Virtual Prototyping for
High Performance Mixed Signal Products

Martin Barnasconi
NXP Semiconductors
Design, Automation and Test in Europe (DATE)
March 22, 2013
Outline

- NXP Semiconductors
- Trends in embedded mixed-signal systems
- Why Virtual Prototyping?
- Automotive example: In-vehicle networking
- Virtual Prototyping Environment
- Verification Environment
- Results
- Outlook
- EDA standardization needs
- Conclusions

- **Headquarters:** Eindhoven, The Netherlands
- **Employee base:**
  - approximately 25,000 employees
  - in more than 25 countries
  - R&D in Europe, US, and Asia
  - Manufacturing in Asia and Europe: 85% out of own production
- **Revenue:** $4.36 billion in 2012
- **Customers:** Leading OEMs worldwide
NXP Semiconductors - Product applications

Key Mega-Trends

Energy Efficiency  Connected Devices  Security  Health

High-Performance Mixed-Signal Solutions
Application-optimized analog and digital solutions that help our customers to truly differentiate their products in terms of features, cost and time to market to address

Highest Growth Segments in 8 Priority Application Areas

- Automotive
- Identification
- Wireless Infrastructure
- Lighting
- Industrial
- Mobile
- Consumer
- Computing
NXP’s embedded mixed-signal systems

Automotive
- In-vehicle networking
- Car access & immobilizers
- Car entertainment
- Solid State Lighting
- Telematics
- Speed & Angular Sensors

Identification
- Secure identity
- Secure transactions
- Tagging & authentication

Consumer
- TV
- Satellite, Cable, Terrestrial and IP set-top boxes
- Satellite outdoor units

Computing
- Tablet PCs
- Note- /Netbooks
- Desktops
- Power supplies
- Monitors and peripherals
Trends in embedded mixed-signal systems

- Higher complexity of AMS ICs and systems
  - Integrated microcontrollers (and thus firmware)
  - Interaction with environment (e.g. sensors, wireless communication)
  - Smarter system partitioning and analog-digital interaction to comply with low power and die size constraints

- Compliancy to functional safety and security standards
  - Automotive requires ISO26262 compliancy
  - Identification business requires security standard (ISO, ITU-T)
  - Guaranteed reliability over the entire lifespan of the product

- More focus of fail-safe operation and zero-defects
  - Requires more emphasis on system verification
  - Challenge: Verification of use cases we initially did not think of
Why Virtual Prototyping?

- Abstract system model to validate specifications, algorithms, and/or embedded firmware
  - Simulation model offers advanced tracing and debugging capabilities to verify fault-scenarios and robustness of the system
  - Verify correct interaction between analog and digital signals and states

- Virtual prototype used as reference model to develop and validate use cases and verify the actual IC implementation
  - Ability to include AMS, digital (RTL) or software/firmware descriptions

- Application-specific verification environment can be (re)defined easily
  - Testbenches for multiple applications and use cases

- Full-chip system model delivery to Tier1 or OEM as executable specification for application or SW development
Automotive example: In-Vehicle Networking
Example 1: Body Controller Module

**Input signals**
(sensors, switches, analog signals)

**Output signals**
(lightning, motors, power supplies)

- **System Basis Chip**
  - UJA107x
  - HS-CAN
  - LIN
  - SPI
  - Watchdog

- **In-Vehicle Network**
  - LIN, CAN, FlexRay

- **In-Vehicle Network Components**
  - ESD prot. PESD1LIN
  - ESD prot. PESDxCAN
  - ESD prot. PESD1Flex

- **HS-CAN transceiver**
  - TJA1051
  - TJA1042
  - TJA1043

- **FT-CAN transceiver**
  - TJA1055

- **LIN transceiver**
  - TJA1021

- **FlexRay transceiver**
  - TJA108x

- **Microcontroller**

- **Broad automotive portfolio**
  - Small-signal MOSFETs
    - 2N2007x
  - MEGA schottky diodes
    - PMEGx
  - Power products
    - Innovative LFPAK packaging

- **Current source**
  - PSSI2021SAY

- **Light control**
  - TDA3629

- **Shunt regulator**
  - TL431

- **Transistor**
  - BISS transistors PBBx

- **Switches**
  - RETs

- **Power products**
  - Innovative LFPAK packaging

- **Standard Logic**
  - 74xxx
Example 2: Front Door ECU
NXP’s System Basis Chip (SBC)

- Complex mixed-signal IC including analog and digital functionality
  - Delivers regulated supply voltage to all external peripherals on the ECU
  - All system states are controlled and monitored by the system controller
  - Communicates via CAN, LIN or Flexray network

