4 s	23,1 s (x 2,5)	44,8 s
Bosch Model	10 s	171,3 s	171,8 s	--	0,188 s (x 900)	0,362 s

- Simulink: Simulink simulation
- Sim-A: Simulink simulation with Accelerator option
- Sim-RA: Simulink simulation with Rapid Accelerator option
- SCA-1: SystemC AMS Simulation, TB and DUT in one single TDF cluster
- SCA-2: SystemC AMS Simulation, TB and DUT in separate TDF clusters connected via **sc_signal** and sca_sc ports

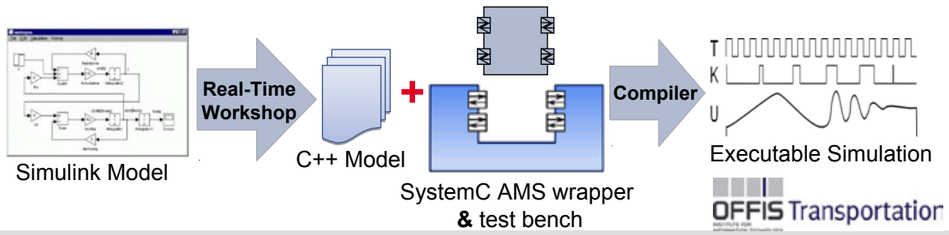
CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

OFFIS Transportation

BOSCH

Summary

- Fully automated generation of equivalent SystemC AMS modules from MATLAB/Simulink models
  - Less co-simulation overhead (synchronization, data conversion, ...)
  - Multiple parallel simulations without additional license costs
- Test bench generation and model instrumentation for validation
- Easy access to model parameters during simulation
  - No model generation and re-compilation after parameter changes
  - Suitable for regression test
- Successful evaluation with complex real-world examples



CR/AEH2 | May 12th, 2011 | © Robert Bosch GmbH 2011. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

OFFIS Transportation

BOSCH

An Efficient Transceiver Design Verification Method by means of SystemC AMS – VHDL Co-simulation

Gerhard Deutsch, Jakob Jongsma, Thomas Herndl

Infineon Technologies Austria AG

mailto:{gerhard.deutsch,jakob.jongsma,thomas.herndl}@infineon.com

SystemC AMS Day 2011

Outline

- Who We Are
- Motivation
- SystemC AMS Modeling
- Outlook
- Summary

Who We Are I

- Involved departments (Graz, Austria):
  - Contactless & RF Exploration (CRE)
  - Automotive Sense & Control (SC)
- CRE: forefront research projects and feasibility studies in the fields of „passive contactless / RF Identification (RFID)" and "active short range RF"; prototypes / demonstrators
- SC: R&D for RF products up to 1GHz
  - automotive key-applications are:

Who We Are II

- Remote Keyless Entry (RKE):

source: letsgomobile.org

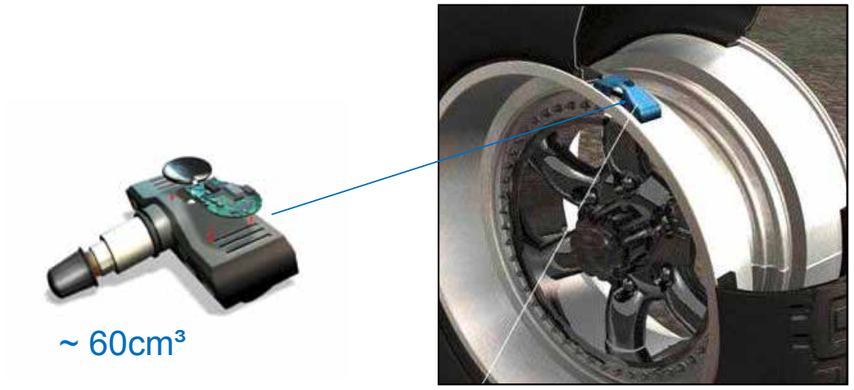
„Smart Key Fob" (Vision)

source: wikipedia.org

Conventional Key Fob

Who We Are III

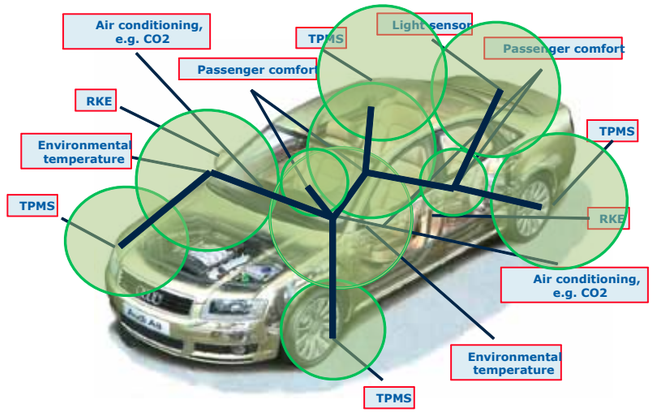
- Tire Pressure Monitoring Systems (TPMS):



Rim Mounted System

Who We Are IV

- industrial and consumer segment: garage door openers, remote controls for home electronics, automated meter reading, home automation, security systems,…
- Future applications: Wireless Sensor Networks (WSN)



E.g. automotive:

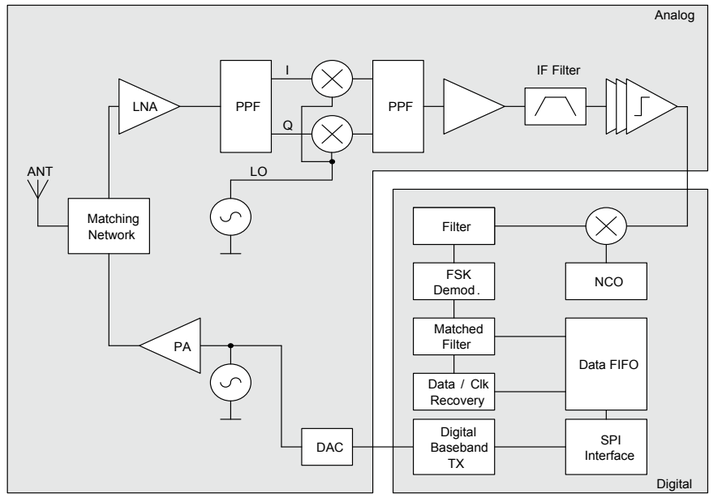
⇨ in-car WSN: non-mission critical sensors connected in a mesh network

Motivation I

- Today's RF devices, even low-end devices, comprise huge system complexity (especially digital part) ⇨ concept, design and verification challenge for product development:
  - a lot of computational load ⇨ time consuming
  - find ways to counter this
- One step in this direction is the reduction of digital design and verification cycles by speeding-up simulation runs:
  - this is achieved by co-simulation of SystemC AMS and VHDL ⇨ SystemC AMS modeling on a high abstraction level
- This work has been partly funded by the EC program "Cooperative Hybrid Objects Sensor Networks" (CHOSeN) and Austrian FIT-IT project "Sensor Network Optimizations by Power Simulation" (SNOPS).

Motivation II

- Transceiver (TRx) ASIC Block Diagram (data path):

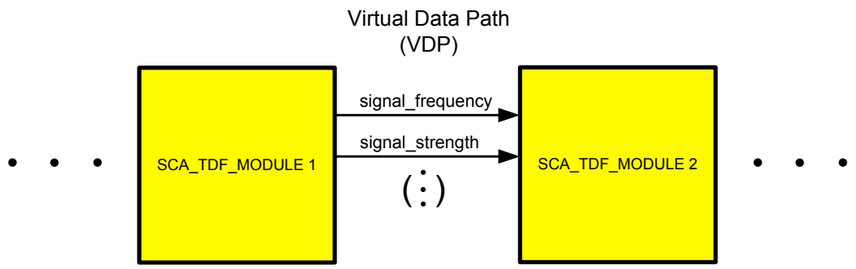


SystemC AMS Modeling I

- Applied IDE is provided by Fraunhofer (FhG IIS/EAS):
  - eclipse based
  - provides a lot of modeling features and automatism
- Main target is control path / Main Control Unit (MCU) operation verification of transceiver (Special Function Register (SFR) settings, enable signals, timing,…):
  - analog front-end data path abstraction ⇨ „Virtual Data Path“ (VDP):
    - ¬ only signal parameters passed between modules, not the signal itself
    - ¬ main signal parameters: signal frequency and signal strength
    - ¬ maybe additional information like signal offset, signal phase,…

SystemC AMS Modeling II

- For this Timed Data Fow (TDF) modeling is applied:

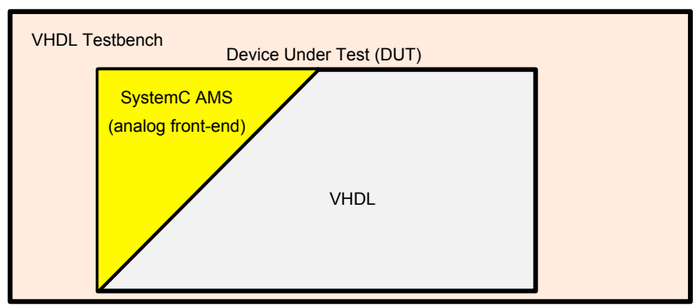


SystemC AMS Modeling III

- The analog front-end modules are kept as simple as possible, keeping their core functionality to a reasonable degree:
  - Mixer: simple addition (transmit-side) / subtraction (receive-side) of local oscillator frequency parameter value
  - Filter: ideal filter characteristic (rectangular magnitude response)
  - …
- At the interfaces to the digital domain real signals are generated from the respective signal parameters (including conversion to SystemC).

SystemC AMS Modeling IV

- Configuration option of TRx modules using “config.txt“ file:
  - includes e.g. filter bandwidth, gain settings, delays,…
  - single-source idea (could become kind of exchange format)
- SystemC AMS - VHDL co-simulation is performed using ModelSim, supporting mixed-language simulation ⇨ SystemC AMS integrated as shared object.

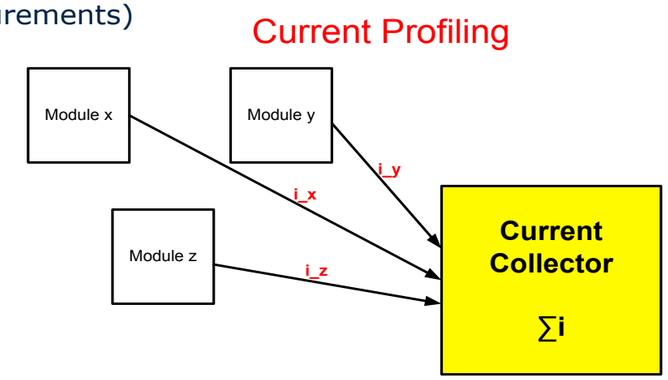


SystemC AMS Modeling V

- Simulation tune-ups:
  - by a proper choice of the simulation cluster sampling time (also test case specific) simulation speed can be (positively) influenced:
    - find a trade-off between simulation performance and timing accuracy
  - simulation performance also influenced by the number of module calls:
    - try to reduce it by integrating as much functionality as possible into the modules

SystemC AMS Modeling VI

- Current profiling as further modeling / verification feature (related to SNOPS project):
  - current collector: TDF module
  - multiple current states per module (values by simulation, measurements)



Outlook

- Transceiver performance / sensitivity simulations:
  - equivalent lowpass representation of the analog front-end data path (TDF modeling)
- Combination of virtual data path and equivalent lowpass modeling towards multi-level simulation is targeted:
  - abstraction vs. performance details (different views / focuses in one simulation)
  - adjustable simulation demands for specific test cases (simulation time)

Summary

- Gave overview on involved IFAT departments in Graz:
  - CRE, SC: focus of departments
- Explained motivation for SystemC AMS – VHDL co-simulation:
  - a lot of computational load in concept, design and verification process ⇨ speed-up simulation runs
- Presentation of SystemC AMS modeling activities up to now:
  - control path verification
  - VDP and TDF modeling
  - SystemC AMS - VHDL co-simulation issues
  - simulation tune-ups
- Outlook on planned SystemC AMS modeling issues:
  - equivalent lowpass modeling
  - multi-level simulation

SystemC AMS model of a CMOS video sensor

Fabio Cenni, STMicroelectronics Grenoble

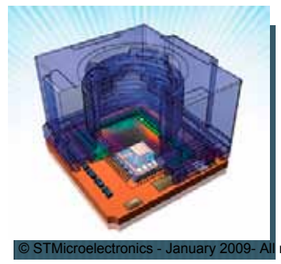
OSCI SystemC AMS Day, Dresden May 12, 2011

Outline

- CMOS image sensor and platform overview
- The challenge
  - Today HW prototype are needed for SW debugging … is virtual prototyping a good solution?
- CMOS image sensor model
  - VHDL-AMS limits
  - SystemC AMS TDF model
- Overview of the TLM platform
- Simulation results
- Demonstration
- Conclusions

CMOS image sensor

© STMicroelectronics - January 2007 - All rights reserved



© STMicroelectronics - January 2009- All rights reserved

Features:

- 5.0 mega pixel resolution sensor (2608 x 1960)
- 400 KHz “camera control interface” (CCI) command interface (I2C)
- Supported data formats: 10-, 8-, 7- and 6-bit RAW
- Fully integrated auto-focus Voice Coil Motor (VCM) Driver
- Analog binning modes: 2x2: 30 fps, 4x4: 60 fps

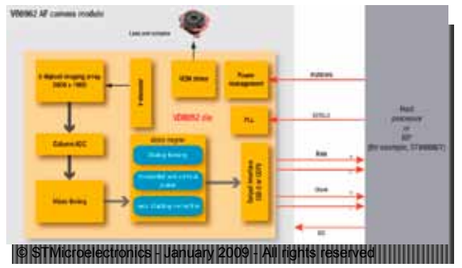
Technical specifications:

- Sensor technology: IMG175 ST’s 90 nm based CMOS imaging process
- Pixel size: 1.75 μm x 1.75 μm
- Exposure control: +81 dB
- Analogue gain: + 24 dB (max)
- Digital gain + 6 dB (max)
- Dynamic range 60 dB
- Min. illumination < 10 lux

source www.st.com

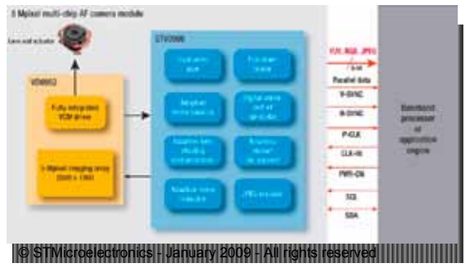
Platform overview

Focus on analogue acquisition part: accurate model needed



© STMicroelectronics - January 2009 - All rights reserved

Focus on post-processing: fast model needed



© STMicroelectronics - January 2009 - All rights reserved

- Camera module + Image Signal Processor (ISP) + CPU and peripherals

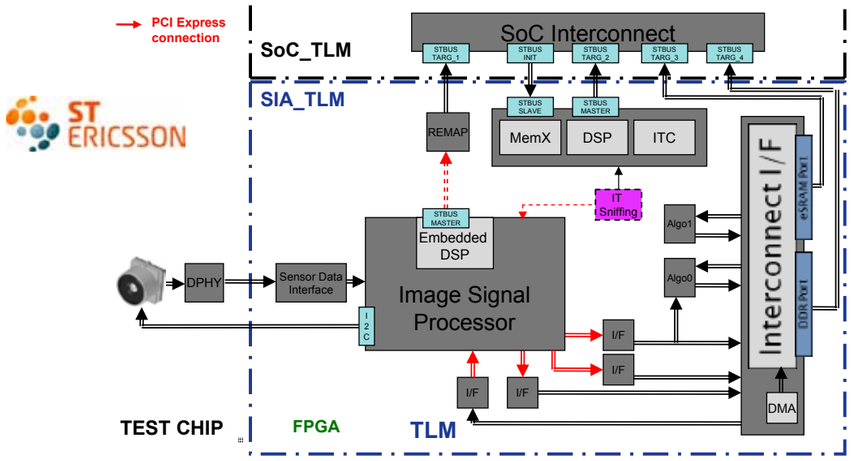
source www.st.com

The application context:

- STEricsson's Smart Image Architecture (SIA) platform uses a STMicroelectronics camera module
- Today it is necessary to wait for HW prototype boards before embedded SW debugging:
  - waiting for months for the boards to be available
  - unexpected test-bench setup delays
- Risk: miss time-to-market frame time

The challenge:

- Is it realistic to build a reliable virtual prototype of the image acquisition platform ? …



Specifications of the model

AMS model constraints:

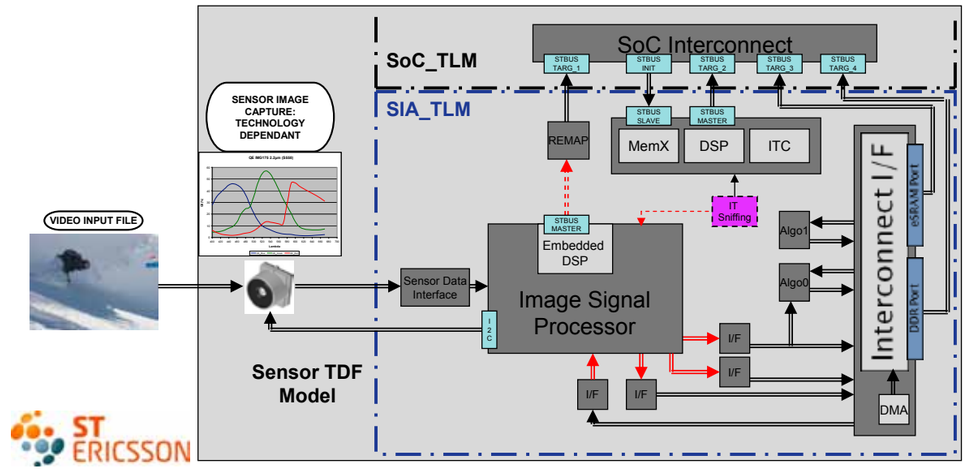
- 2 mega pixels sensor
- Coarse modeling: 1 frame per seconds of run-time
- Fine modeling: 0.1 frame per seconds of run-time
- SystemC TLM interfacing for control feedback
- Modeling of non-idealities
  - (lens shading, resin adsorption, defective pixels, etc…)

Expected benefits:

- embedded software development before the silicon is available
- validation of ISP algorithms for different sensors
- architecture space exploration

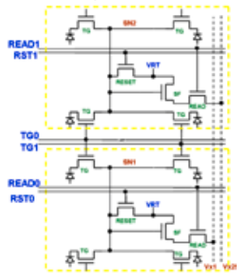
The targeted SystemC AMS/TLM platform:

- ISP wrapped or natively coded in SystemC TLM
- SystemC AMS Timed Data Flow (TDF) model of the sensor

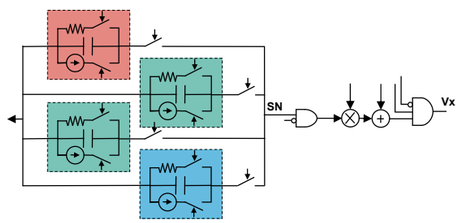


VHDL-AMS model

1T75 pixel schematic

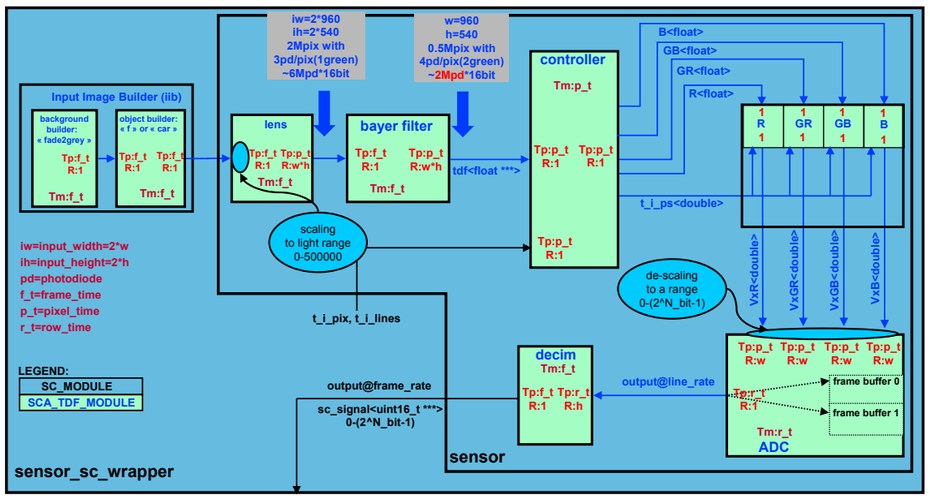


VHDL-AMS model structure

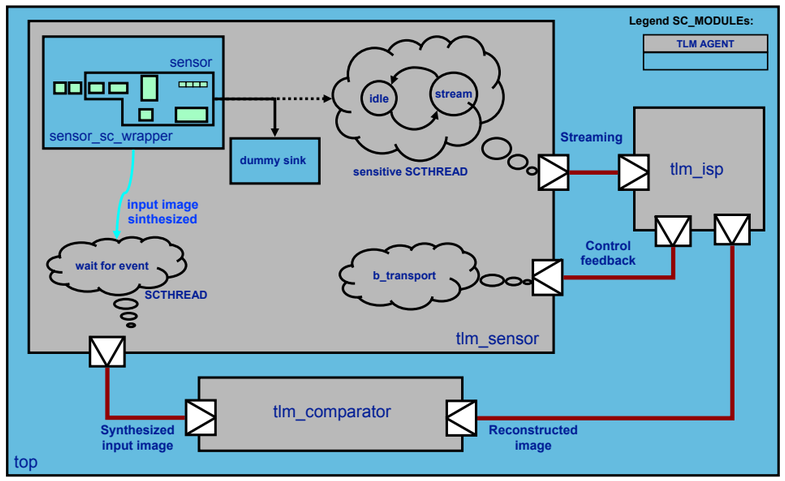


- Pros:
  - SPICE-like waveform results
  - Mainly used for debugging video timing control signals
  - Allow to model transistor-level effects
- Cons:
  - High simulation time demanding model (2h30mn for a 2x496 array)
  - Tough to enrich the model with other computation greedy aspects
  - It is not realistic to integrate/interact with a virtual model of the overall platform

