RF Library based on Block Diagram and Behavioral descriptions

Benjamin Nicolle¹,², Jean-José Mayol², William Tatinian¹, Jean Oudinot², Gilles Jacquemod¹

1. Laboratoire d'Electronique Antennes et Télécommunications – UNSA CNRS UMR 6071 – 06560 Valbonne, France
2. Mentor Graphics – ZIRST Montbonnot – 38334 Saint Ismier, France

Contact: benjamin.nicolle@unice.fr
Introduction
- Classical simulation support
- Proposed simulation support
- ... and EDA Softwares?
- One solution with ADVance System Model Extractor (SME)

RF Library presentation
- Noise Figure implementation
- Phase Noise implementation
- RF blocks modeling summary

Transceiver simulation results
- Specifications extraction
- Transceiver results
- Focus on Transmitter

Conclusion
Introduction
- Classical simulation support
- Proposed simulation support
- ... and EDA Softwares ?
- One solution with ADVance SME

RF Library presentation
- Noise Figure implementation
- Phase Noise implementation
- RF blocks modeling summary

Transceiver simulation results
- Specifications extraction
- Transceiver results
- Focus on Transmitter

Conclusion

Mixing description languages to reduce development time, increase model accuracy
Introduction
- Classical simulation support
- Proposed simulation support
- ... and EDA Softwares?
- One solution with ADVance SME

RF Library presentation
- Noise Figure implementation
- Phase Noise implementation
- RF blocks modeling summary

Transceiver simulation results
- Specifications extraction
- Transceiver results
- Focus on Transmitter

Conclusion
Introduction
- Classical simulation support
- Proposed simulation support
- ... and EDA Softwares?
- One solution with ADVance SME

RF Library presentation
- Noise Figure implementation
- Phase Noise implementation
- RF blocks modeling summary

Transceiver simulation results
- Specifications extraction
- Transceiver results
- Focus on Transmitter

Conclusion
Wireless application:

How can I do it?

Environment

Previous Project

Time-to-Market

System Specifications

etc

- Classical simulation support
Introduction

Wireless application:

- How can I do it?
- How can I divide it?

System Specifications

- Environment
- Previous Project
- Time-to-Market

- etc

- Subsystem DSP
- Subsystem RF
- Subsystem BB

Classical simulation support

Wireless application:
Introduction

Wireless application:

- Classical simulation support

System Specifications

- Environment
- Previous Project
- Time-to-Market

Subsystems:
- DSP
- RF
- BB

Blocks:
- Block 1
- Block 2
- Block 3
- ...
- Block N

Programming languages:
- VHDL
- Verilog
- VHDL-AMS, Verilog-AMS
- Simulink

etc
Introduction

Wireless application:

- **System Specifications**
  - Environment
  - Previous Project
  - Time-to-Market
  - etc

- **Subsystem**
  - DSP
  - RF
  - BB

- **Block**
  - 1
  - 2
  - 3
  - ...
  - N

- **Modeling**
  - VHDL
  - Verilog
  - VHDL-AMS, Verilog-AMS
  - Simulink

- **Implementation**
  - SPICE implementation, Transistors Level
**Introduction**

- Classical simulation support

**System Specifications**

- Environment
- Previous Project
- Time-to-Market

- Subsystem DSP
- Subsystem RF
- Subsystem BB

- Block 1
- Block 2
- Block 3
- ... Block N

- VHDL
- VHDL-AMS, Verilog-AMS

- Simulink

- SPICE implementation, Transistors Level
Introduction

- Classical simulation support

System Specifications
- Environment
- Previous Project
- Time-to-Market

Subsystem
- DSP
- RF
- BB

Block 1
- Block 2
- Block 3
- ...
- Block N

- VHDL
- VHDL-AMS, Verilog-AMS
- Simulink

- VHDL-AMS, Verilog-AMS
- Simulink

- VHDL-AMS, Verilog-AMS
- Simulink

- VHDL-AMS, Verilog-AMS
- Simulink

System Level Models
- Matlab / Simulink or VHDL-AMS / Verilog-AMS
- Excel, ADS...