- Requirement: Safe and predictable behaviour under all conditions

- Virtual prototype and verification environment needed to verify all possible conditions and use cases
System verification aspects

- Checking temperature conditions
- Checking battery conditions
- Checking undervoltage conditions
Virtual prototyping and verification in the design cycle

Product Concept & Function → Architecture & Implementation → Application & Design-In

Product Specification → System Ref. Simulation

Use cases → Models

System Verification

Application Know-how via Use Cases

System Maturing via Results

Application Egr → Design Egr → System Egr
Virtual Prototype: Mixed-Lang. Environment

Sequencer → Use Cases -----> Rules ----> Checker

- CAN
- LIN
- Dig In
- Ana In
- Ana In (Ramp)
- Registers

DUT System Basis Chip

- Comp
- Ana Ext R, L, C

μController Software

- SystemC/AMS
- C++
- XML
- C

Dig Out

AnaOut

Registers

Detectors
Verification Environment (1)

- Test specification independent from testbench implementation
- Testbench environment independent from DUT implementation
- Concept development phase
  - C++/SystemC/SystemC-AMS simulation environment
  - Microsoft Windows platform
Verification Environment (2)

- Test specification independent from testbench implementation
- Testbench environment independent from DUT implementation

- Concept development phase
  - C++/SystemC/SystemC-AMS simulation environment
  - Microsoft Windows platform

- IC verification phase
  - Including Verilog-AMS + Schematics
  - EDA vendor simulator
  - Linux platform
Use cases and rules

A mistake in the concept could make product **useless** (even if specification is met for 100%)
- Clearly specify and verify the chip functionality
- Clearly define what should not happen (“anti-spec”)

- Example of product use cases
  - sequence: “write 0x7 to register SDMC using 32 bits and 250ns transmission time per bit”
  - sequence: “in parallel, wait 5us and set RSTN low”
  - abstract sequence: “run PowerOn sequence”

- Example of checker rule
  - “after PowerOn sequence, VBAT should be $\geq 12V$ and status register bit X should be 0”
  - possibly multiple conditions per rule
Verification Environment – Implementation

- Test specification language options
  - same as testbench implementation language: not independent
  - XML: widely used, no testbench implementation knowledge required

- Testbench implementation language options
  - SystemVerilog / Verilog / VHDL: needs proprietary simulator
  - SystemC: tool / platform agnostic (C++ based)

- Verification methodology options
  - Universal Verification Methodology (UVM):
    • Current UVM implementation only based on SystemVerilog, not SystemC
    • No support for advanced testbench control
  - Property Specification Language (PSL): needs proprietary simulator

- Chosen implementation
  - UVM-like approach for testbench architecture in SystemC
  - Test specification language in XML, interpreted by testbench sequencer
Virtual prototyping and verification results

- Prototype environment used for System Basis Chip concept development and tape-out IC verification
- First customer received early simulation model for SW development
- Revealed specification misinterpretations (Concept vs. Chip level)
- #IC design issues found in time: 50
  - 35 by regular verification approach
  - 15 by XML driven prototype, verification approach (based on 30 XML use cases)
Outlook: UVM-SystemC/AMS

- Standardized API to quickly develop well-constructed and reusable verification environment, components and tests
- Introduce standardized features dedicated to functional verification: phasing, component overriding, coverage, configuration, comparing, scoreboard, reporting, etc.
- SystemC/SystemC-AMS based class extensions compatible with UVM capabilities

*Part of FP7 project VERDI (http://verdi-fp7.eu)
EDA standardization needs

- Better leverage from proven digital-centric verification methods and ‘port’ them to the AMS domain
  - Universal Verification Methodology (UVM) for AMS
  - Assertion-based Verification (ABV) methods for AMS
  - Coverage-based and Metrics Driven AMS Verification

- Standardized AMS language extensions dedicated to mixed-signal system verification
  - AMS extensions and SystemC-based implementation of UVM
  - AMS extensions for SystemC (SystemC AMS)
Conclusions

- **Virtual Prototyping**
  - Shown mixed-signal approach adds to entire dev flow (spec → evaluation)
  - Benefits in the increase of concept robustness and chip design quality
  - Flexible approach allows adaptation of use cases to product variants
  - Good to support customer for early software development

- **Methodology**
  - C++/SystemC enables system, design, and application level simulation
  - Application-specific stimuli (XML) deployed into chip verification & system evaluation
  - EDA tool limited to Verilog-AMS for design integration; limits debug facility

- **Outlook and needs**
  - Leverage from verification methodology concepts like UVM
  - UVM needs extensions towards system-level (SystemC) and AMS (SystemC-AMS)
Acknowledgements

- Bas Arts
- Jens Fleischhacker
- Steffen Müller
Your questions