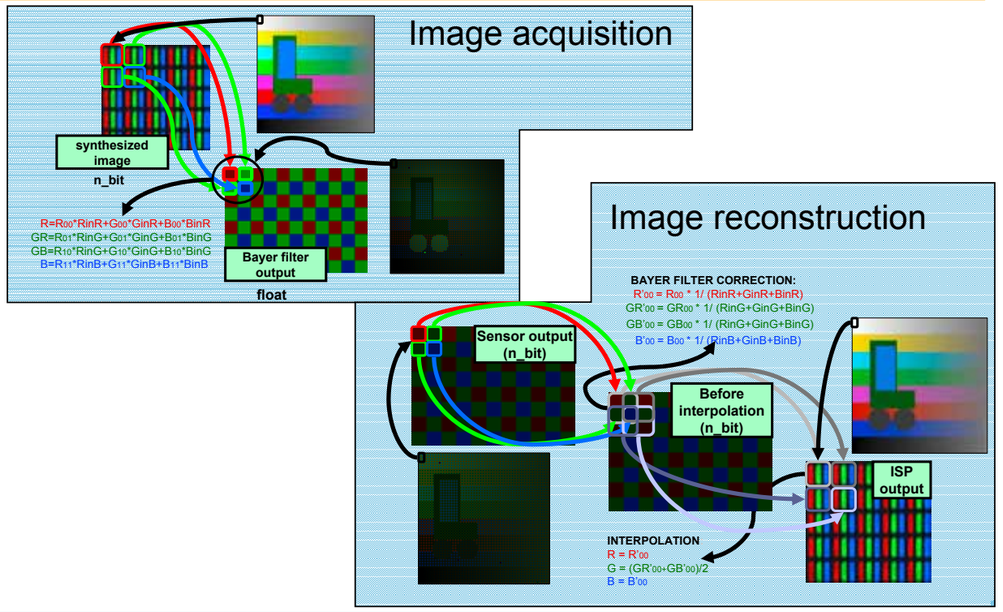
SystemC AMS TDF model architecture



SystemC AMS TDF/SystemC TLM 2.0 platform demonstrator



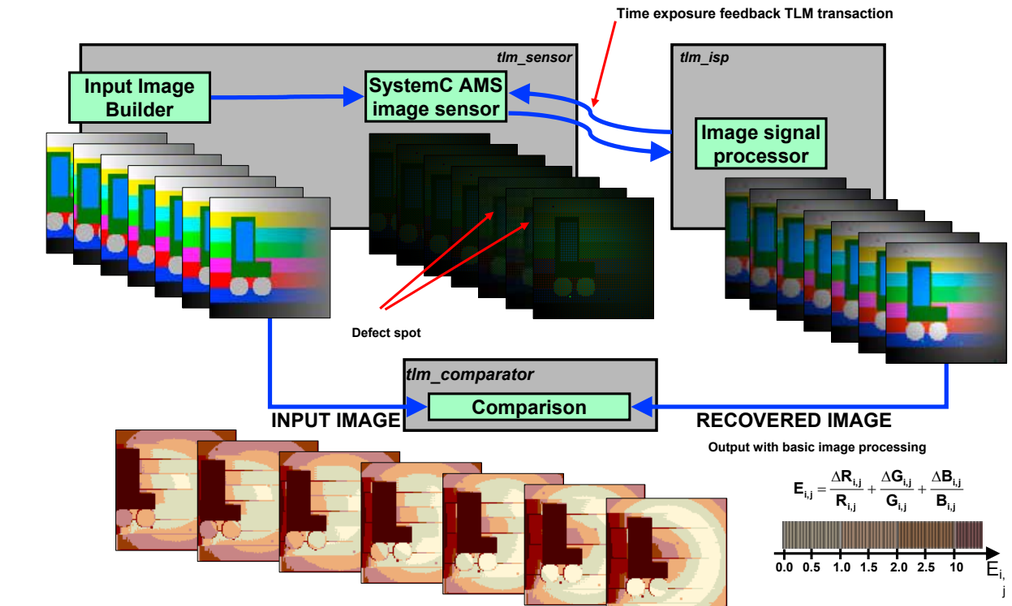
The principles



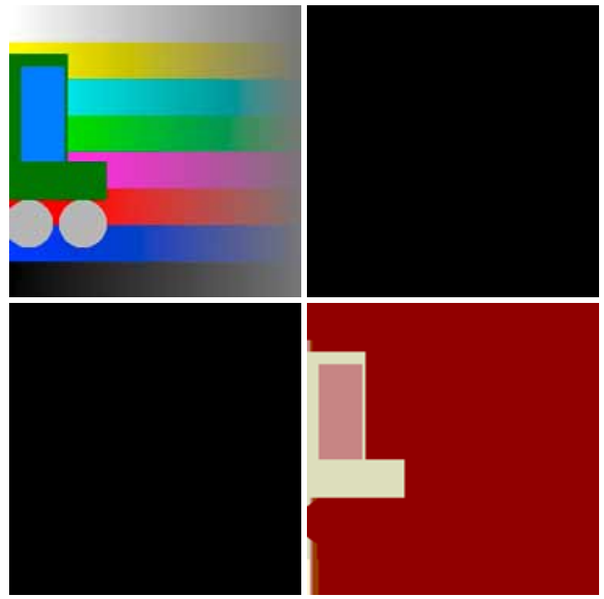
SystemC AMS TDF model evaluation

- Pros:
  - Simulation time depending on the abstraction level:
    - Fine modeling: 2minutes and 40sec for a 48x48 array (130 times faster than VHDL-AMS model)
    - Coarse modeling: 7sec for a 1920x1080 array (no more waveforms)
  - Easy to enrich the model with other computation greedy aspects
  - The model is integrated and interacts with a virtual SystemC TLM platform
- Cons:
  - Behavioral modeling: TDF MoC not intended to represent transistor-level non-idealities such as dark current or noise sources
  - No traceability of the waveforms for the rapid model

Platform simulation results



Platform demonstration



Conclusions

- A SystemC AMS TDF model of an image sensor has been developed
- First successful integration in STEricsson's Smart Image Architecture SystemC TLM virtual platform
- Simulation results are in line with the expected performances:
  - 2 mega pixels in about 7 seconds
- Embedded software development before the silicon is now a reality
- Future works and deployments:
  - Enhancement of the model through experimental data to improve the accuracy taking into account more physical aspects
  - Validation of ISP algorithms for different sensors
  - Architecture space exploration

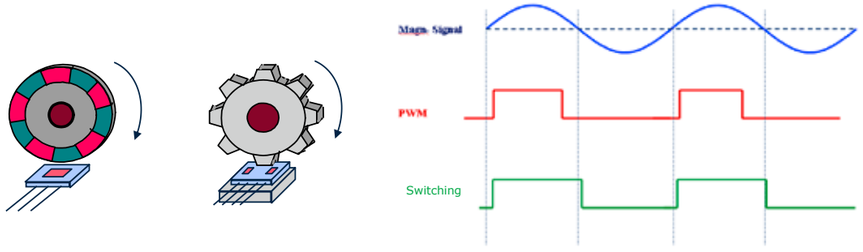
SystemC executable specification for magnetic speed sensors

Dresden 2011

Tobias Werth

Infineon Technologies Austria AG Villach

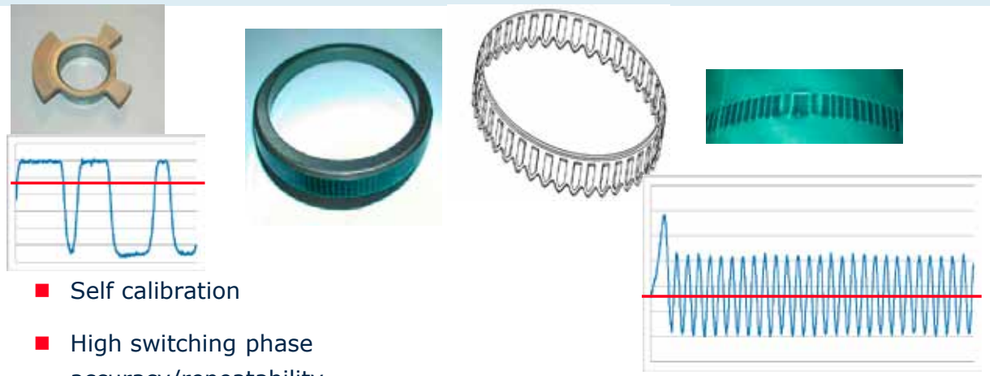
Magnetic speed sensor application principle



- Conversion of a passing coded target wheel into digital switching information.
- Magnetic sensors can work with magnetized rings or with ferrous target wheels and back bias magnet.
- Commonly used sensor elements are Hall or magneto resistive
- Automotive sensors are used in ABS, Crank, Cam or Transmission Applications

Copyright © Infineon Technologies 2009. All rights reserved.

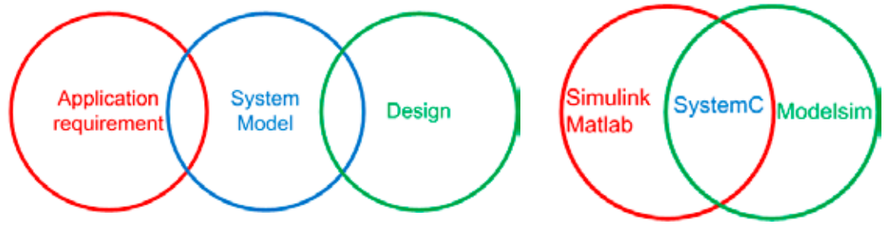
Magnetic speed sensor application requirements



- Self calibration
- High switching phase accuracy/repeatability
- Rotation direction detection
- High accuracy at first switching
- Suppression of disturbances
- Stability over temperature
- Sensor suitable for many different target wheel types
- Sensor configurable for many different applications
- High mounting tolerance - air gap capability

Copyright © Infineon Technologies 2009. All rights reserved.

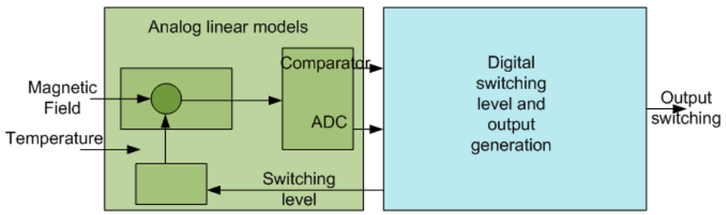
Linking the application requirements to the Design specification



- The system model link the application requirements to the design specification
- SystemC flow for executable specification
  - Simulink/Matlab used for application modeling due to high level approach and interfacing
  - Mentor/Modelsim included standard design flow with native SystemC support
  - SystemC model as common link to verify the application requirements and design
- Goals set for the SystemC flow
  - develope improvement in nonlinear digital regulation
  - Show the general behavior with different use cases
  - Simplify the verification during design phase

Copyright © Infineon Technologies 2009. All rights reserved.

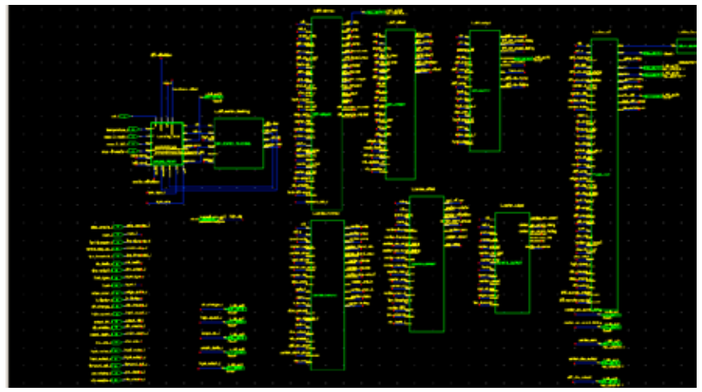
IC model blocks



- Basic modeling blocks
  - Magnetic field and temperature input
  - Linear analog amplification
  - Comparator
  - ADC, analog to digital conversion
  - Digital switching level regulations
  - Output interface

Copyright © Infineon Technologies 2009. All rights reserved.

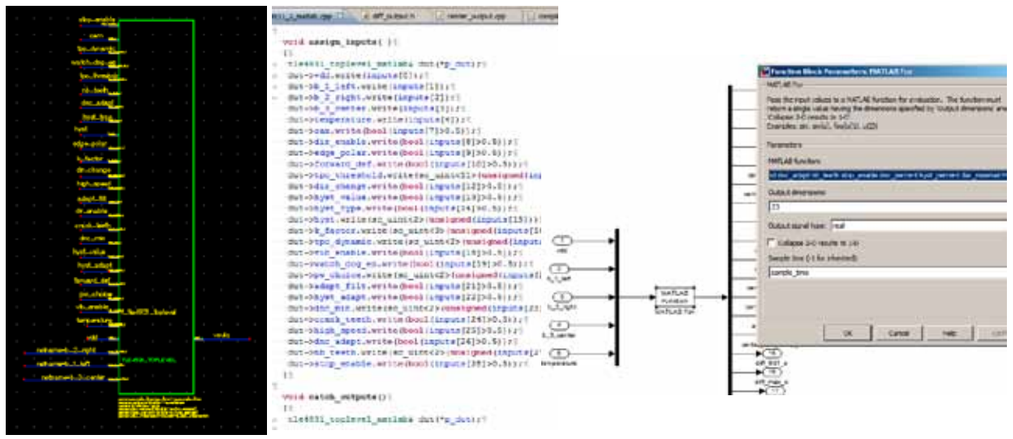
IC model in SystemC



- SystemC code on RTL level for digital part
- SystemC code ~1/2 of VHDL code
- Only sc_method processes were used due to limitation of the matlab wrapper
- Highest data rate was used in digital part
- No AMS or SDF SystemC extension were used for analog description

Copyright © Infineon Technologies 2009. All rights reserved.

Steps for SystemC/Simulink coupling

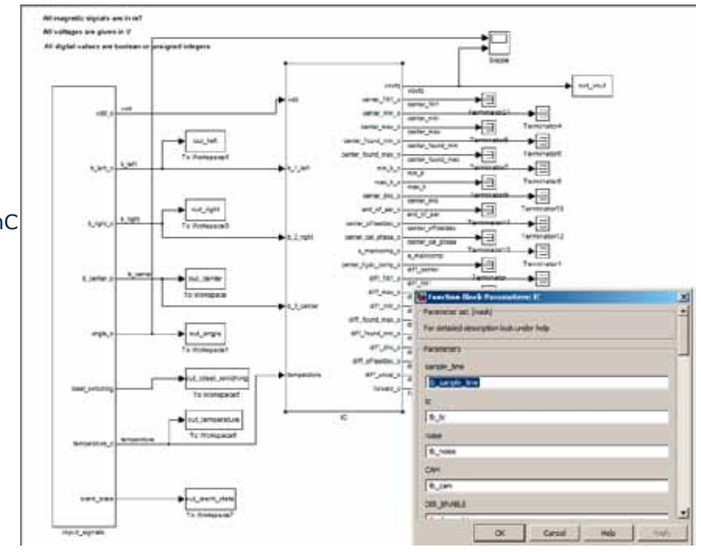


1. Generation of top level sc_module
2. Definition of inputs and outputs for simulink via mex function
3. Compilation and generation of dll with scripts provided by FHG
4. Matlab Fcn interface block definition

Copyright © Infineon Technologies 2009. All rights reserved.

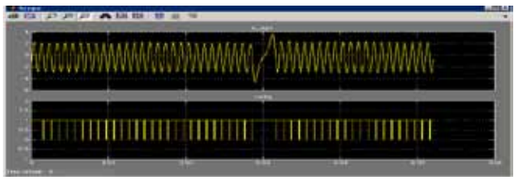
Application environment in simulink

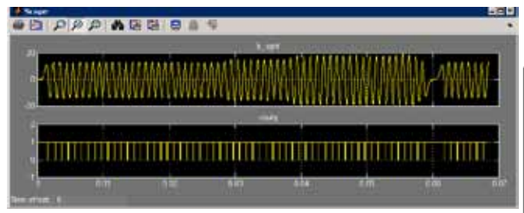
- Shared and co developed with external partner
- Conversion of physical parameter like air gap, rotation speed, target wheel and temperature into inputs for the SystemC model.
- Matlab scripting and calculations used for performance evaluation
- Easy debugging of algorithm due to availability of central internal signals

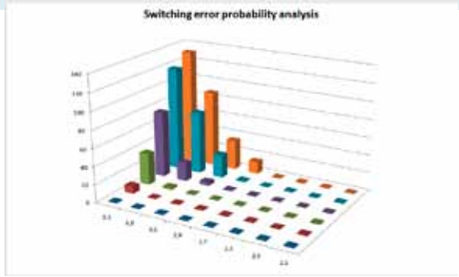


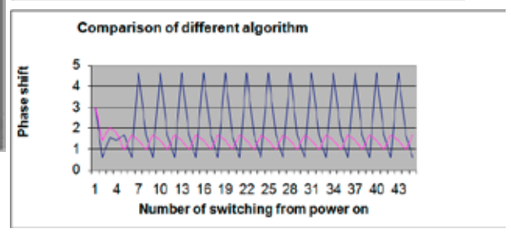
Copyright © Infineon Technologies 2009. All rights reserved.

Application requirement simulations with simulink









- Many different use cases analyzed
- Some improvements in digital algorithm identified

Copyright © Infineon Technologies 2009. All rights reserved.

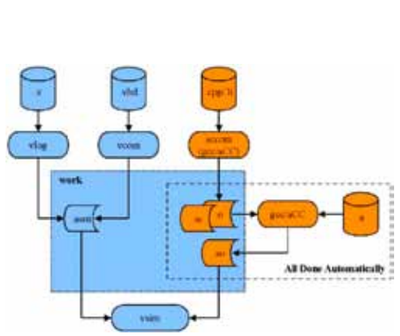
SystemC model during predevelopment

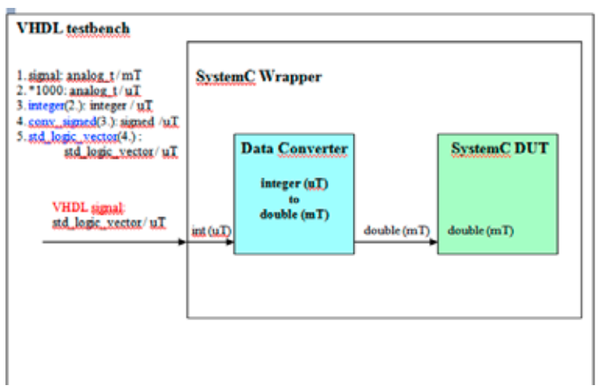
- SystemC model was shared with external partner
  - Many possible improvements identified early during the simulation result reviews with application experts
  - Some potential misunderstanding of written requirements were clarified
- High effort for the model generation
  - Bit and cycle true models generate high effort
- High effort of model verification

Copyright © Infineon Technologies 2009. All rights reserved.