- VHDL, SystemC
- Simulink or VHDL-AMS / Verilog-AMS

- VHDL, SystemC
- Simulink or VHDL-AMS / Verilog-AMS

- VHDL, SystemC
- Simulink or VHDL-AMS / Verilog-AMS

- VHDL, SystemC
- Simulink or VHDL-AMS / Verilog-AMS

SPICE implementation, Transistors Level
Proposed simulation support

Introduction

System Specifications

- Environment
- Previous Project
- Time-to-Market

- Subsystem DSP
- Subsystem RF
- Subsystem BB

- Block 1
- Block 2
- Block 3
- ...
- Block N

- VHDL, Verilog
- VHDL-AMS, Verilog-AMS, Simulink
- VHDL-AMS, Verilog-AMS, Simulink
- VHDL-AMS, Verilog-AMS, Simulink, ...
- VHDL-AMS, Verilog-AMS, Simulink, ...

SPICE implementation, Transistors Level
Introduction

Proposed simulation support

Environment

Previous Project

Time-to-Market

System Specifications

Subsystem DSP

Subsystem RF

Subsystem BB

Block 1

Block 2

Block 3

... 

Block N

VHDL

Verilog

VHDL-AMS, Verilog-AMS, Simulink

VHDL-AMS, Verilog-AMS, Simulink

VHDL-AMS, Verilog-AMS, Simulink, ...

VHDL-AMS, Verilog-AMS, Simulink, ...

SPICE implementation, Transistors Level
... and EDA Softwares?

- System Simulators: ADS, CoCentric, Matlab, Simulink, SPW, Excel...
- Mixed-signal simulators: ADVance MS, Smash, AMS Designer, Saber...
- Circuit Simulators: Eldo, Spectre, Spice, ADS...
- Layout Simulators: Assura, Calibre, Hercules...
... and EDA Softwares?

System Simulators
ADS, CoCentric, Matlab, Simulink, SPW, Excel ...

Mixed-signal simulators
ADVance MS, Smash, AMS Designer, Saber ...

Circuit Simulators
Eldo RF, Spectre, Spice, ADS ...

Layout Simulators
Assura, Calibre, Hercules ...

Now, we can impact system simulations!
One solution with ADSME

Introduction

- One solution with ADSME
- Digital Simulator
- AMS Behavioral
- Transistor Level
- Mentor Graphics ADVance MS Software
  - VHDL, SystemVerilog
  - Verilog, SystemC
  - VHDL-AMS
  - Verilog-AMS
- Mathworks Simulink & Matlab Softwares
  - Library blocks
  - User-defined blocks
  - Matlab code
  - Other Languages
- Simulink
- Matlab
- Real Time Workshop
- Fast Spice
- Spice
- RF Simulation
- Different targets
- Fixed-Step
- C export

How can I send back it?
How can I do it?
One solution with ADSME

Introduction

- Mentor Graphics ADVance MS Software
  - VHDL, SystemVerilog
  - Verilog, SystemC
  - VHDL-AMS
  - Verilog-AMS

- Transistor Level
  - Fast Spice
  - Spice
  - RF Simulation

- AMS Behavioral
  - VHDL-AMS
  - Verilog-AMS

- Digital Simulator
  - VHDL, SystemVerilog
  - Verilog, SystemC

- Mathworks Simulink & Matlab Softwares
  - Library blocks
  - User-defined blocks
  - Matlab code
  - Other Languages

- Matlab
  - Matlab code

- Simulink
  - Library blocks
  - User-defined blocks

- ADMS Target
- VHDL Implementation
- ADVance SME

- Real Time Workshop
Introduction

One solution with ADSME

System Specifications

Environment

Previous Project

Time-to-Market

Subsystem

DSP

Subsystem

RF

Subsystem

BB

System Level Models

Matlab / Simulink or VHDL-AMS / Verilog-AMS (??)

or Excel, ADS...

