Product Development Flow Including Model-Based Design and System-Level Functional Verification

Ascension Vizinho-Coutry, avizinho@mathworks.fr
Agenda

- Introduction to Model-Based-Design and Simulink
- System Level Design:
  - Modeling
  - Fixed-Point Analysis
  - link to Implementation
- Verification: Link for Cadence Incisive
- Conclusion
Developing Electronic Systems with Simulink

- Wireless communications
- Video processing
- Speech and audio processing
- Many others
Executable models
- Unambiguous
- Only “one truth”

Simulation
- Reduces “real” prototypes
- Systematic “what-if” analysis

Automatic code generation
- Minimizes coding errors

Generated code
- Embedded Software
  - C, RTOS, IDE
- Electronic Hardware
  - HDL, Synthesis, Netlist

Generate
- MCU
- DSP
- FPGA
- ASIC

Test and Verification

Requirements and Specifications

System Behavior

Design

Subsystem Behavior

Implementation

Component Behavior

Embedded Software

Electronic Hardware
Requirements

Environment

System Behavior

Subsystem Behavior

Subsystem Behavior

Component Behavior

Component Behavior

Assertions, Tests, etc.

Assertions, Tests, etc.

Assertions, Tests, etc.

Assertions, Tests, etc.

Assertions, Tests, etc.

Assertions, Tests, etc.

Assertions, Tests, etc.

Automation and Test Systems (T&M, HILS)

C, RTOS, IDE

MCU

DSP

FPGA

ASIC

HDL, Synthesis, Netlist

Generate

Embedded Software

Electronic Hardware

Test with Design

- Detects errors earlier

Model Verification and Validation

Test and Verification

Co-simulation
How to Get Started in Simulink®?

- Graphically drag-and-drop blocks from libraries and connect them to build models
Multi-Domain, Multirate Modeling

- Simulink® handles multirate systems easily
  - Time is built-in to Simulink
  - Simulink engine takes care of updating states and outputs of every block at every time step
  - User focuses on algorithms during system development

- Efficient solvers ➔ fast simulations
  - *Variable Step Solvers* – allows for modeling systems with various rates including analog signals
  - *Fixed Step Solvers* – useful for modeling discrete systems with single or multiple rates
Multirate, Multi-Domain System

- Analog to Digital Data-Converter

Multiple rates in the system show up in different colors

*Format → Port Signal Displays → Sample Time Colors*
Modeling and Implementation

**Modeling**

- **Communication**
  - Communications Blockset
  - Communications Toolbox
  - RF Blockset
  - RF Toolbox

- **Filter Design**
  - Filter Design Toolbox
  - Wavelet Toolbox

- **Video and Imaging**
  - Video and Image Processing Blockset
  - Image Processing Toolbox
  - Image Acquisition Toolbox

- **Others**
  - Embedded Matlab, S-Function

**Implementation**

- **Verification**
  - Link for Code Composer Studio™
  - Link for Modelsim
  - Link for Cadence Incisive
  - Cadence AMS Designer

- **Code Generation**
  - Real-Time Workshop® Embedded Coder
  - Embedded Target for DSPs
  - Filter Design HDL Coder
  - Simulink HDL Coder
  - Xilinx System Generator™
  - Altera DSP Builder
Link for Cadence Incisive Brings Together Leading Tools for Modeling and HDL Simulation

- Link for Cadence Incisive is a fast, bidirectional cosimulation interface for system-level functional verification.

- Benefit: Reuse test environment in the executable specification to verify the implementation
Use Case 1: Product Development Using Simulink and Incisive

- Data Source: Bernoulli Binary
  - Samples per Frame = 172
  - Sample Time = 20e-3/172

- Encoding: Channel Bits, Modulation Symbols

- Interfaces

- Transmitter
  - Modulation Symbols
  - ToChannel

- BER Results: Raw BER
  - BER Error Rate Calculation
  - Received Delay = 2T

- Receiver
  - PC SubChannel
  - FromChannel

- Decoding: Channel Bits, Modulation Symbols

- BER Results: Channel Bits (encoded)
  - BER Error Rate Calculation
  - Received Delay = 2T

- Sample-based and frame-based signals
- Floating-point and fixed-point data types
- System metrics (BER, PER, S/N)
Algorithm Development Using MATLAB and Incisive

Use Case 2: MATLAB Test Bench
- Top-level entity is M-code
- M-code is a test bench

Use Case 3: MATLAB Component
- Top-level entity is HDL
- M-code is an untimed placeholder for a “to-be-coded-in-HDL” algorithm
Demo: Edge Detection in an Image
Link for Cadence Incisive 2

- Verify and cosimulate HDL designs using MATLAB and Simulink
- Released in R2006b+, now available on DVD
- New in R2007a
  - Native VHDL language support
    - Supports both major hardware description languages: Verilog and VHDL
    - Mixed language support (Verilog and VHDL) in same Simulink model
    - Option to deactivate HDL cosimulation block for faster Simulink model debugging

- Interfaces and platforms:
  - Shared memory
    - MathWorks and Cadence tools on LINUX or Solaris
  - TCP/IP Sockets
    - MathWorks tools on Windows, LINUX, or Solaris
    - Cadence tools on LINUX or Solaris
System Requirements for Link for Cadence Incisive

- Required:
  - MATLAB R2006b, R2007a
  - Incisive simulator 5.5, 5.6, 5.7 or 5.8

- Required to use Link for Incisive with Simulink:
  - Simulink
  - Fixed-Point Toolbox
  - Simulink Fixed Point

- Recommended for improved fixed-point performance in Simulink
  - Simulink Accelerator
Link for Cadence Incisive Brings Together Leading Tools for Modeling and HDL Simulation

- Link for Cadence Incisive is a fast, bidirectional cosimulation interface for system-level functional verification.

- Benefit: Reuse test environment in the executable specification to verify the implementation.
For more Information

Link for Cadence Incisive 2.0

Cosimulate and verify VHDL and Verilog using Incisive simulators

Link for Cadence® Incisive® software is a cosimulation interface that integrates MATLAB and Simulink into the hardware design flow for application-specific integrated circuit (ASIC) and field programmable gate array (FPGA) development. It provides a bidirectional link between MATLAB and Simulink and Incisive® simulators (from Cadence Design Systems). Link for Cadence Incisive software enables you to verify your HDL design from within MATLAB and Simulink. It provides native cosimulation support for both Verilog and VHDL.

- Introduction and Key Features
- Working with Link for Cadence Incisive Software
- Typical Applications
- Using Link for Cadence Incisive Software with MATLAB
Conclusion

- **Link for Cadence Incisive:**
  - RTL verification
  - speed-up simulation
  - possibility of re-using System Testbench

- **Simulink:**
  - Multi Rate/Domain modelling
  - Increase productivity
  - Code Generation
Thank you

Ascension Vizinho-Coutry, avizinho@mathworks.fr
Guido Scheiner, gschrein@mathworks.com