SystemC integration into design flow

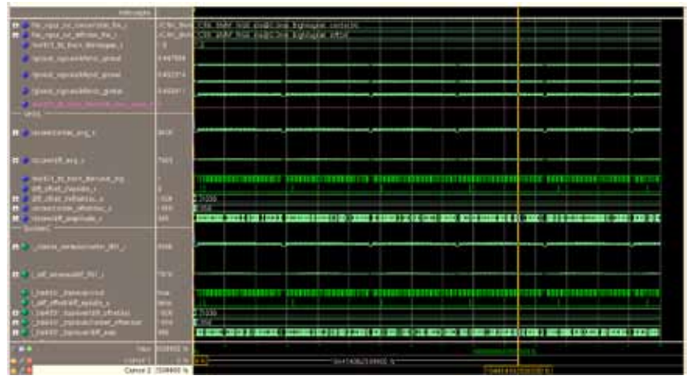
- SystemC compiled with MentorGraphics sccom
- VHDL wrapper around SystemC model to include it correctly into test benches
- VHDL wrapper generated with vgencomp





Copyright © Infineon Technologies 2009. All rights reserved.

Design verification with ModelSim



Error flag

Vhdl signals

SystemC signals

- SystemC model used as reference model
- One central error flag shows mismatch to reference
- Verification on wave form comparison of output and some central internal signals
- Easy debugging due to availability of all internal signals

Copyright © Infineon Technologies 2009. All rights reserved.

SystemC model experiences during Design verification

- Easy setup due to reduced SystemC language constructs (no sdf, no AMS)
- Easy verification due to nearly bit/cycle true behavior
- Easy debugging due to good reference model and comparable internal signals
- Many bugs also found in the SystemC model during design verification
- SystemC code base was ~1/2 of VHDL code base

Copyright © Infineon Technologies 2009. All rights reserved.

SystemC modeling Pros and cons

Pro

- Easy to set up interfaces
- Good integration into Matlab
- Really good integration in ModelSim
- Many algorithm improvements found
- Many potential bugs found during design verification
- Happy customer

Con

- Not al SystemC language constructs supported natively in Modelsim
- Effort for bit and cycle true models is high → double implementation of some parts
- sc_thread method generated problems in Matlab

Copyright © Infineon Technologies 2009. All rights reserved.

Introducing AMS Parts into TLM Virtual Platforms

Yossi Veller – Chief Scientist ESL

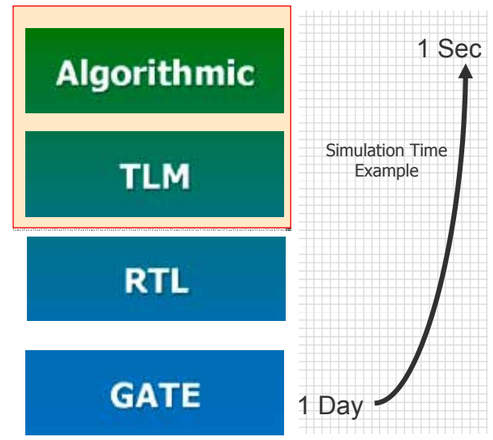
DCS

Mentor Graphics

Why ESL?

Electronic System Level (ESL): A set of **electronic hardware/software design methodologies** using **abstraction above RTL** for designing systems on chips (SoCs), FPGAs and boards

- Abstraction allows you to meet your performance, low power and cost design goals by:
  - Simulating the design functional spec in the system's complete context
  - Defining your HW and SW partitioning to meet design goals
  - Optimizing your multi-core architecture for Performance/Power
  - Quantifying power in the context of the system
  - Optimizing implementation using high-level synthesis
  - Optimizing your software to meet your design goals



© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

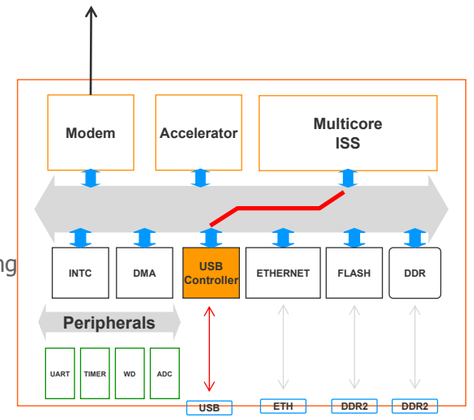
Driver For Change

- 3 factors driving product innovation:
  - System Integration (Audio, Video, graphics, Analog)
  - Superior performance, lower power, lower cost
  - Differentiated operating system and application software
- Trend: **Transition to Multi Core**
- Trend: **Simulation of the whole system including Analog**
- System trade-offs can provide power savings factor of 5x

© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

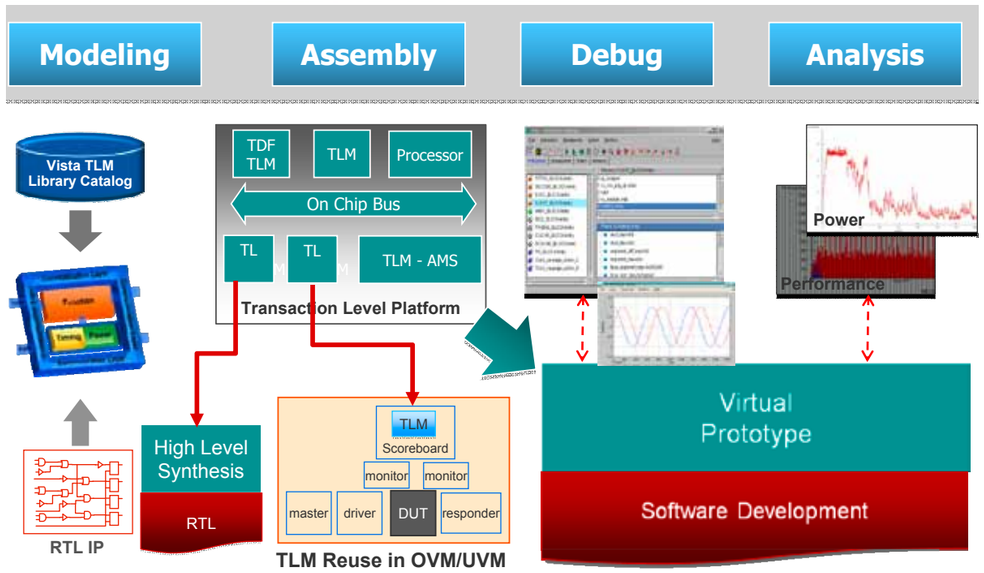
Designing the System Architecture

- Partition between HW and SW
  - Define functions implemented in SW
  - Define the HW "accelerator" components
- Design the SW multi-core configuration
  - Define initial number of cores
  - Number of processor ports
  - Define Cache & Memories layering & sizing
- Design the multi-core interconnect fabric
  - Identify bus protocol
  - Identify bus layering & arbitration
- Design the HW topology
  - Define new and reusable IP
  - Add peripheral and I/O busses



© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

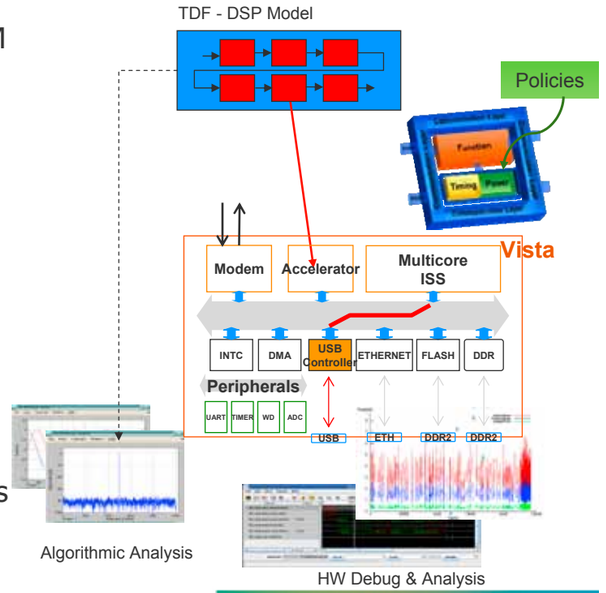
Vista Design Flow



© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

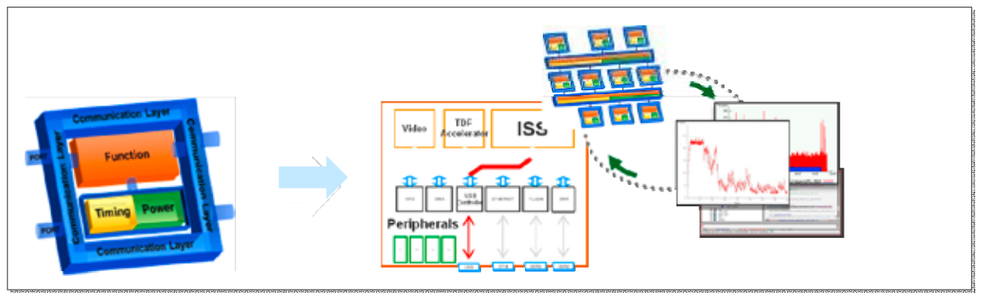
Integrated AMS Blocks

- Use AMS Blocks as the TLM Functional Models
- Attach Power Policies
- TLMs integrated into SoC Transaction Level Platform
- TDF algorithm analysis in the context of your Transaction Level Platform
- Include Analog TLM models in your system



© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

Creating A TLM Platform

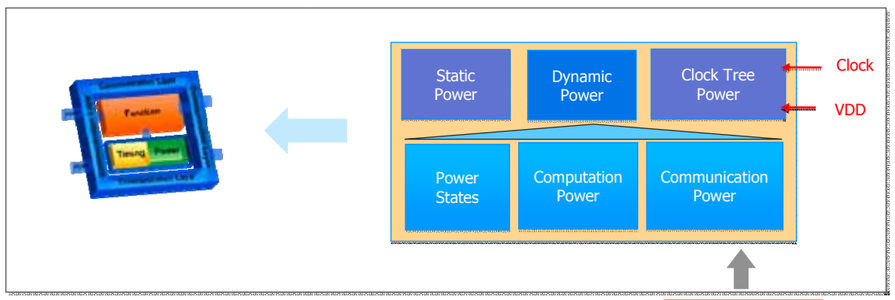


- Vista unique Scalable Modeling
- Standard Based (SystemC AMS / TLM-2.0)
- Fully functional
- Register / Bit compatible
- Timing/Power Policies Layer
- Dynamically switch timing abstraction (LT/AT)
- Quick exploration turn-around

- Model Catalogue
- Fast Processor Models
  - GDB Compliant

© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

Vista Power Foundations



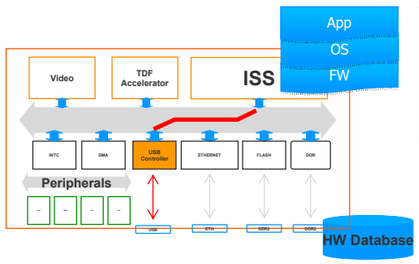
- Vista Power Focus targeting ***SW developers*** and ***HW architecture*** analysis for low power optimization
- State-of-the-art Power Modeling
  - Computation & Communication Power
  - State Power
  - Supports dynamic voltage/frequency scaling (DVFS)
- Automated Power Extraction

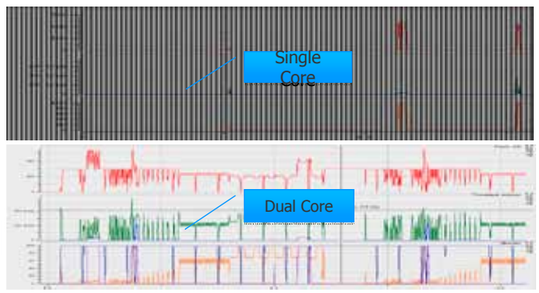
+ Bottom-up Power Modeling

© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

SW engineer Use Case for multi-core

- Explore multi-core configurations
- Explore various software parallelization techniques
- Early test of data regularity for optimized cache access
- Early-stage power estimation for most efficient thread partitioning

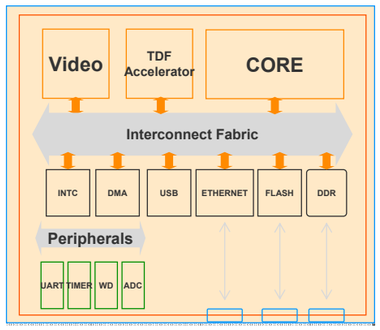


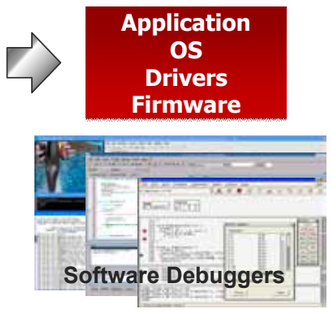


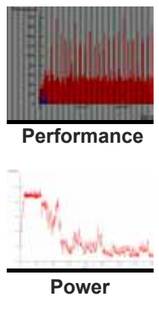
© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

Virtual Prototyping

- Deliver a target HW model to the software team before RTL
- Integrate operating system, middleware and application software
- Validate and debug software against actual hardware architecture
- Tune software to meet performance and power requirements







© 2010 Mentor Graphics Corp. Company Confidential
www.mentor.com

Using IEEE 1685 Standard (IP-XACT) for Managing AMS Design Flow Based on SystemC AMS

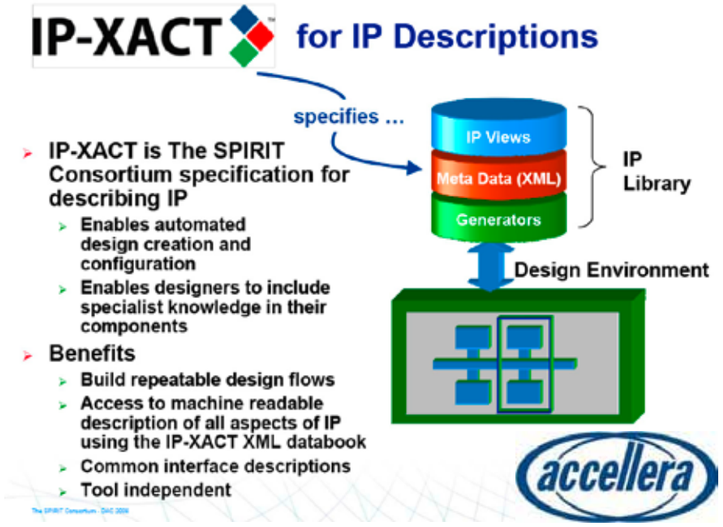
Agenda

- IP-XACT at a glance
- AMS design flow state of the art and issues to be solved
- Proposal of enhancements in the AMS flow using IP-XACT and tooling
- IP-XACT standard extensions for AMS
- Conclusion

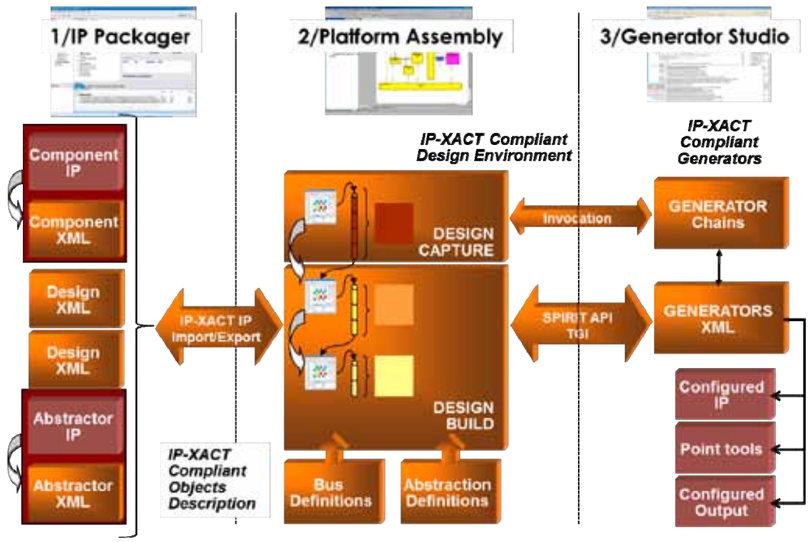
Agenda

- IP-XACT at a glance
- AMS design flow state of the art and issues to be solved
- Proposal of enhancements in the AMS flow using IP-XACT and tooling
- IP-XACT standard extensions for AMS
- Conclusion

IP-XACT at a glance



IP-XACT in system design tools

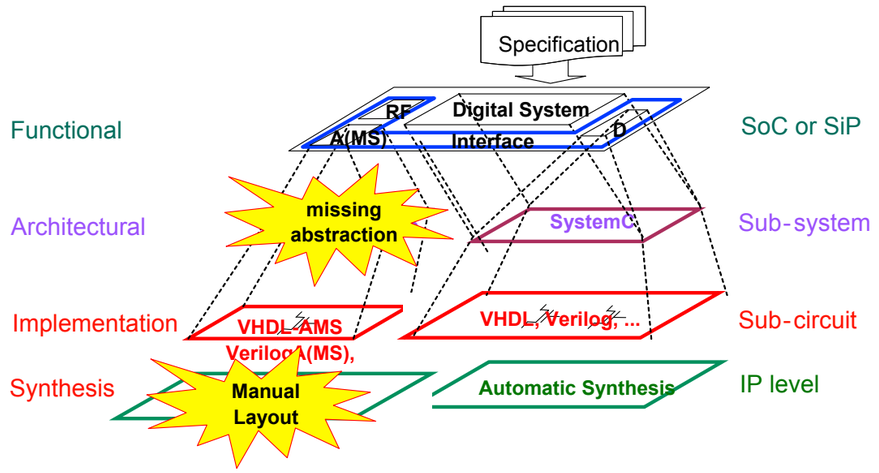


Agenda

- IP-XACT at a glance
- AMS design flow state of the art and issues to be solved
- Proposal of enhancements in the AMS flow using IP-XACT and tooling
- IP-XACT standard extensions for AMS
- Conclusion

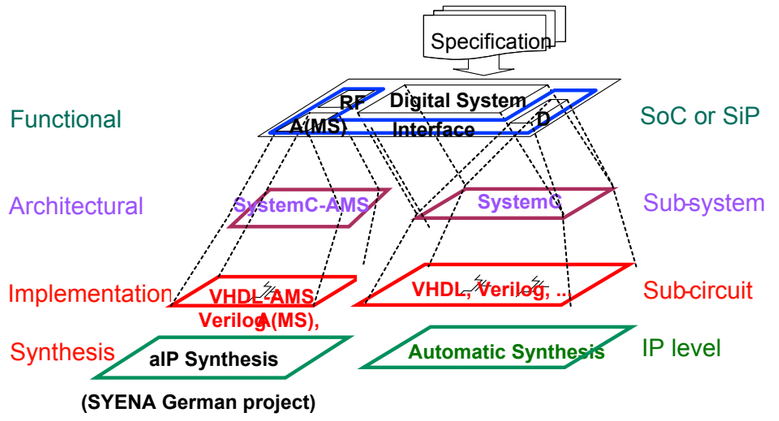
Mixed design flows state of the art

- Two abstraction levels are currently missing :



Using SystemC-AMS

- SystemC provide missing levels for a complete AMS platform prototyping



Main issues to be solved in the AMS flow

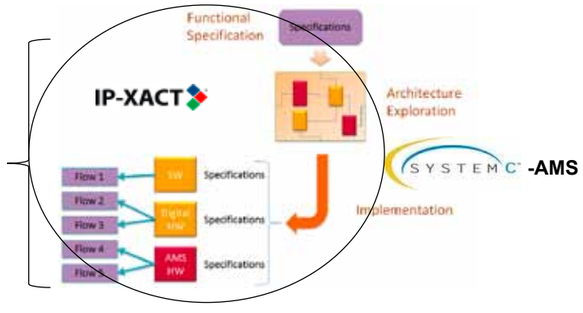
- **The system is made of several sub-systems**
- **Separate design flows: Analog, RF, Mixed, Digital, ...**

Flow separation is mainly caused by the use of specific tools required by the different disciplines.

- **Two major flows exist**: digital/system and analog/RF/AMS.

These flows are not completely separated: overall system specification and integration needs to dispatch / merge data to / from the different flow.

System integration



Main issues to be solved in the AMS flow

- **Description of the system specifications: where do they came from, format, how their realization is monitored along the flow, ...**
- **Specification in the AMS flow are Assembly, functional and interfaces specifications. They are used across several teams (spread in the world).**

- **Specification are expressed though the following formats :**

- Human readable specifications (mathematical expression or not)
- Matlab Simulink model and analysis
- **SystemC TLM, SystemC AMS (specs for behaviour and structure)**
- **IP-XACT can be a used as specification entry (specs for structure, configurations, parameters, tools)**

- **No standardized machine readable format**

To exchange specification between architecture exploration and design implementation, and to monitor them along the flow.