Block 1

Block 2

Block 3

...

Block N

SPICE, Simulink + VHDL-AMS

SPICE, Simulink + Verilog-AMS

SPICE + Simulink + VHDL + Verilog

Simulink + VHDL-AMS + VHDL

Simulink + Verilog-AMS + Verilog + SystemC

Simulink + VHDL-AMS + SPICE + SystemC

...

Now, I can mixed description languages.

SPICE implementation, Transistors Level
Introduction
- Classical simulation support
- Proposed simulation support
- ... and EDA Softwares ?
- One solution with ADVance SME

RF Library presentation
- Noise Figure implementation
- Phase Noise implementation
- RF blocks modeling summary

Transceiver simulation results
- Specifications extraction
- Transceiver results
- Focus on Transmitter

Conclusion
Noise Figure Implementation

**LNA (1):**

- **Input Impedance**
- **Frequency Response**
- **Nonlinear Response**
- **Output Impedance**

Noise Figure Generation

VHDL or Simulink

VHDL-AMS or Simulink

LNA_{in} \rightarrow + \rightarrow ... \rightarrow LNA_{out}
LNA (1):

Noise Figure Implementation

LNA in  →  LNA out

Input Impedance  →  Frequency Response  →  Nonlinear Response  →  Output Impedance

VHDL-AMS or Simulink

Noise Figure Generation

VHDL or Simulink

\[ TF = \frac{1.0}{1 + j\left(\frac{\omega}{\omega_1} - \frac{\omega_2}{\omega}\right)} \]

\[ G_v = \sqrt{G_p \cdot \frac{Z_{\text{input}}}{Z_{\text{output}}}} \]

Input Impedance

Frequency Response

Nonlinear Characteristic

Output Impedance
Noise Figure Implementation

**LNA (2):**

- **Input Impedance**
- **Frequency Response**
- **Nonlinear Characteristic**
- **Output Impedance**

Noise Figure Generation

VHDL-AMS or Simulink

VHDL or Simulink

\[
\text{If } (V_{in} < \text{Maximal Input})
\]

Output Voltage = \(G_{v1} \times V_{in} - G_{v3} \times V_{in}^3\)

\[
\text{Elsif } (V_{in} \text{ positive})
\]

Output Voltage = Maximal output Voltage

\[
\text{Else}
\]

Output Voltage = - Maximal output Voltage
**Noise Figure Implementation**

**LNA (3):**

- **Input Impedance**
- **Frequency Response**
- **Nonlinear Response**
- **Output Impedance**

\[
\sigma_{\text{noise}}^2 = 4 \cdot k_B \cdot T_0 \cdot R_{\text{Input}} \cdot (F - 1)
\]
Noise Figure Implementation

**LNA (4):**

Voltage gain & NF extraction – SPICE Simulations

- **Noise Figure Implementation**
  - Power Gain: 15.06 dB
  - ICP1: -7.63 dBm
  - Noise Figure: 1.81 dB
  - Power: 4.8 mW
  - Input Impedance: 49.48 Ω
  - Output Impedance: 50.09 Ω

Will be presented on IEEE ICECS 2007 - Morocco
Phase Noise Implementation

VCO (1):

- White Noise Generation
- Transfer Function: $1/f^2$ and $1/f^3$
- Floor Noise
- Neighbor cut-off frequency
- FN conversion cut-off frequency

Simulink

- Sub-band calibration
- Voltage Control
- VCO Gain
- Quiescent Frequency

Modulo Integrator

Cosine generation

Output Frequency

VHDL-AMS
Phase Noise Implementation

VCO (2):

VCO Architecture

- VCO 5 / 3,5 GHz
- Divider by 4
  - GSM/DCS/GPS
- Divider by 2
  - WIFI/BT/DCS/GPS
- WLAN

- Multiple VCO gains
- Multiple PN characteristics
- Multiple behaviors

VCO Architecture

Frequency Variation Modeling

“High Level Modelling of ΣΔ Fractional PLL for Noise Estimation”