- **Our proposal is to use the IEEE 1685 standard (IP-XACT).**

Main issues to be solved in the AMS flow

- **Refinement and languages: Matlab -> Vhdl -> Spice -> Post layout?**

- Refinement process is not necessary: many industrials perform the implementation directly from the specifications without the refinement step.

the architect gives the definition at top level, and then analog AMS are specified at the block level.

- Functional level is described in Matlab, SystemC, SPW or Excel,

- Architecture level can be SystemC-AMS based, and implementation level is done using VHDL or Verilog, then Spice.

➢ SystemC AMS is not replacing these implementation languages, but is a new mean for architecture exploration and performance analysis
➢ IP-XACT may be used to link these several languages through a unified/standardized way to express specifications

Main issues to be solved in the AMS flow

- **Tools and characterization**

The implementation from mathematical specifications described in a PDF document may be performed using e.g. Agilent ADS and Cadence suites.

After the characterization is realized with e.g. Eldo RF or ADS, Spectre RF

Specifications may be back annotated with real characteristics, but sometimes, back annotation is never propagated to the Matlab level.

Possibility to validate characteristics against the specifications at block level, subsystem level, and at system level..

➢ SystemC AMS models and IP-XACT description may be used for enhancing characterization and validation at system level

Main issues to be solved in the AMS flow

- **Usage of IP generators (configurable IP)**

IP generators are not used for analog functions due to two main reasons: the libraries are not rich enough for each technology, and this solution is not mature.

Experimentations are under development, based on the usage of firm-IP. Analog design is done manually.

➢ Maybe simulatable specification in SystemC-AMS and synthezisable specification (with or without IP-XACT) can facilitate usage of IP generators

Main issues to be solved in the AMS flow

- **IP-Reuse: with the same technology, with different technology, with external IP provider**

It is commonly admitted that an analog block can only be reused for the same technology.

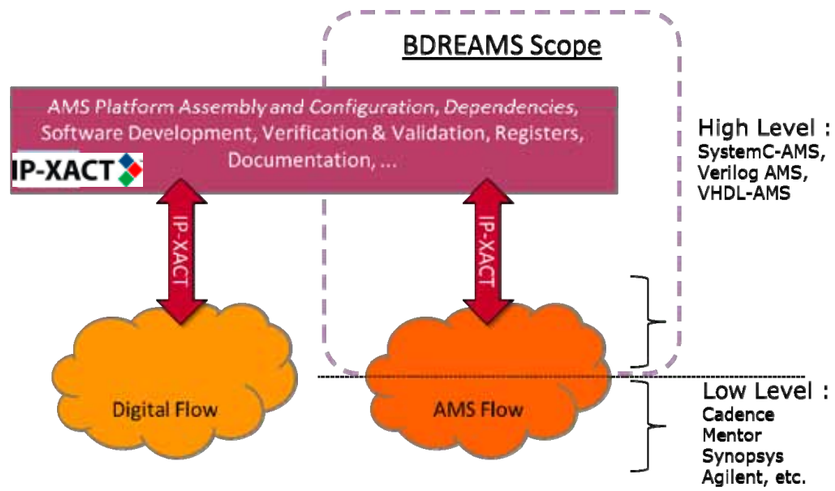
Unlike digital blocs that can be synthesized from the RTL level, an analog block has to be re-implemented from scratch.

Firm-IP can, under certain circumstances, help to reuse an analog IP with similar technologies.

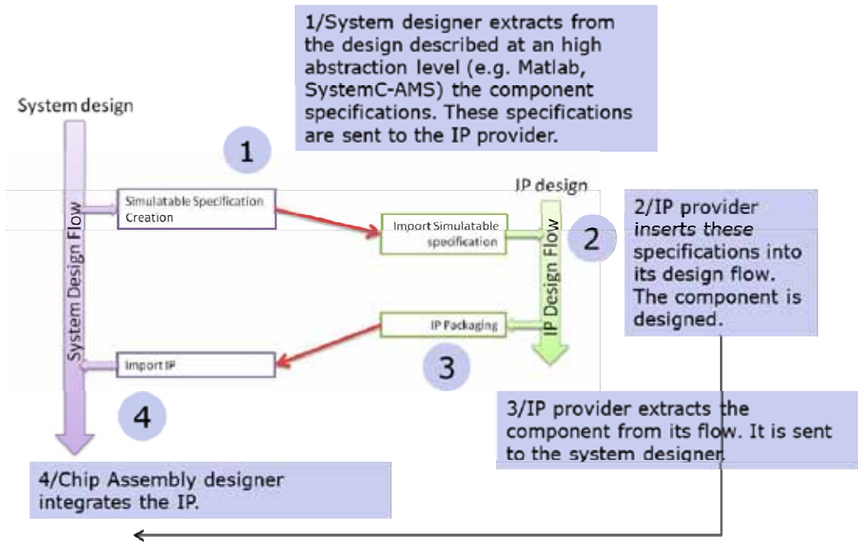
The only analogue components that are massively reused are the IOs (e.g. USB interface), or the ones provided by the technology provider.

➢ Model Reuse can be envisaged with SystemC AMS and IP-XACT

Connecting Analog and digital flows



System design vs IP design

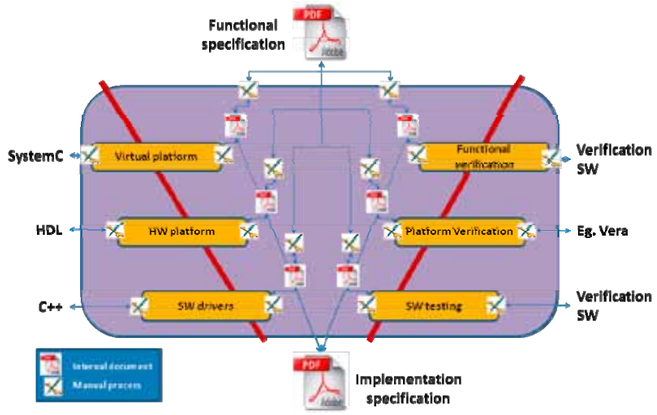


Agenda

- IP-XACT at a glance
- AMS design flow state of the art and issues to be solved
- Proposal of enhancements in the AMS flow using IP-XACT and tooling
- IP-XACT standard extensions for AMS
- Conclusion

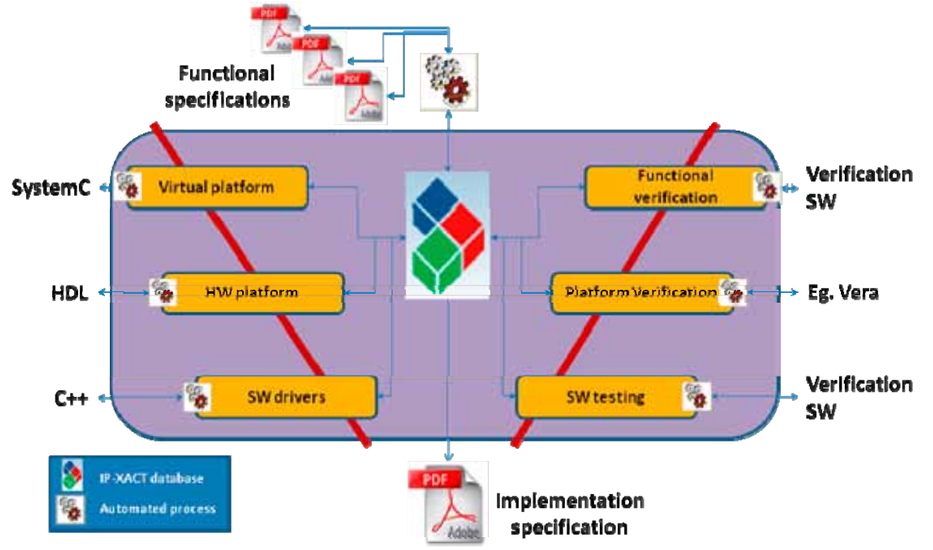
Inconsistencies of specification in process cycle

All teams refer to a written specification and create from this paper documentation the appropriate representation based on their specific activity language and to create the appropriate environment for their tools set.



- As all those steps are performed manually: they are error prone and difficult to maintain in regards with update or modifications
- The interpretation of these documentations is not always straight as each of the domains use specific conventions which may not be shared by the other actors
- A lot of possible misunderstanding, undocumented or inconsistent information and consume a lot of time and effort.

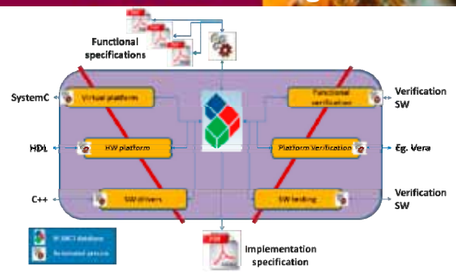
Using a common backbone for information management



Using a common backbone for information management

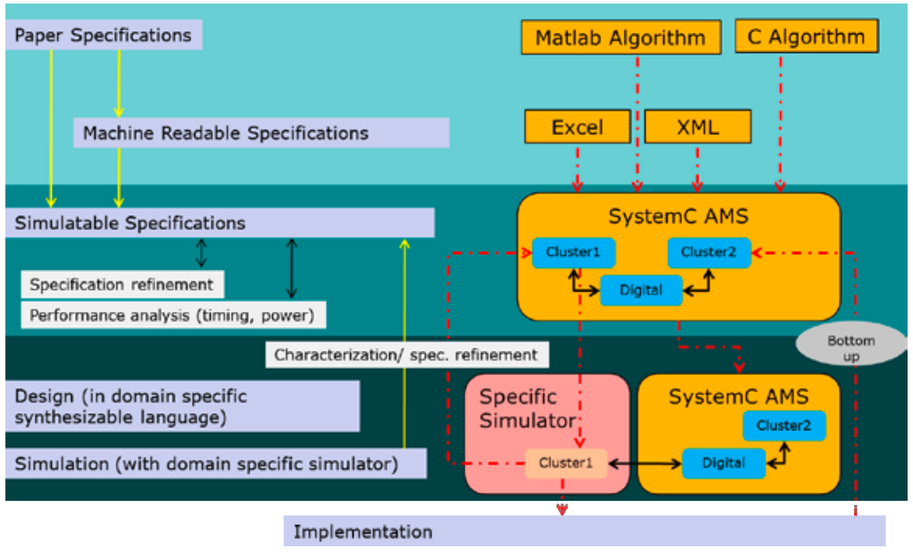
A standard is available for answering to this issue:

IP-XACT IEEE 1685 (XML schema)

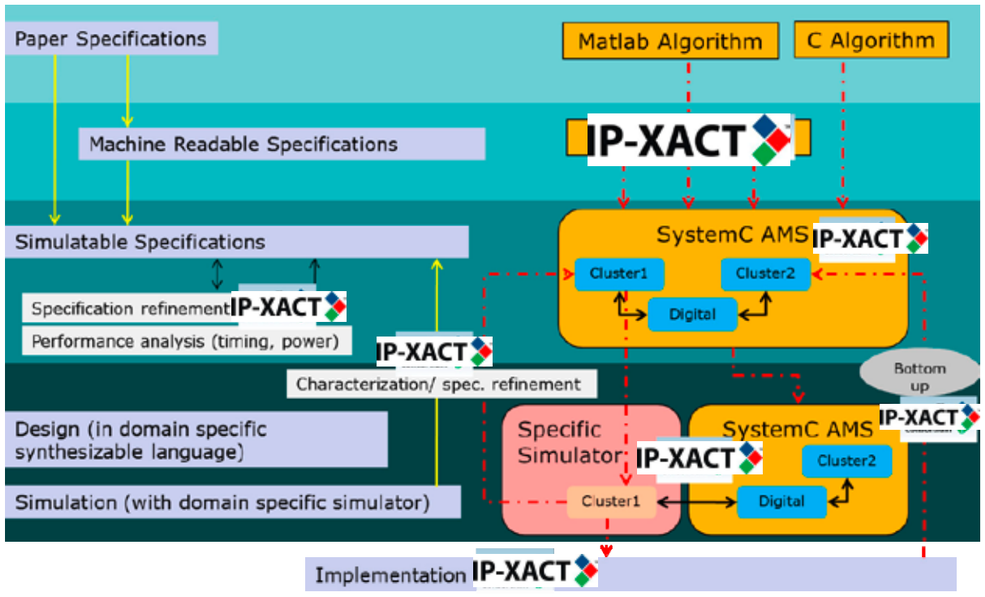


- Commercial solutions exist and provide a complete information backbone to cope with information exchange, synchronization and traceability.
- Share the same information between all the actors, use a common language to describe this information, automate the generation of multiple formats depending on the task needs, automate the update of the golden reference from multiple formats, compute impact of a modification on specifications at any level of the process, perform checks between steps.
- Relies on a computer readable format to exchange all those information and results in a single shared specification which will be enriched at each steps of the process.
- Sharing a common way to describe the exchanged information and use an automated process to extract the piece of information that is relevant for their activity and to automatically translate it into their domain languages or specific tool representations.

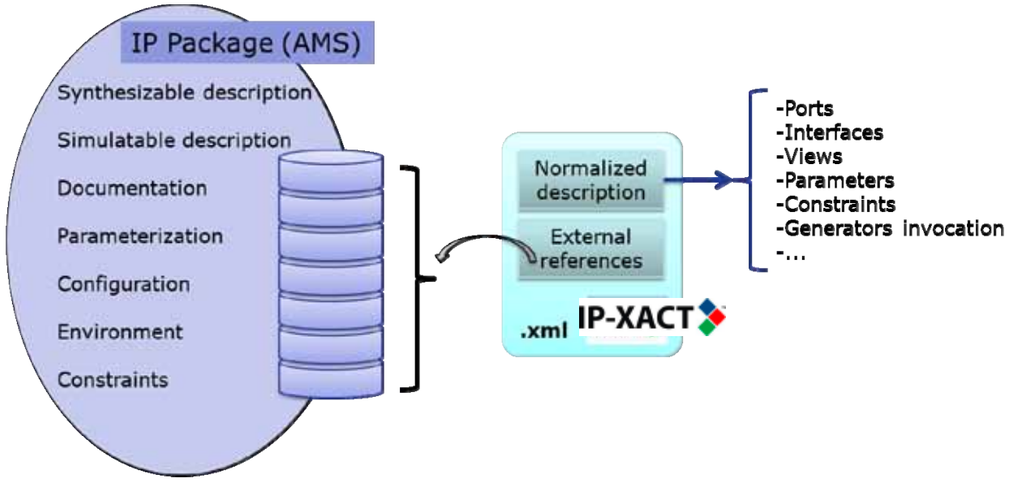
SystemC AMS Design flow



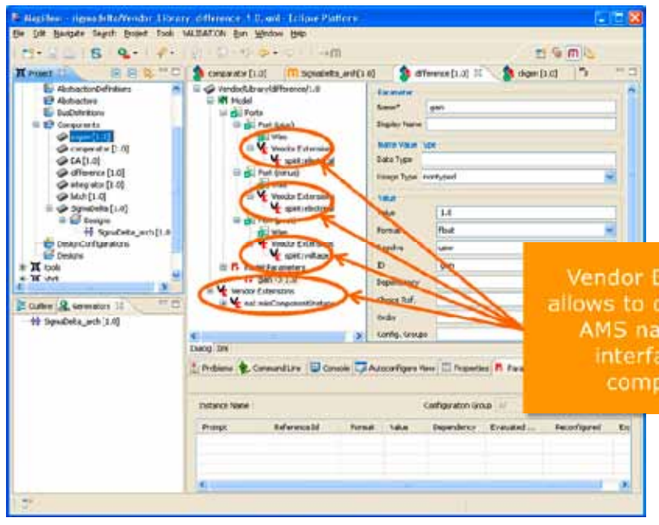
SystemC AMS Design flow using IP-XACT



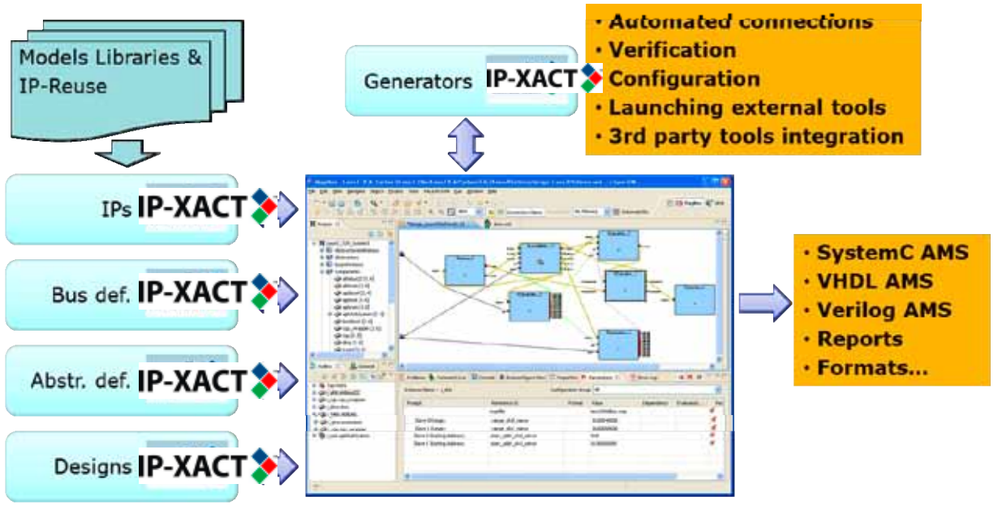
Extensions of IP-XACT standard for AMS



Extensions of IP-XACT standard for AMS



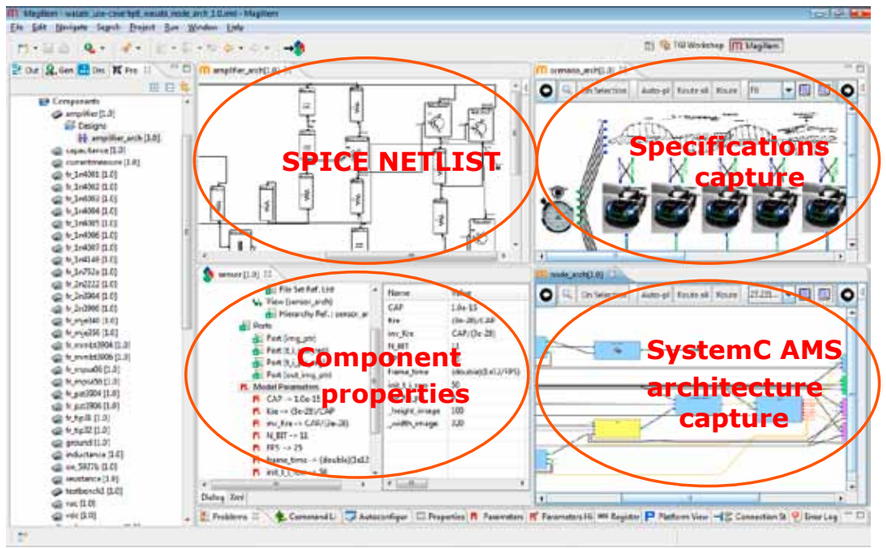
AMS design framework



Agenda

- IP-XACT at a glance
- AMS design flow state of the art and issues to be solved
- Proposal of enhancements in the AMS flow using IP-XACT and tooling
- IP-XACT standard extensions for AMS
- Conclusion

Design Framework



Description of heterogeneous analogue and mixed signal (AMS) IP's and systems in IP-XACT

- IP-XACT is dedicated to the description of digital IPs and platforms
- Possible to use IP-XACT without modification for the structural description of AMS blocks and systems
- Some attributes are missing for supporting properly some specific functionalities of an AMS integration and verification flow.
- Motivation: no way to specify, integrate or validate properly the integration of several AMS IPs or subsystems and especially with digital systems.
- IP-XACT schema should be extended to support some few additional specifics of AMS blocks. This will allow the description, at several levels of abstraction of an heterogeneous system and thus to manage characteristics propagation, assembly and interoperability issues, management of heterogeneous languages and simulation issues, etc.
- Extensions in Accellera Consortium

Possibility to tag each port of a component with the corresponding “Domain” and “Signal Type Def”

- Domain of the port: it specifies the physical support for the transmitted information. The following domains can be listed for compatibility with IEEE / VHDL-AMS: Mechanical, Thermal, Electrical, Radiant, Fluidic
- The Signal type def. is used to define how the signal transmitted through the port is taken into account for simulation: continuous or discrete time and continuous or discrete value.

Parameters related to a single port

- Many operations in the flow require getting values for different parameters on a port.
- In the current IP-XACT release, this information is not available: there is no location to store a parameter related to a particular port.
- Even for digital design this is a lack that customer complains about: actual circuit contains many power domain. The common operation that consists to verify whether a voltage adaptor should be inserted must rely on the voltage on each port.
- For an AMS component, the number of data (timing, electrical, power, etc.) associated to each port may be more than a simple voltage.
- For example, the description of an electrical analog output port may include DC voltage, AC voltage, maximum current, capacitance, and so on.
- A common mechanism must be used to associate a parameter to a port: We propose to add in IP-XACT schema the capability to define a parameter list on each port.

Default value for the non connected ports

- The current IP-XACT schema allows specifying a default value for an unconnected input port of a component.
- As IP-XACT targets only digital circuit, the format for this value is “scaledNonNegativeInteger”.
- In an AMS context, common data type transmitted through a port is based on real numbers (voltage, pressure, lux, etc).
- The default value for a non connected input port should support any kind of value.
- Replace the format “scaledNonNegativeInteger” by a more general format (e.g. double)

Units and prefixes for parameters

- If we want to support the description of AMS blocks and systems, we need to define units and prefixes for all values in IP-XACT schema.
- We propose to add in the IP-XACT schema, two attributes in the parameter definition:
  - Prefix (e.g. Kilo)
  - Unit (e.g. Ohm).
- For the list of legal values, we will follow the SI units for allowed values.

Agenda

- IP-XACT at a glance
- AMS design flow state of the art and issues to be solved
- Proposal of enhancements in the AMS flow using IP-XACT and tooling
- IP-XACT standard extensions for AMS
- Conclusion

Conclusion

- SystemC AMS is good solution for virtual prototyping at system level
- Enhancements in the global design flow are required to support specification management and flows interactions
- IP-XACT is used today for digital flow and shall be used to support AMS flow
- Extensions in the standard (Accellera)
- Emergence of a new generation of tools

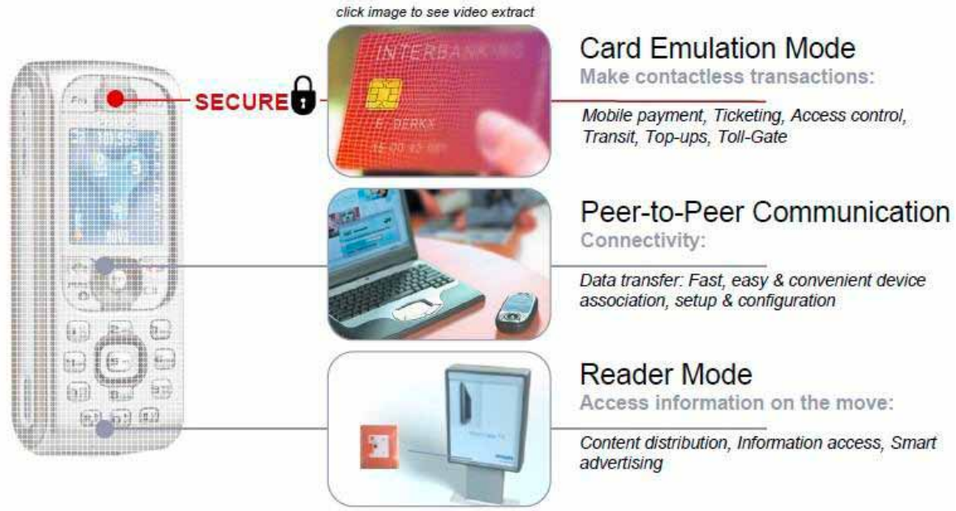
SystemC/-AMS system level model of a Near Field Communication (NFC) radio front-end

Bas Arts
NXP Semiconductors

NFC Technology at a Glance

- Short-range (~ 10 cm), 13,56 MHz secure contactless technology
- Standardized in ISO 18092, ECMA and ETSI
- Compatible with existing ISO 14443 contactless cards & reader infrastructure
- Reader and card mode modes possible in same device
- Device-device connectivity
- Data exchange rate up to ~~424~~ 848kbit/sec

NFC – three application families



RF and wired standards (I)

- NFC RF standards:
  - ISO 14443 Type A/B (card / reader, “proximity device”)
  - FeliCa (card / reader, “proximity device”)
  - ISO 15693 (card / reader, “vicinity device”)
  - NFC-IP1 (peer to peer, active/passive)
- NFC wired standards:
  - NFC-WI (communication to secure element)
  - SWP (communication to SIM card)

RF and wired standards (II)

OSI Model			
	Data unit	Layer	Function
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation and encryption
		5. Session	Interhost communication
	Segment	4. Transport	End-to-end connections and reliability
Media layers	Packet	3. Network	Path determination and logical addressing
	Frame	2. Data Link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission

example:

HTTP

TCP

IP

Ethernet / 802.11g

RF and wired standards (III)

- What do these wireless standards define?
  - Physical characteristics
    - antenna dimensions
    - field strength
    - carrier frequency
    - modulation type
    - bit representation & coding
  - Data link characteristics; error detection & correction
    - byte format
    - framing
    - initialization & collision protocols
    - transmission protocols

RF and wired standards (IV)

- NFC RF modulation is based on Amplitude Shift Keying (ASK)
- Bit coding is either
  - Modified Miller: position of pulse within bit frame
  - NRZ: one of two defined physical states
  - Manchester: order of sequence of two defined physical states

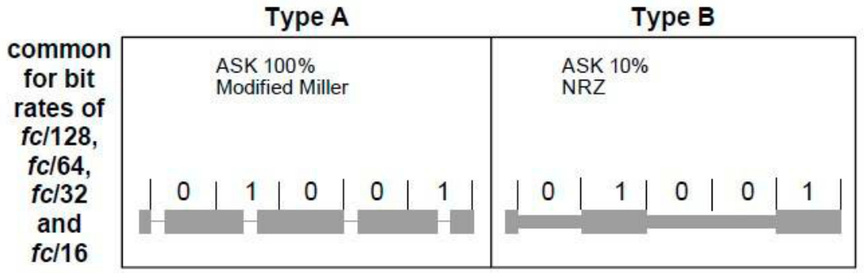
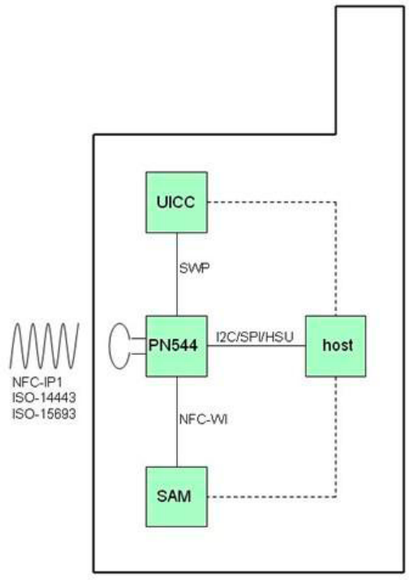


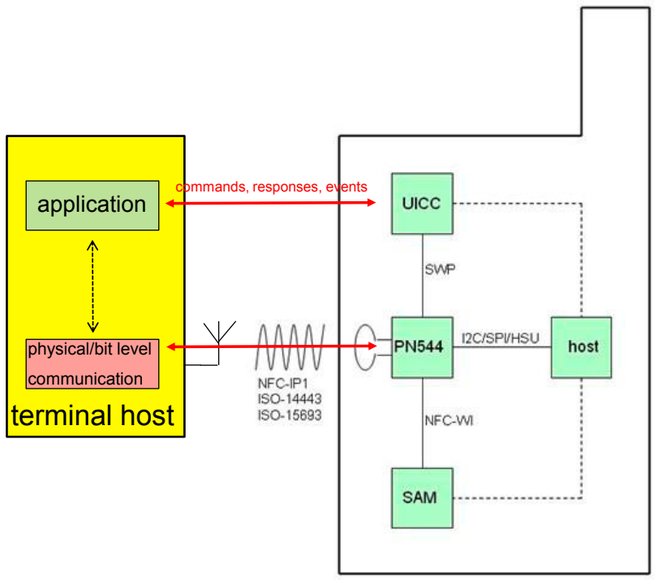
Figure 1 — Example PCD to PICC communication signals for Type A and Type B interfaces

NXP PN544 NFC controller inside phone

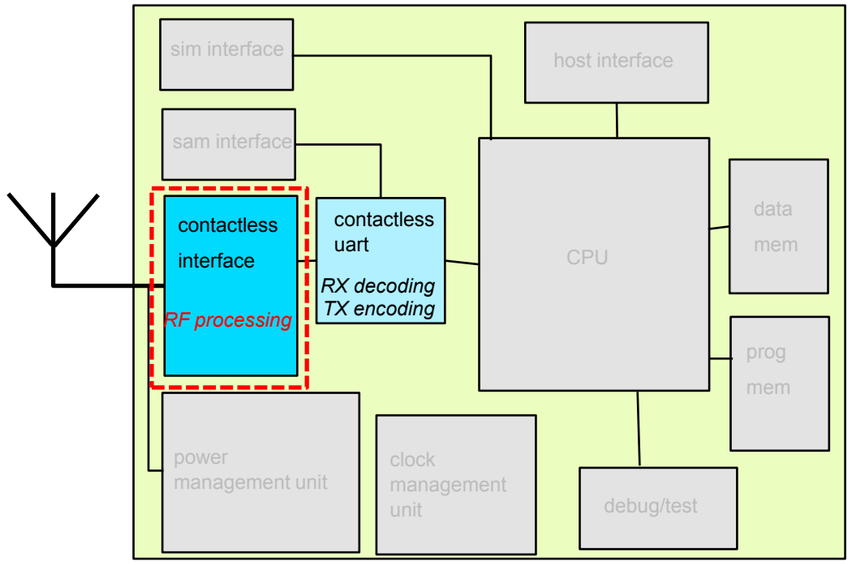


NXP PN544 NFC controller inside phone

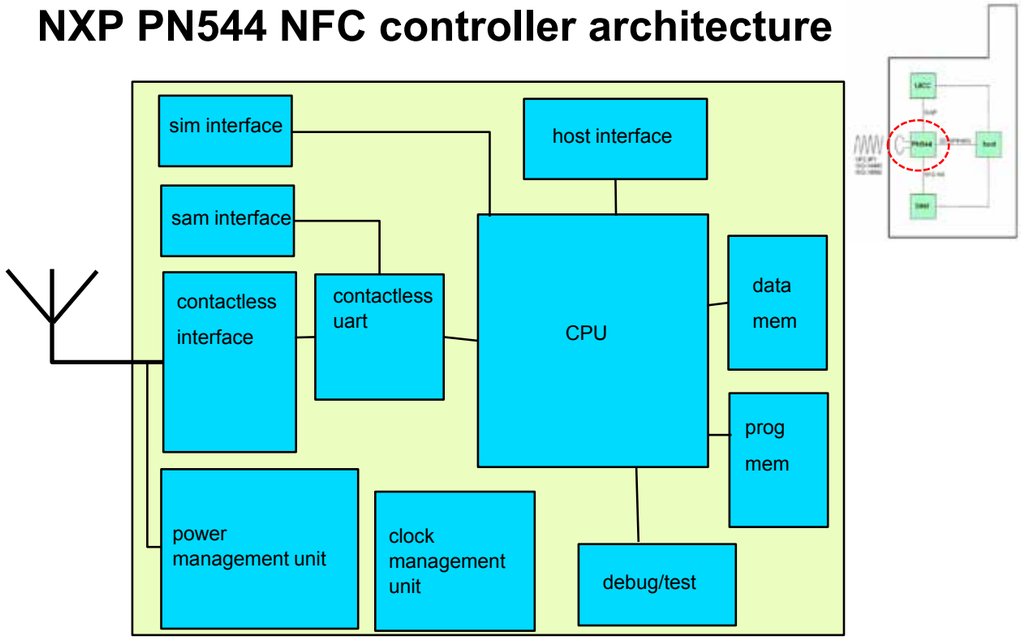
OSI Model			
	Data unit	Layer	Function
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation and encryption
		5. Session	Interhost communication
	Segment	4. Transport	End-to-end connections and reliability
Media layers	Packet	3. Network	Path determination and logical addressing
	Frame	2. Data Link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission



NXP PN544 NFC controller analog front-end



NXP PN544 NFC controller architecture



Use cases and methodology for system level model of analog front-end

- Use cases (why do we need a system level model?)
- Modeling requirements for the use cases
- Modeling methodology / concepts (reuse, configurability)

Use cases

- Why do we need a system level model?
  - Analog architecture exploration
  - Software development (PHY control sw)
  - Verification of IC implementation

Model abstraction requirements per use case

	Architectural exploration	Software development	IC verification
Structure	Detailed	Abstract	Abstract
Behaviour	Abstract	Abstract	Detailed
Communication	Abstract	Detailed	Detailed
Timing	Abstract	Abstract	Detailed

- One reusable model?

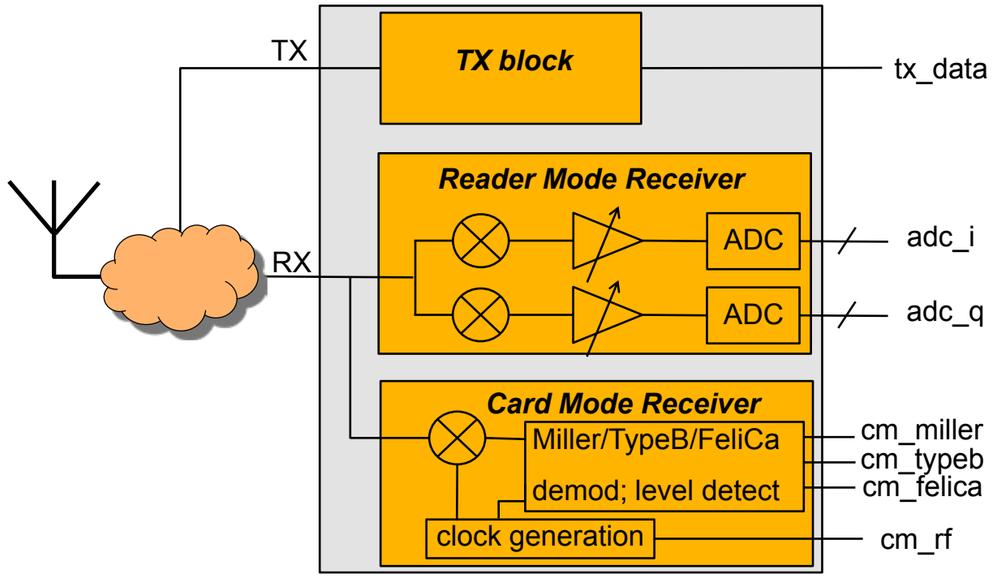
Modeling concepts

- Structure:
  - Keep functional blocks like mixer, BBA, ADC
- Behaviour
  - Ideal functional including some analog properties (gain, cut-off frequency)
- Communication (interface)
  - No control pins for architectural exploration
    - Configurable control settings (see next slide)
- Timing
  - Configurable timestep for analog parts
  - Cycle accurate modeling digital parts
    - Dependencies on clock signals (mixer, ADC,)…
  - Configurable #samples/period for artificial input signal

Modeling concepts

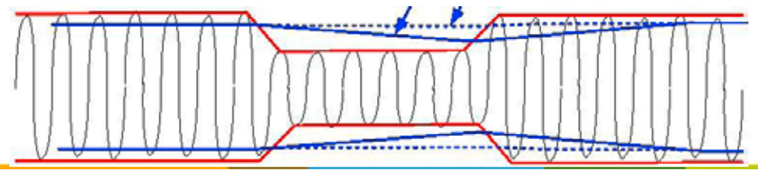
- Configurability:
  - Run-time: SCML properties
    - Replace control behaviour with fixed settings (gain settings, cutoff settings, lock signals etc)
    - Enables exploration (gain values, cutoff values, time constants, peak detection delay etc.)
    - Set simulation characteristics like analog timestep, #samples/period
  - Compile-time: `#ifdef`
    - Include/exclude pins + associated behaviour

Analog front-end architecture (datapath)



Analog front-end

- Reader Mode Receiver
  - Uses external 13.56 Mhz clock for sampling
  - Sample-and-hold mixing
  - Digitized I and Q signals sent to digital processing block
- Card Mode Receiver
  - Sampling clock generated from input signal
    - ¬ no fixed timestep
    - ¬ analog circuit needs time to lock sample point
  - Generates reference level to decide “0” / “1”



Analog front-end model - configuration

```
SC_MODULE(Cardmode)
{
    public:
        /* analog signal inputs */
        sc_in <double> RX_port;
        sc_in <double> Vmid_port;

        /* digital signal outputs */
        sc_out <bool> felica_port;
        sc_out <bool> typeB_port;
        sc_out <bool> miller_port;
#if EXPLORATION_USECASE || VERIFICATION_USECASE
        sc_out <bool> typeB_pre_port;
        sc_out <bool> miller_pre_port;

        /* analog signal outputs (test purposes) */
        sc_out <double> typeB_differential_vref_port;
        sc_out <double> miller_differential_vref_port;
#endif
```

```
#if EXPLORATION_USECASE || MINIMAL_INTEGRATION_USECASE
        /* SCML properties */
        scml_property <double> felica_offset_prop;
        scml_property <double> felica_hyst_prop;
        scml_property <double> typeB_adjustment_factor_prop;
        scml_property <double> typeB_time_constant_prop;
        scml_property <double> miller_adjustment_factor_prop;
        scml_property <double> miller_time_constant_prop;
        scml_property <bool> miller_en_lock_prop;
#endif
```

```
#if EXPLORATION_USECASE || MINIMAL_INTEGRATION_USECASE
    felica_offset_prop("felica_offset_prop", 0.0),
    felica_hyst_prop("felica_hyst_prop", 0.027),
```

```
scml_simple_property_server myserver(configpathstring + "Properties.ini");
scml_property_registry::inst().setCustomPropertyServer(&myserver);
```

```
############
# Cardmode #
############
[double]
pn544_analog_top.cardmode_module.felica_offset_prop : 0.0
pn544_analog_top.cardmode_module.felica_hyst_prop: 0.027
```


Analog front-end model - BBA

```
SCA_TDF_MODULE(Rcvbba)
{
    public:
        /* signal input ports coming from digital part */
        sca_tdf::sca_de::sca_in <double> signal_in_port; /* Port for signal coming from mixer */
#if VERIFICATION_USECASE
        /* control input ports coming from digital part */
        sca_tdf::sca_de::sca_in <sc_uint<2> > gain_port;
        sca_tdf::sca_de::sca_in <sc_uint<2> > hpcf_port;
        sca_tdf::sca_de::sca_in <bool> hp_lowf_port;
        sca_tdf::sca_de::sca_in <bool> oc_enable_port;
#endif

        /* output ports going to digital part */
        sca_tdf::sca_de::sca_out <double> signal_out_port; /* Port for signal going to ADC */

        /* filter functionality */
        sca_tdf::sca_ltf_nd ltf; /* Laplace transform for computing rcvbba transfer function */
        sca_vector <double> num; /* Numerator coefficients for Laplace transform */
        sca_vector <double> den; /* Denominator coefficients for Laplace transform */

#if VERIFICATION_USECASE
        Rcvbba(const sc_module_name & name, const sc_time & ts) {timestep = ts; };
#elif EXPLORATION_USECASE || MINIMAL_INTEGRATION_USECASE
        Rcvbba(const sc_module_name & name, const sc_time & ts) :
            gain_prop("gain_prop", 28.18),
            flow_prop("flow_prop", 43e3),
            fhigh_prop("fhigh_prop", 2.6e6)
    {
        timestep = ts;
```

Analog front-end model - BBA

```
#if VERIFICATION_USECASE
        /* read gain/hpcf control signals, set indices and set gains/frequencies */
        unsigned gain_index = gain_port.read();
        unsigned hpcf_index = hpcf_port.read();

        current_gain = gain_table[gain_index][hpcf_index];
        current_flow = flow_table[gain_index][hpcf_index];
        current_fhigh = fhigh_table[gain_index][hpcf_index];

        /* set (de)numerators for filtering
         *
         *                          (2*pi*fhigh - 2*pi*flow) * s
         * H(s) = gain * ------------------------------------------------------
         *               s^2 + s(2*pi*flow + 2*pi*fhigh) + 2*pi*flow * 2*pi*fhigh
         *
         */
#elif EXPLORATION_USECASE || MINIMAL_INTEGRATION_USECASE
        current_gain = gain_prop;
        current_flow = flow_prop;
        current_fhigh = fhigh_prop;
#endif
#endif
        double omegalow = 2 * M_PI * current_flow;
        double omegahigh = 2 * M_PI * current_fhigh;
        num(0) = 0.0;
        num(1) = current_gain * (omegahigh - omegalow);
        den(0) = omegalow * omegahigh;
        den(1) = omegalow + omegahigh;
        den(2) = 1.0;

        /* compute and write output */
        double out = ltf(num, den, signal_in_port.read());
```


Simulation inputs

- 1) Signal generation
  - C++ functions that implement ISO standards

```
/*******************************
 * RX signal creation functions *
 *******************************/

/* Write ASK100, Modified Miller encoded RF signal */
void globalfuncs_write_ASK_MM_signal(
        sc_out<double> &    outputport,
        const unsigned &    bps,
        const std::string & bitstring,
        const double &      amp,
        const double &      amp_offset,
        const double &      mod_index);

/* Write ASK, Manchester encoded RF signal */
void globalfuncs_write_ASK_MAN_signal(
        sc_out<double> &    outputport,
        const unsigned &    bps,
        const std::string & bitstring,
        const double &      amp,
        const double &      amp_offset,
        const double &      mod_index);
```


Simulation inputs

- 2) Lab trace
  - Sampled NFC signal

```
/************************
 * Write data from file *
 ************************/

/* Write data from file to port */
void globalfuncs_write_data_from_file(
        const std::string & filename,
        sc_out<double> &    outputport,
        const sc_time &     sampleTime);
```

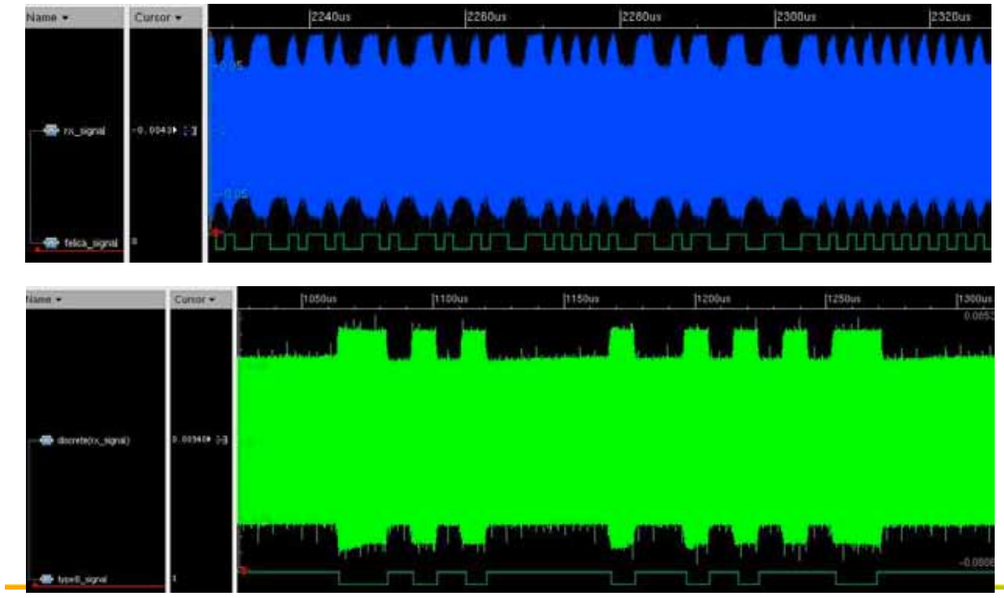
```
globalfuncs_write_data_from_file(tracefilepathstring +
        "typeB_card_106kbps_1ms_Ts2e-9.txt", RX_port,
        sc_time(2e-9, SC_SEC));
```

```
-6.1821643263101578e-02
-5.9133745729923248e-02
-5.5773873813450336e-02
-5.4429925046861172e-02
-5.8461771346628666e-02
-5.3757950663566589e-02
-4.3006360530853271e-02
-2.6207000948488712e-02
-1.0079615749418736e-02
 6.7197438329458237e-03
 2.2847129032015800e-02
 3.4270693548023701e-02
 4.5022283680737019e-02
 4.5694258064031601e-02
 4.6366232447326183e-02
 4.4350309297442436e-02
 4.9726104363799095e-02
 6.1149668879806995e-02
 6.7869412712752819e-02
 6.8541387096047401e-02
 6.5181515179574490e-02
 6.3837566412985325e-02
 5.7789796963334084e-02
 4.7710181213915348e-02
 3.7630565464496613e-02
 2.6878975331783295e-02
 2.1503180265426636e-02
 1.2767513282597065e-02
 3.3598719164729118e-03
-9.4076413661241531e-03
-2.4191077798604965e-02
-3.8302539847791195e-02
-5.5773873813450336e-02
```

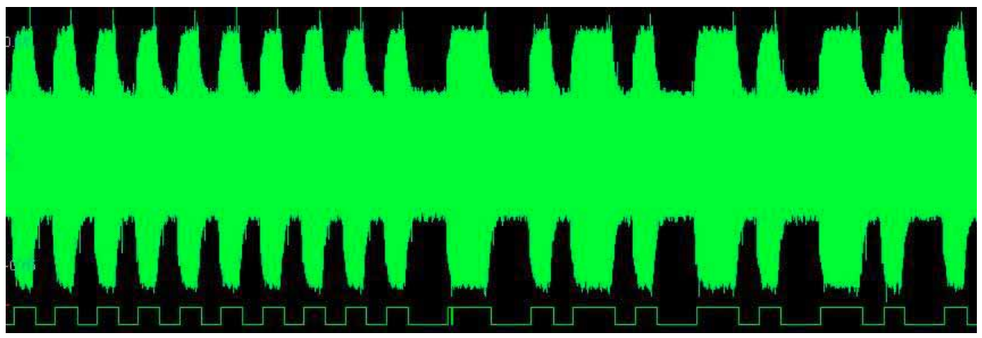

Results

- Simulation speed
  - stand alone lab trace simulation: **23 sec**
    - 10 ms
    - sample rate 2 ns
    - 5,000,000 samples
  - digital VPE simulation with analog stub: **2:10 minutes**
  - digital VPE simulation with analog front-end model: **7:59 minutes**
    - ~3.7x simulation speed decrease
    - full chip functional verification, software development/validation

Results



Results



SystemC AMS Modelling of a Metallic Line Testing System

May 2011

Gerhard Nössing

- Working since 1997 for the Design Center Villach
  - Siemens => Infineon => Lantiq
  - Concept Engineer for POTS codecs
  - Concept Engineer for ADSL CO AFE
  - System Architect for POTS System
  - System Architect for IVD Systems
  - System Architect for MELT
- Main Topics regarding Modeling and Simulation
  - System Modeling in Matlab, Simulink, COSSAP (before 2001) and SystemC
  - Since 2000 we were working together with Fraunhofer Institute for Integrated Circuits in Dresden on the development of SystemC-AMS (MEDEA Anastasia Project)
  - OSCI SystemC-AMS working group (Infineon)
  - System Simulation Expert Group within Lantiq

Copyright © Lantiq 2010. All rights reserved.

Overview

- Who am I
- What is MELT?
- Electrical Network to be measured
- SystemC AMS Model of MELT
- Proposals for Simulation Speed up

Copyright © Lantiq 2010. All rights reserved.

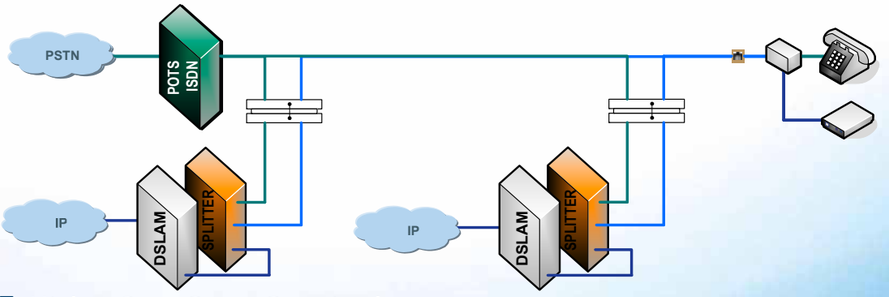
POTS Plain Old Telephone Service

- BORSCHT
  - **B**attery Feed
  - **O**vervoltage Protection
  - **R**inging
  - **S**ignaling
  - **C**oding
  - **H**ybrid
  - **T**esting

Copyright © Lantiq 2010. All rights reserved.

Adding Digital Services

- POTS in parallel with DSL
- Connection of DSL via a Splitter at CO and at CPE side
- Testing is still possible via the BORSCHT function of POTS
- Parallel Network Infrastructure:
  - IP Network (Internet Protocol)
  - PSTN Public Switched Telephone Network

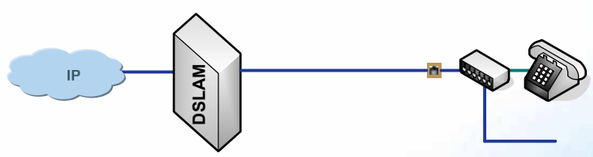


- Higher Data Rates Required
- Move DSLAM close to customer (street cabinet)

Copyright © Lantiq 2010. All rights reserved.

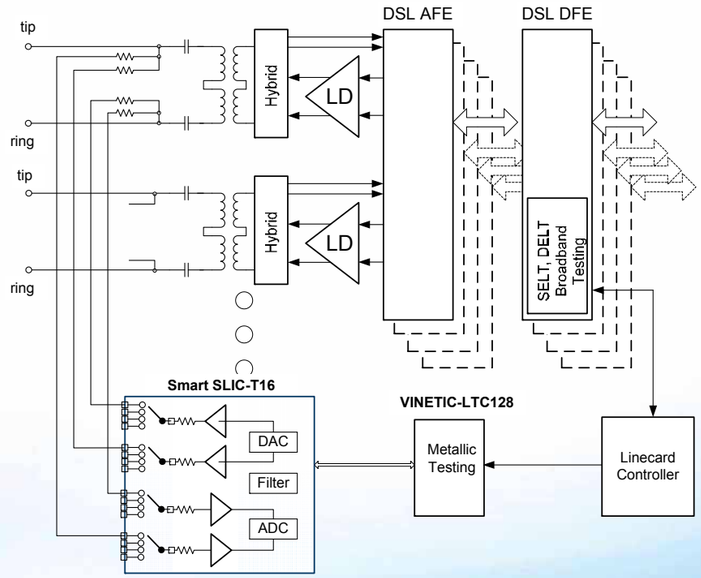
Next Generation Network: All Digital Loop (ADL)

- Only the IP network will remain
- Voice is terminated at CPE side
  - ATA Analog Telephony Adaptor
  - IAD Integrated Access Device
- Problem: No more BORSCHT (=Testing) from POTS



Copyright © Lantiq 2010. All rights reserved.

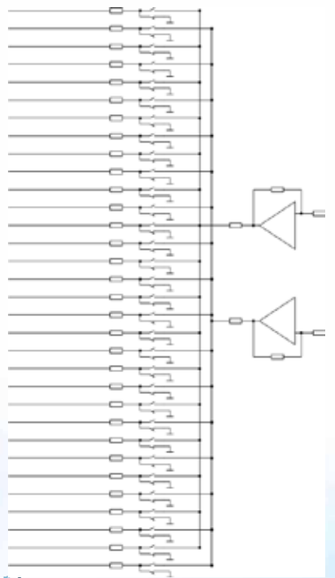
Lantiq MELT Solution



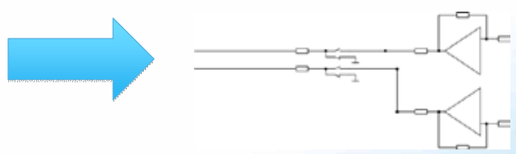
Copyright © Lantiq 2010. All rights reserved.

Simplify Structure for SystemC AMS

- Build up Model with only one channel

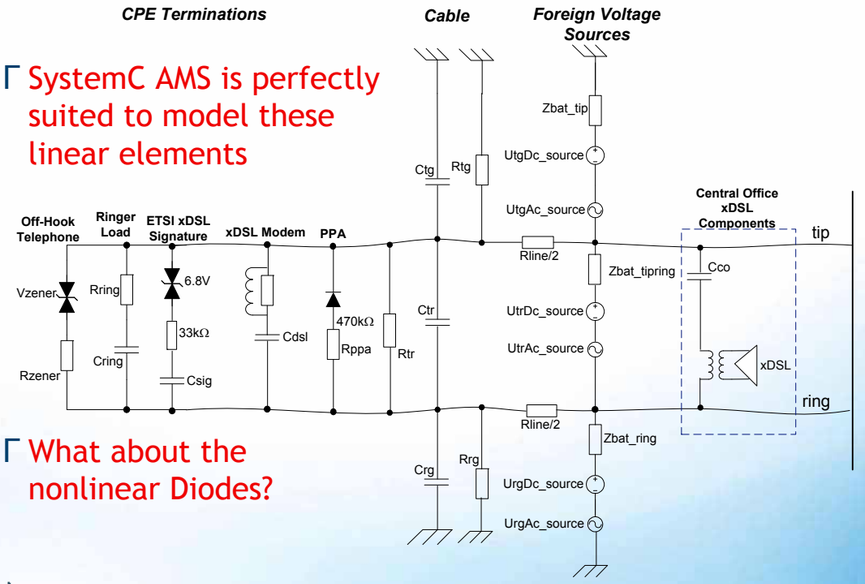


- We do not loose information because with the multiplexer we anyhow measure only one channel at the time



Copyright © Lantiq 2010. All rights reserved.

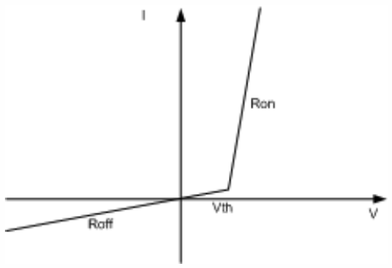
Metallic Line Testing: Network Equivalent Model

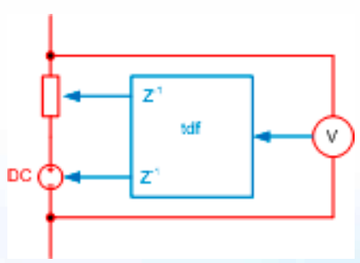


- What about the nonlinear Diodes?

Copyright © Lantiq 2010. All rights reserved.

How to model a Diode

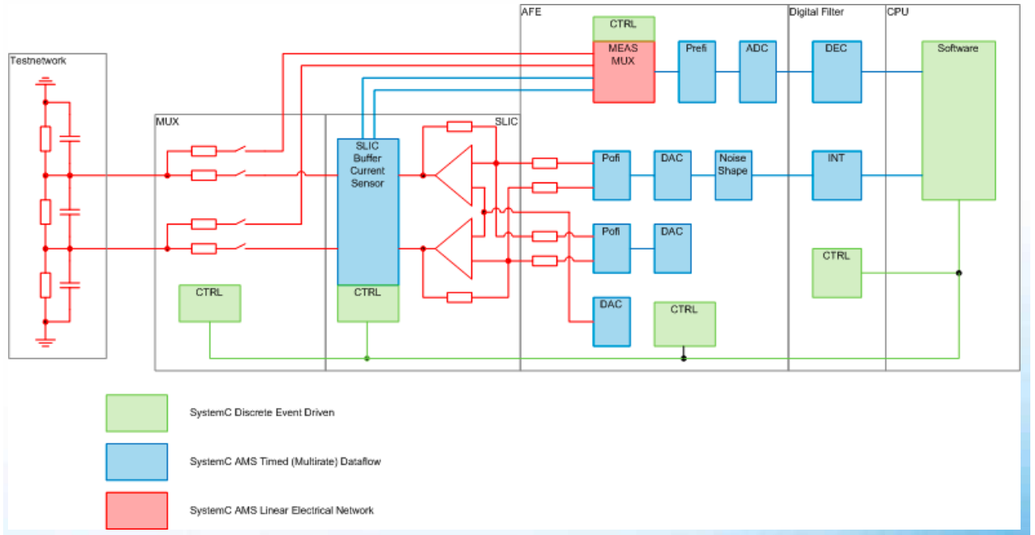




- Start with a resistor
- Measure the voltage
- Convert Voltage to TDF
- Change resistor for next run
- Change voltage for next run

Copyright © Lantiq 2010. All rights reserved.

SystemC AMS MELT Model

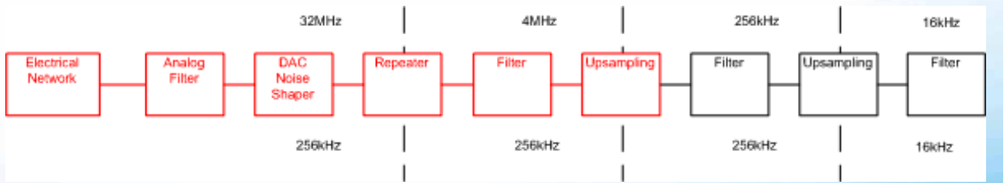


Copyright © Lantiq 2010. All rights reserved.

Reduce Oversampling if possible

- DAC, ADC and the electrical network is running on 32MHz
- Test signals are only up to 3 kHz
- Minimum required oversampling is 10

- Some simulations require the high oversampling
  - Check influence of ADSL signal (2 MHz) to MELT measurement
  - Check influence of MELT out of band noise into DSL band



- Reduced oversampling for Standard MELT Tests

Copyright © Lantiq 2010. All rights reserved.

Conclusion / Proposals to Speed up Simulation

- Reduce Complexity
  - Simplify Structure if possible
- Use Models with different Abstraction
  - E.g. Use a electrical model of the DCDC Converter
  - Or use a behavioral model of the DCDC Converter
- Reduce Oversampling Factor for Simulation
  - Some Simulation require high speed of electrical net
    e.g. out of band noise, DSL disturber test
  - Most of the Simulation do not need high oversampling
- Avoid big electrical nets
  - Sometimes it is better to split the network by adding converter to timed data flow in between

Copyright © Lantiq 2010. All rights reserved.

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Institut für Computertechnik
Institute of Computer Technology

A Range Based System Simulation and Refinement Design Flow

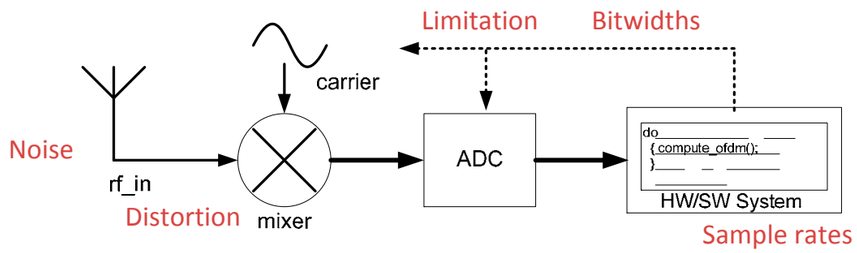
TU Vienna, Chair of Embedded Systems

F. Schupfer, M. Svarc, C. Radojicic, C. Grimm

Embedded Mixed-Signal Systems

Embedding „Cyber“ and „physical“ systems

Application SW Stack



- *Verification of system functionality and parameters (accuracy, power) requires system simulation over long time period*
- *Problems/Errors often detected too late, during/after design*

www.ict.tuwien.ac.at

Overview

- Introduction
- Refinement methodology
- MARC/SYCYPHOS Design environment, examples
- Future work

www.ict.tuwien.ac.at

4 Major Problems in Verification of AMS Systems

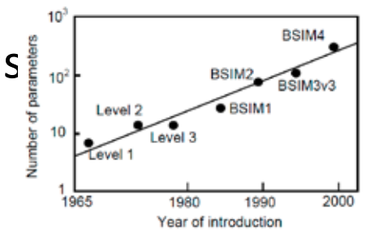


Fig. 2. The increase of MOSFET model parameters

1. Incomplete specification
2. Models too abstract: Power, accuracy, … ?
3. Process variations
4. Insufficient verification coverage, system integration

More accurate models …

More abstract models needed for more simulation runs …

State of the Art, Related Work

Mainstream tries to use RT/circuit level models and simulation

- Mixed-Level, Multi-Run, Monte-Carlo, etc.
- Design of Experiments [Rafaila]
- **Earlier estimation of power + accuracy needed!**

SystemC AMS Methodology enables modeling and simulation of embedded mixed-signal systems at functional, architecture level.

- Power consumption?
- Accuracy?

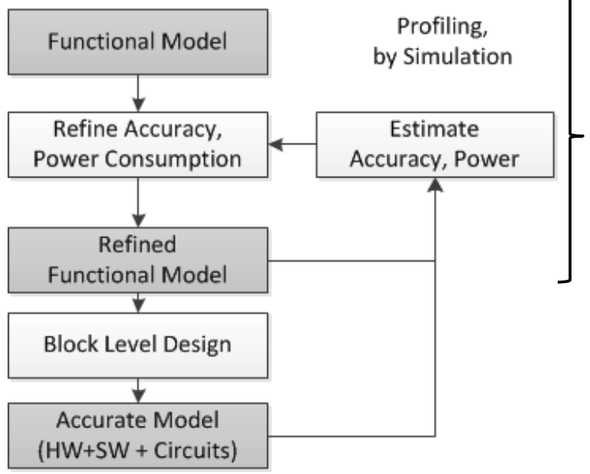
www.ict.tuwien.ac.at

Overview

- Introduction
- Refinement methodology
- MARC/SYCYPHOS Design environment, examples
- Future work

www.ict.tuwien.ac.at

Profiling/Refinement at Functional Level



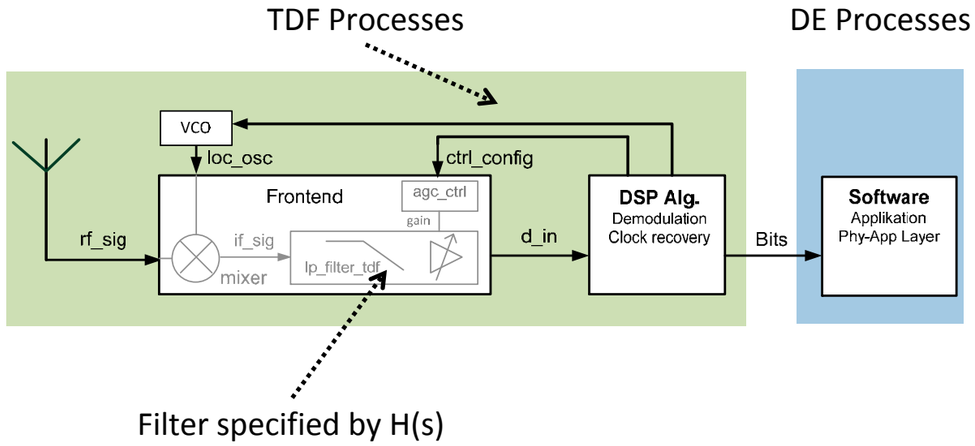
Supported by tools and libraries based on SystemC AMS and TLM extensions:

***Why** is accuracy reduced? Risks?*
=> Accuracy budgeting

***Why** is power consumed?*
=> Power budgeting

www.ict.tuwien.ac.at

Start: Functional Model



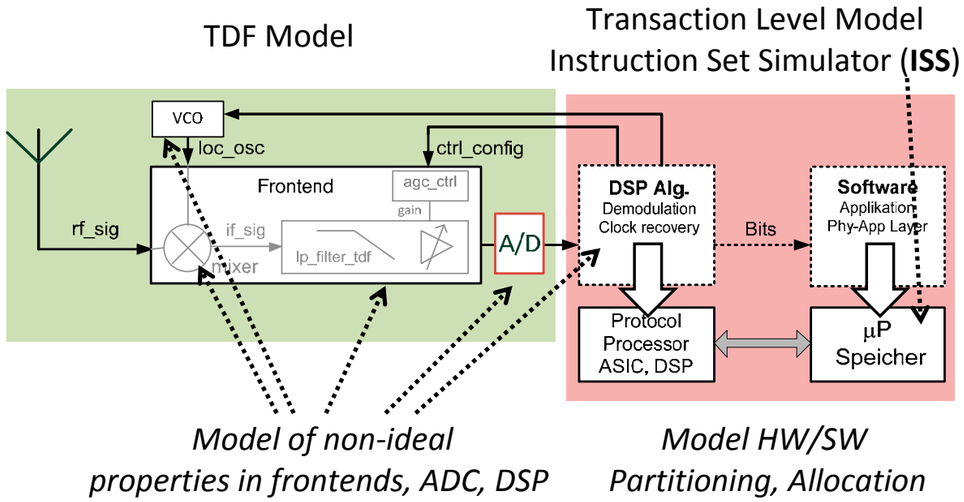
TDF: Filter in Receiver


```
SCA_TDF_MODULE(lp_filter_tdf)
{
  sca_tdf::sca_in<double>  in;
  sca_tdf::sca_out<double> out;
  sca_tdf::sca_in<double>   gain;
  sca_tdf::sca_ltf_nd ltf;        // computes transfer function
  sca_util::sca_vector<double> num, den; // coefficients

  void initialize()
  {
    num(0) = 1.0;
    den(0) = 1.0; den(1) = 1.0/(2.0*M_PI*1.0e4);
  }
  void processing()
  {
    out.write( ltf(num, den, in.read() * gain.read() ) );
  }
  SCA_CTOR(lp_filter_tdf) {}
};
```


www.ict.tuwien.ac.at

Refined functional model



www.ict.tuwien.ac.at

TDF: Filter in Receiver


```
SCA_TDF_MODULE(lp_filter_tdf)
{
  sca_tdf::sca_in<AAF>  in;
  sca_tdf::sca_out<AAF> out;
  sca_tdf::sca_in<AAF>   gain;
  sca_tdf::sca_ltf_nd ltf;        // computes transfer function
  sca_util::sca_vector<double> num, den; // coefficients

  void initialize()
  {
    num(0) = 1.0;
    den(0) = 1.0; den(1) = 1.0/(2.0*M_PI*1.0e4);
  }
  void processing()
  {
    out.write( ltf(num, den, in.read() * gain.read() ) + noise() );
  }
  SCA_CTOR(lp_filter_tdf) {}
};
```

From budgeting

From characterization

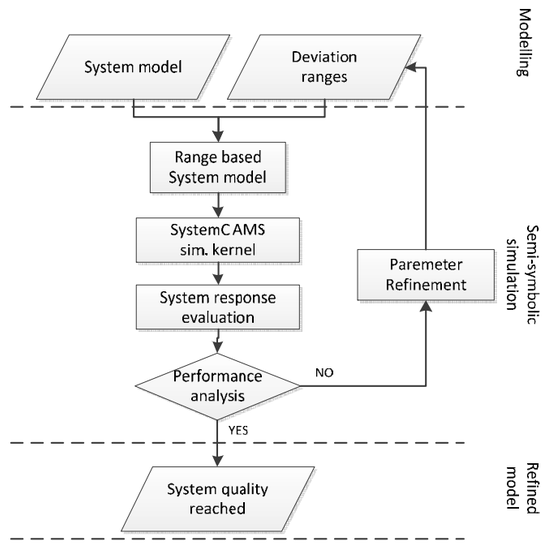
www.ict.tuwien.ac.at

Accuracy Profiling and Iterative Refinement

Range based model

Semi-symbolic simulation with SystemC AMS

Refined system model



SystemC (TLM, AMS) based analysis

- SystemC – based tracing of power and accuracy

„Body“	AAF extension	(or Power ext.)	(or Air ext.)
Sample (or package)	Partial deviations	Accumulated Power	Routing history
			Trace through network

Profiling Accuracy

Power Profiling

Tracing messages through HW and. Network

Range-Based Simulation

Affine Arithmetic – general

- Enhancement of Intervall Arithmetic [Comba]
- Accurate range-based computations for linear systems

Affine Arithmetic - simulation

- Static and dynamic deviations [Heupke, Grimm]
- SystemC AMS integration [Heupke, Grimm]
- Transistor level solver [Grabowski, Grimm]

Affine Arithmetic [Andrade et al.]

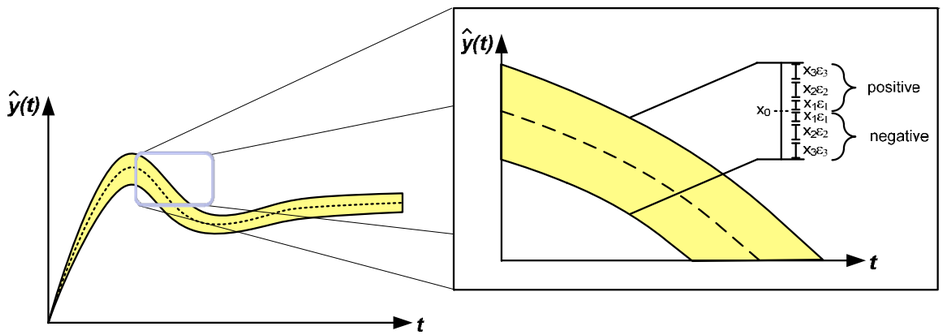
- Improves Interval Arithmetics by conserving *correlations* in a symbolic way

Affine Arithmetics represents a size $\hat{x}$ by

- an ideal, numerical 'central value' x_0, and
- n partial deviations x_i scaled by noise symbols $\epsilon_i \in [-1, 1]$

$$\hat{x} = x_0 + \sum_{i=1}^{n} x_i \epsilon_i$$

Graphical representations

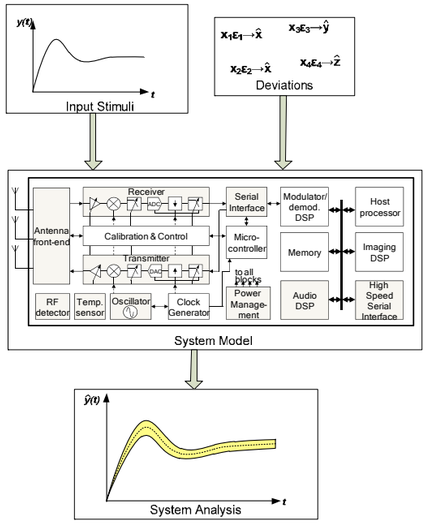


Range based system response

Signal construction by sub-ranges

Semi-symbolic Simulation

- Numeric simulation extended by symbolic representatives
- Multiple simulation results by one run
- Range dependency preservation
- Guaranteed, conservative result inclusion



www.ict.tuwien.ac.at

Overview

- Introduction
- Refinement methodology
- **MARC/SYCYPHOS Design environment, examples**
- Future work

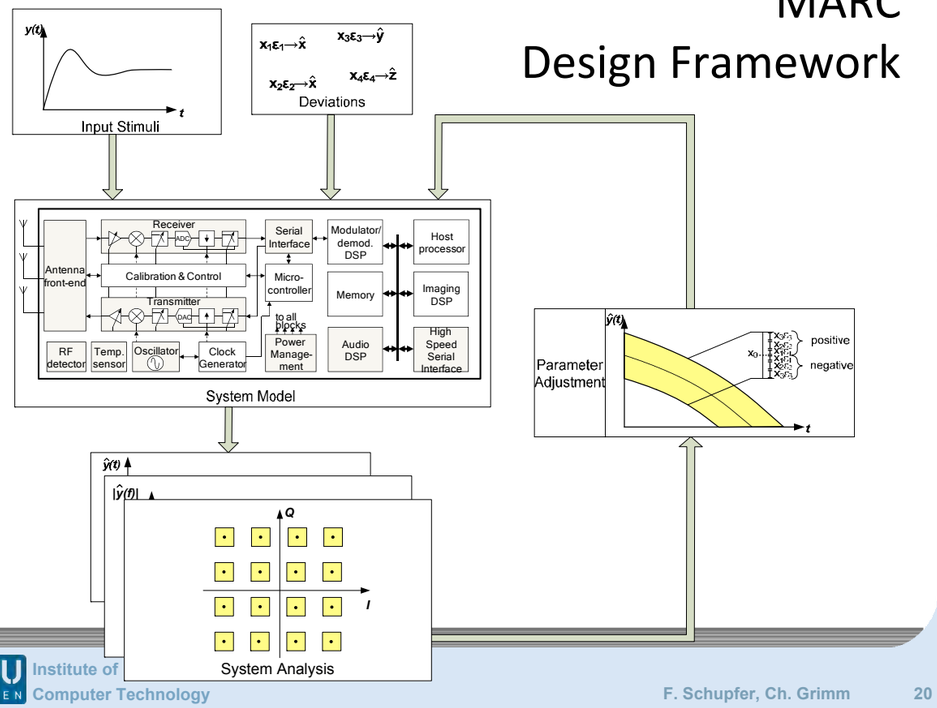
www.ict.tuwien.ac.at

Libraries & Tools

1. Library of functional blocks
   - Blocks for receiver/transmitter (serializer, modulators, mixers, ACD, …)
   - Non-ideal properties (Noise, offset, nonlinearities, …)
   - Models von processors (ISS)
2. Profiling tools
   - **Accuracy profiling**
   - Power (see poster)

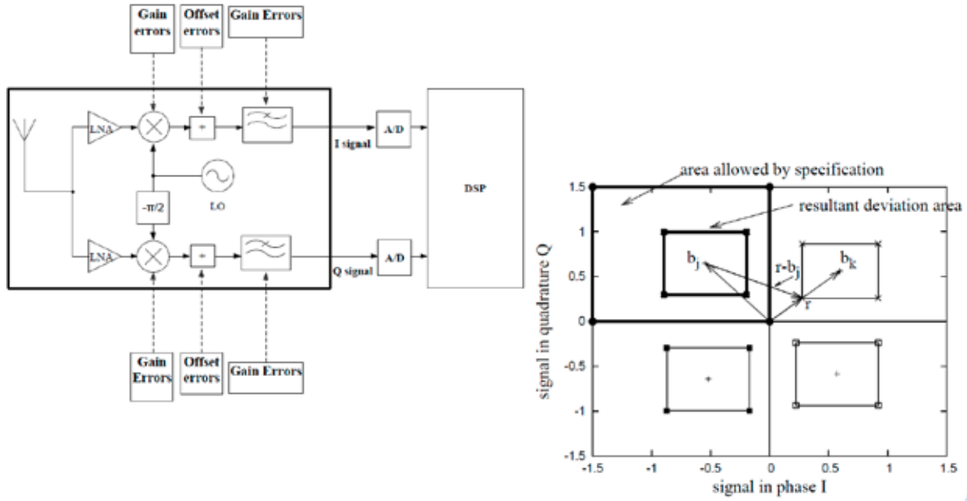
www.ict.tuwien.ac.at

MARC Design Framework



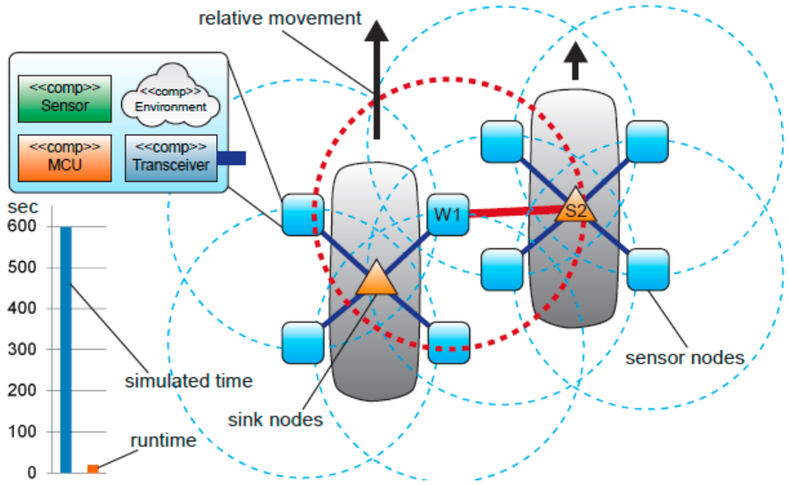
www.ict.tuwien.ac.at

Example: Profiling Accuracy in 4-QAM receiver



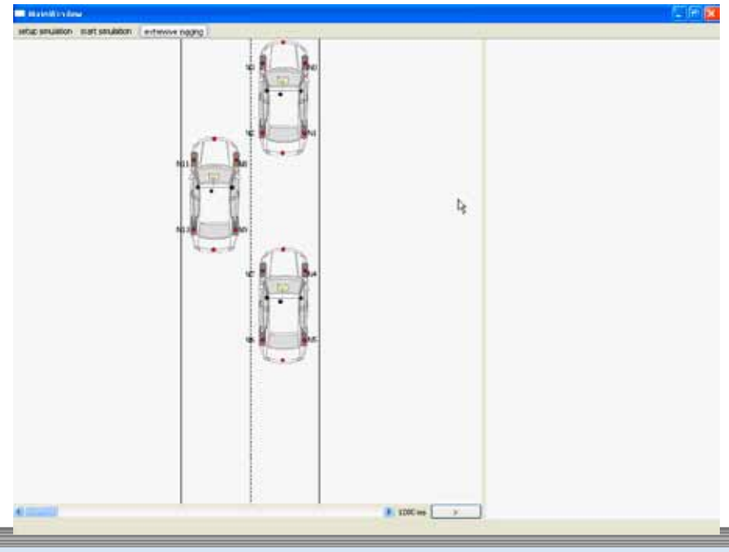
www.ict.tuwien.ac.at

Poster: Power Profiling of In-Car WSN; 18 8-Bit uC with Firmware + Transceiver + Sensors



www.ict.tuwien.ac.at

Poster: Power Profiling of In-Car WSN; 18 8-Bit uC with Firmware + Transceiver + Sensors



www.ict.tuwien.ac.at

Overview

- Introduction
- Refinement methodology
- MARC/SYCYPHOS Design environment, examples
- **Conclusion, Future work**

www.ict.tuwien.ac.at

Conclusion, Outlook

- Range based refinement methodology
  - Complements Worst-Case Analysis
  - Single run, traceable deviations influence
  - Refinement information = recommendations, maybe automation?
- Planned extensions
  - Automated management of ressource “accuracy”
  - “Expert-models” that include typical risks as kind of IP-Knowledge from recent projects

www.ict.tuwien.ac.at

Future work: SYCYPHOS/MARC

- Synthesis of Cyber Physical Systems and Applications *integrates* all TUV Tools:
  - Modeling of scenarios and high-level communication in cyber and physical worlds
  - Modeling of accuracy, robustness, power consumption in microelectronic systems
  - ***Challenge:***
    Automatical analysis, verification, and improvement of accuracy, resilience/adaptivity, power consumption

www.ict.tuwien.ac.at

Thank you for your attention

www.ict.tuwien.ac.at

Affine Arithmetic: System Simulation

System Simulation, SystemC AMS

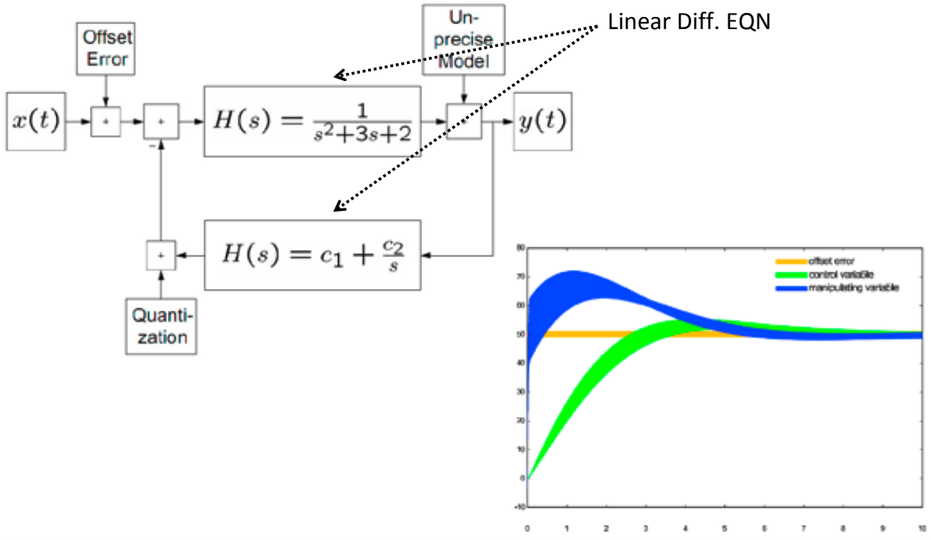
- Directed signal flow; *output = f(input, state)*
- Models of Computation: Synchronous & Dynamic Data flow, KPN, Discrete event modeling, Signal flow

System Simulation with AA straight forward:

- Class library provides abstract data type AAF and associated linear and nonlinear operations
- Number of noise terms increases with each nonlinear operations → „Garbage collection“

www.ict.tuwien.ac.at

Affine Arithmetic: System Simulation



Circuit Simulation with Affine Arithmetic

Component models | Netlist | Parameters p

Modified Nodal Analysis (MNA)

$$\underline{F}(\underline{x}, \dot{\underline{x}}, \underline{p}, t) = \underline{0}$$

Numerical Integration

$$LTE = \left| \frac{\delta t}{2} \ddot{x} \right|$$

$$\underline{F}(\underline{x}, \underline{p}, t) = \underline{0}$$

Symbolic Differentiation

$$\mathbf{J} = \frac{\partial \underline{F}(\underline{x}, \underline{p}, t)}{\partial \underline{x}} \qquad \mathbf{P} = \frac{\partial \underline{F}(\underline{x}, \underline{p}, t)}{\partial \underline{p}}$$

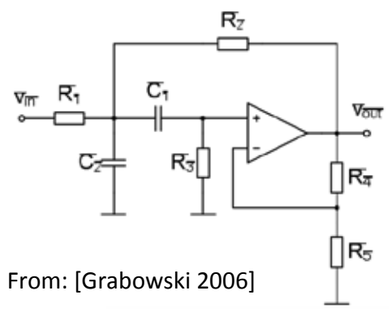
Computation of Affine ASPs

Computaton of Affine ASP as follows:

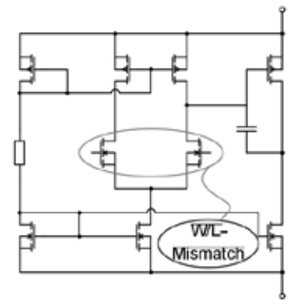
1. Compute x_0 by existing Newton-Raphson iteration:
   $$\underline{F}(\underline{x}_0, \underline{p}_0, t) = \underline{0} \quad \rightarrow \quad \hat{\underline{x}} = \underline{x}_0$$
2. Compute $x_i \epsilon_i$ by sensitivity analysis:
   $$\mathbf{J}|_{x_0,p_0} \Delta \underline{x} + \mathbf{P}|_{x_0,p_0} \Delta \underline{p} = \underline{0} \quad \rightarrow \quad \hat{\underline{x}} = \underline{x}_0 + \sum \underline{x}_p \epsilon_p$$
3. Compute NL ϵ_{i+1} (in n-dim space) by approximation scheme in vector/matrix form (Grabowski 2006, 2007, 2008).

$$\hat{\underline{x}} = \underline{x}_0 + \sum \underline{x}_p \epsilon_p + \underline{x}_{epd} \epsilon_{epd,i}$$

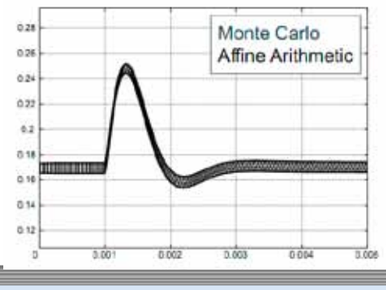
Affine Arithmetic, Circuit Simulation

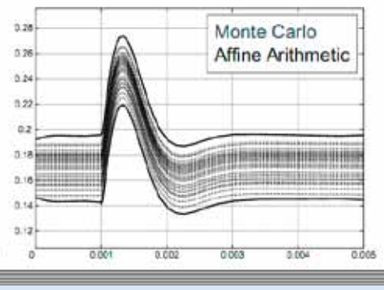


From: [Grabowski 2006]



AAF:
5 sec.
50 MC runs:
50 sec.





Monolithic Three Phase Grid-Tie DC-AC Inverter

Amal Banerjee - 1, 2
Balmiki Sur - 3
Jim Freeman - 1
Andreas Gerstlauer - 2

1 : Analog Electronics
2 : University of Texas at Austin
3 : IC Manage, Los Gatos, California

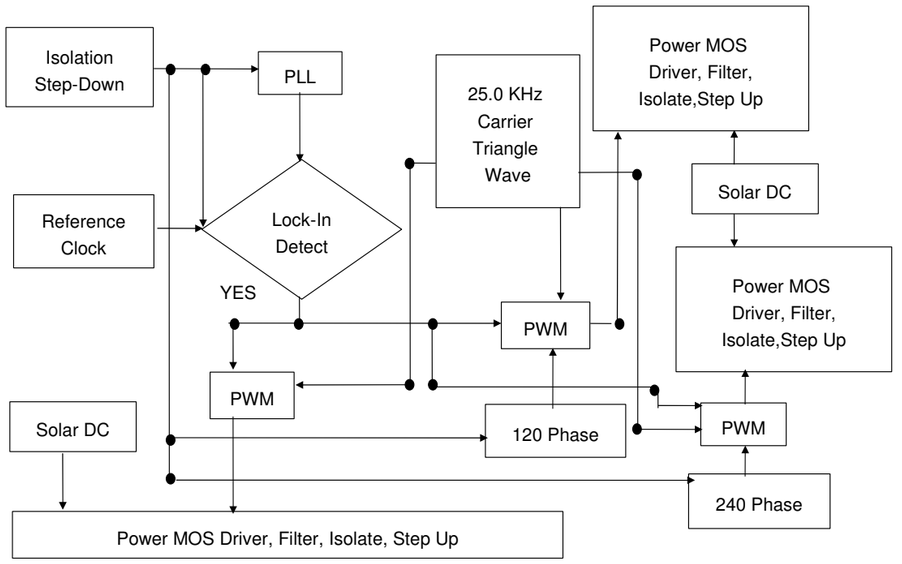
Outline

- Problem Statement
- Inverter Design Overview
- Inverter Components
- SystemC-AMS/SPICE Simulation Results
- Conclusion

Problem Statement

- Need efficient, simple (“plug-n-play”), portable device for solar photo-voltaic panel DC to AC conversion
- Extra AC output injected into local utility power grid - “Grid-Tie”. Inverter output must start/stop with power utility grid output – ensure safety
- Accurate power injection into utility grid depends on strict phase match between local inverter and external utility grid output waveforms

Inverter Design Overview



Inverter Design Overview – Cont'd

- Phase locked loop (PLL) synchronizes with *one* of 3 phases of 3 phase utility grid power supply – “reference grid input”
- Phase lock detector continuously monitors phase lock condition – output *true* / *false* for lock / no-lock states
- Pulse width modulation (PWM) circuitry is active *only when* phase lock detector output is *true*

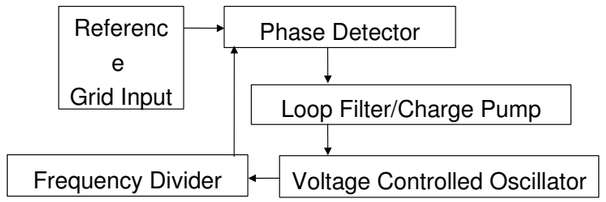
Inverter Design Overview – Cont'd

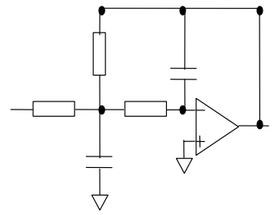
- In parallel, reference grid input is fed into into its own PWM sub-circuit, phase delayed 120 and 240 degrees and each delayed phase fed into its own PWM sub-circuit
- Each PWM sub-circuit drives its own external power MOSFET drivers
- Each PWM sub-circuit is activated *only* in phase lock state

Inverter Design Overview – PLL Lock-In Detector

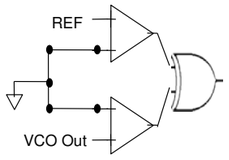
- Count fixed number of clock ticks of 25.0 KHz reference clock to get fixed time interval
- Count number of zero-crossings of both reference grid input and PLL VCO output in fixed time period
- Equal number of zero-cross counts indicates phase lock

Inverter Subcomponents – Phase Locked Loop





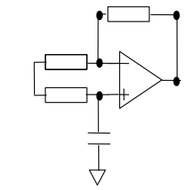
Loop Filter :
sca_eln::sca_nullor,
sca_eln::sca_c
sca_eln::sca_r



Phase Detector
Comparator :
sca_eln::sca_nullor
XOR gate :
SCA_TDF_MODULE

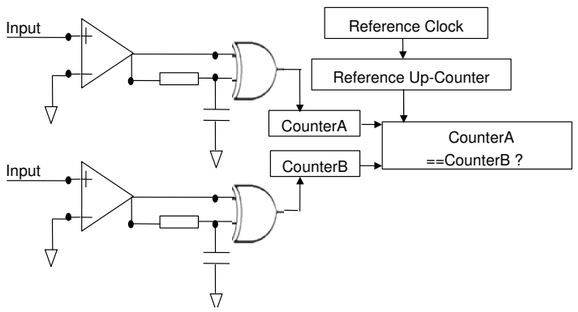
Voltage Controlled Oscillator, Frequency Divider : SCA_TDF_MODULE

Inverter Subcomponents – Phase Delay, Phase Lock Detector, Pulse Width Modulation



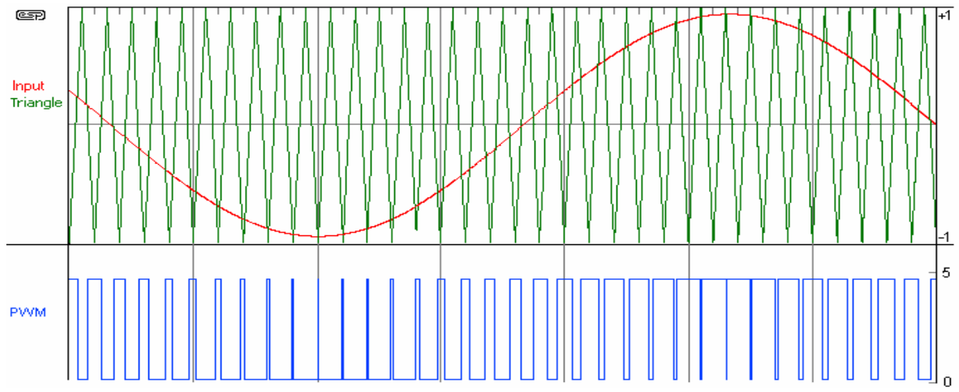
Phase Delay – All Pass Filter
sca_eln::sca_nullor, sca_eln::sca_c, sca_eln::sca_r

Pulse Width Modulation SCA_TDF_MODULE



Lock In Detector -
comparator sca_eln::sca_nullor
edge detector sca_eln::sca_c, sca_eln::sca_r
XOR gate SCA_TDF_MODULE
Counter SCA_TDF_MODULE

Triggered Sine – Triangle Pulse Width Modulation (PWM)



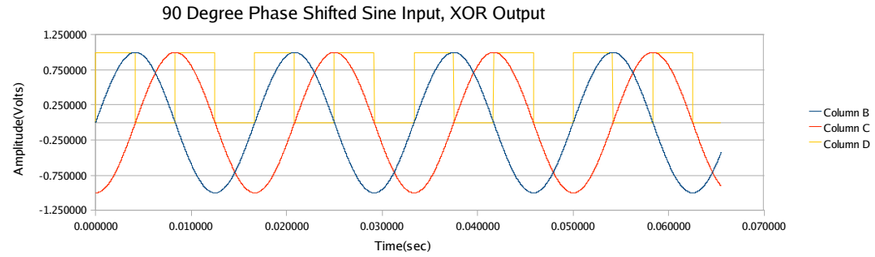
Trigger enables/disables output

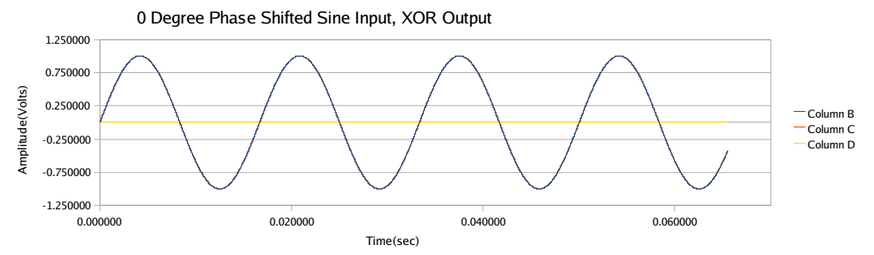
Adjust output for up/down lobes of sine modulation

Simulation Results

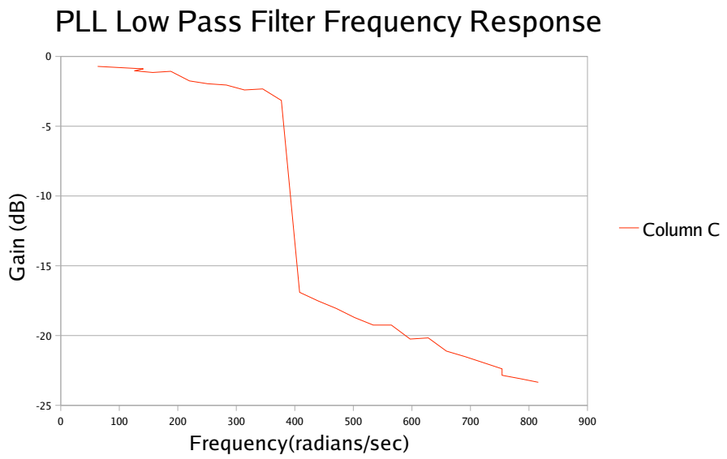
- Phase locked loop controls inverter functionality
- Examine characteristics/output of each PLL sub-component (phase detector, loop filter, voltage controlled oscillator)
- Examine characteristics/output of phase delay generator, pulse width modulation module

PLL Phase Frequency Detector





PLL Low Pass Filter

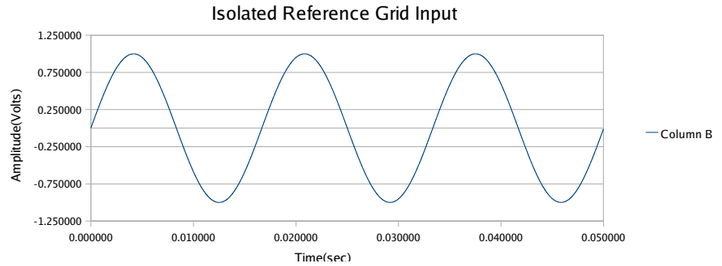


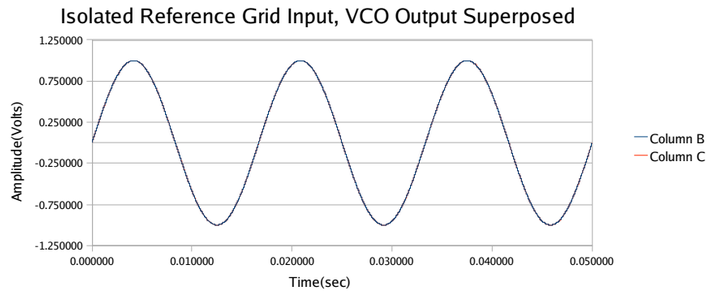
Cut-off frequency : 60.0Hz

PLL – Voltage Controlled Oscillator

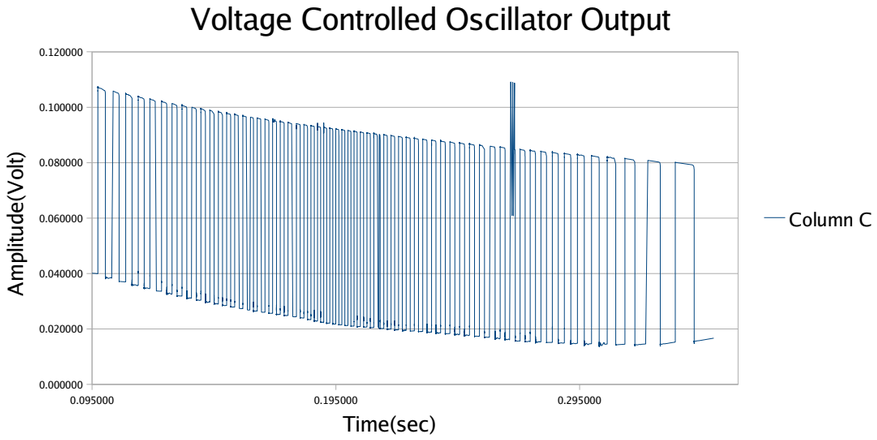
- Implemented in SystemC-AMS and SPICE
- Gain : 1.0, center frequency : 60 Hz, sensitivity : 1.0676 radian/sec/Volt, tuning range : 30 Hz – 90 Hz

PLL Voltage Controlled Oscillator (SystemC-AMS)



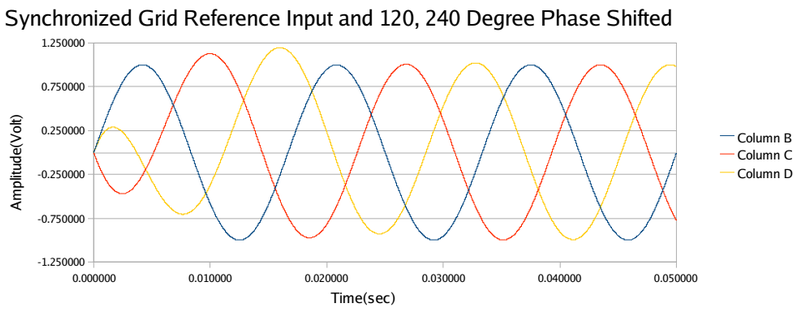


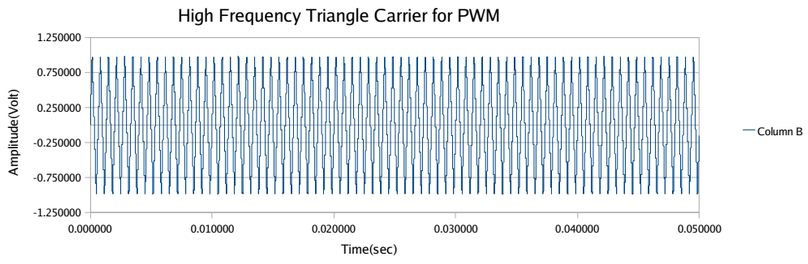
PLL Voltage Controlled Oscillator (SPICE)



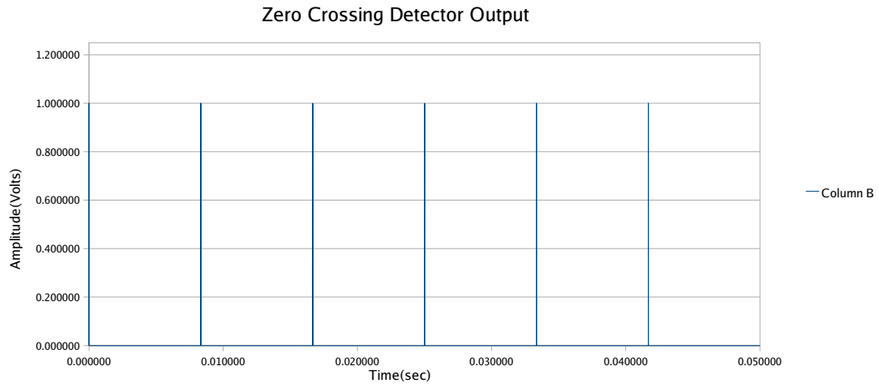
VCO astable multivibrator

Phase Delay - Pulse Width Modulation

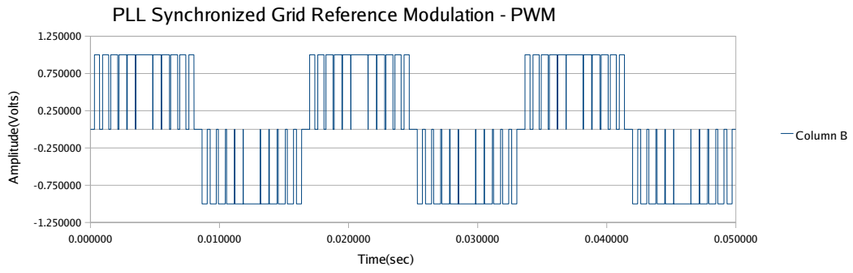


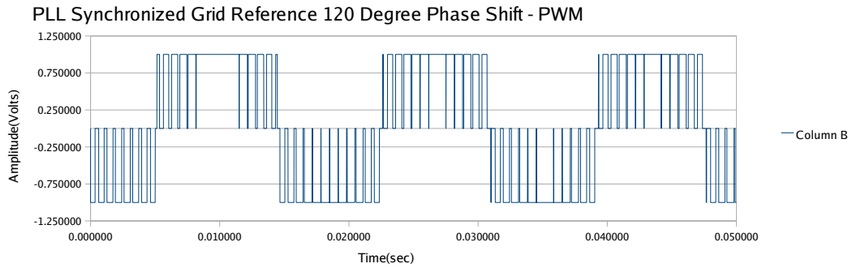


Lock In Detector



Pulse Width Modulation





Conclusion

- 60.0 Hz center frequency phase locked loop/voltage controlled oscillator designed, analyzed with SystemC-AMS and SPICE
- Monolithic 3 phase DC – AC inverter designed with the PLL as the phase synchronizing element, correctly converts DC input to 3 phase grid compatible output
- Currently verifying/improving SPICE results