Transient VCO Modeling

VCO Output

Divider by 2

Divider by 4

Inductor Command

Time (s)
Phase Noise Implementation

VCO (3):

- Output Amplitude Voltage (V)
- Quiescent Frequency (Hz)
- VCO Gain (Hz/V)
- Initial Phase (rad)
- Neighbor Noise Frequency (Hz)
- Floor Noise Frequency (Hz)
- Floor Noise Level (dBc/Hz)

Simulink Model (included & user-defined)

Oscillation Frequency (Hz)

Tune Voltage (V)

Nonlinear
Ideal

Noisy VCO model in Simulink

Phase Noise (dBc/Hz)

Offset Frequency (Hz)

“Top-Down PLL Design Methodology combining Block Diagram, Behavioral and Transistor Level Simulators”
### Library Summary

<table>
<thead>
<tr>
<th>Component</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Noise Amplifier</td>
<td><strong>Power Gain, 3rd order Intercept Point, Input – Output impedance matching, Frequency, Noise figure</strong></td>
</tr>
<tr>
<td>Power Amplifier</td>
<td><strong>Power Gain, 3rd order Intercept Point, Input – Output impedance matching, Frequency, Noise figure</strong></td>
</tr>
<tr>
<td>Mixer</td>
<td><strong>Power Gain, 3rd order Intercept Point, Input – Output impedance matching, Frequency, Noise figure and Phase Noise</strong></td>
</tr>
<tr>
<td>Filter</td>
<td><strong>Laplace modeling (or Z transform) Input – Output impedance matching, Noise figure</strong></td>
</tr>
<tr>
<td>Channel</td>
<td><strong>Input – Output impedance matching, Noise figure, Phase Noise, Distance, Frequency, IQ imbalance</strong></td>
</tr>
</tbody>
</table>
## Library Summary

<table>
<thead>
<tr>
<th>Block</th>
<th>Parameters</th>
<th>2nd order Intercept Point</th>
<th>Next steps?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Noise Amplifier</td>
<td>Power Gain, 3rd order Intercept Point, Input – Output impedance matching, Frequency, Noise figure</td>
<td>2nd order Intercept Point</td>
<td></td>
</tr>
<tr>
<td>Power Amplifier</td>
<td>Power Gain, 3rd order Intercept Point, Input – Output impedance matching, Frequency, Noise figure</td>
<td>2nd order Intercept Point</td>
<td></td>
</tr>
<tr>
<td>Mixer</td>
<td>Power Gain, 3rd order Intercept Point, Input – Output impedance matching, Frequency, Noise figure and Phase Noise</td>
<td>2nd order Intercept Point, losses between ports</td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>Laplace modeling (or Z transform) Input – Output impedance matching, Noise figure</td>
<td>Any idea ??</td>
<td></td>
</tr>
<tr>
<td>Channel</td>
<td>Input – Output impedance matching, Noise figure, Phase Noise, Distance, Frequency, IQ imbalance</td>
<td>(work in progress) Radiation pattern</td>
<td></td>
</tr>
</tbody>
</table>
Introduction
- Classical simulation support
- Proposed simulation support
- ... and EDA Softwares?
- One solution with ADVance SME

RF Library presentation
- Noise Figure implementation
- Phase Noise implementation
- RF blocks modeling summary

Transceiver simulation results
- Specifications extraction
- Transceiver results
- Focus on Transmitter

Conclusion
Specifications extraction

Bluetooth Transceiver (1)?
Specifications extraction

Bluetooth Transceiver (2)

Transceiver simulation results

Device 1

Device Master
DSP

TX BB Modulator

Hopping

Bits

I

Q

TX RF
IF + RF

Channel

RX RF
IF + RF

RF_{out}

Device 2

Device Slave
DSP

RX BB Demodulator

Hopping

Bits

I

Q

RF_{in}
Requirements & objectives

- Bluetooth RF Transceiver modeling => Block specifications extraction

- Bluetooth Baseband Transceiver modeling => Modulation & BER

- Objectives:

Specifications extraction

Transceiver simulation results
Requirements & objectives

- Bluetooth RF Transceiver modeling => Block specifications extraction
  - 2.4 GHz ISM Band – Receiver Sensitivity: -73 dBm - Power: 1mW to 100 mW
  - Quadrature waveforms I/Q – Data Rate: 1 Mbps (1, 2, 3 bits symbol)
  - Transmitter & Receiver with dual-conversion technique (2, 4 / 1, 6 / 0, 8 GHz)
  - Need of a RF library with key generic and critical parameters:
    - LNA: Gain, IP3 & Noise Figure, Impedance Mismatches
    - PA: Gain, IP3 & Impedance Mismatches
    - Mixers: Gain, IP3 & Noise Figure, Phase Noise & Impedance Mismatches
    - Channel: Distance, Frequency & Attenuation, Phase Noise, White Noise Adjunction

- Bluetooth Baseband Transceiver modeling => Modulation & BER
  - Bit Error Rate = 0.1%
  - Gaussian Frequency Shift Keying (GFSK) + Frequency Hopping (from -39 to 39 MHz)

- Objectives:
  - Adjust the RF Block Specifications to match BER
  - BT Transceiver Model with hierarchical level (Simulink / ADMS / SPICE) and mixes it!
  - Bonus: Combines it with SystemC (protocol / baseband)
Transceiver results

Device 1

- Device Master
- DSP
- Simulink
- TX BB
- Modulator
- I
- Q
- Simulink
- TX RF
- IF + RF
- VHDL-AMS
- Channel
- RF_{out}

Device 2

- Device Slave
- DSP
- Simulink
- RX BB
- Demodulator
- I
- Q
- Simulink
- RX RF
- IF + RF
- VHDL-AMS
- RF_{in}

Transceiver simulation results
### Transceiver results

**Required Specifications for each blocks along the BT signal in the transceiver**

<table>
<thead>
<tr>
<th>Blocks name</th>
<th>RF Transmitter</th>
<th>RF Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IF Mixers</td>
<td>RF Mixer</td>
</tr>
<tr>
<td>Gain (dB)</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>IIP3 (dBm)</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Noise Figure (dB)</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

**Transceiver simulations results (work in progress)**

<table>
<thead>
<tr>
<th>Software</th>
<th>MAC</th>
<th>Channel</th>
<th>PA</th>
<th>RX Mixers</th>
<th>TX Mixers</th>
<th>LNA</th>
<th>Elapsed</th>
<th>Bits</th>
<th>BER</th>
<th>1 bit</th>
<th>Obs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Attenuation</td>
<td>Ideal</td>
<td>Ideal</td>
<td>Ideal</td>
<td>30'</td>
<td>1211</td>
<td>0.0</td>
<td>1.49&quot;</td>
<td>Ts=1e-10</td>
<td></td>
</tr>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Attenuation</td>
<td>Real</td>
<td>Ideal</td>
<td>Ideal</td>
<td>38'</td>
<td>1051</td>
<td>0.0</td>
<td>2.17&quot;</td>
<td>Ts=1e-10</td>
<td></td>
</tr>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Attenuation</td>
<td>Real</td>
<td>Real</td>
<td>Ideal</td>
<td>49'</td>
<td>1033</td>
<td>0.0</td>
<td>2.84&quot;</td>
<td>Ts=1e-10</td>
<td></td>
</tr>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Attenuation</td>
<td>Real</td>
<td>Real</td>
<td>Real</td>
<td>14h17'</td>
<td>10780</td>
<td>0.0</td>
<td>4.77&quot;</td>
<td>Ts=1e-10</td>
<td></td>
</tr>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Attenuation</td>
<td>Real</td>
<td>Real</td>
<td>Real</td>
<td>81h</td>
<td>4270</td>
<td>0.0</td>
<td>68.29&quot;</td>
<td>Ts=1e-11</td>
<td></td>
</tr>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Attenuation</td>
<td>Real</td>
<td>Real</td>
<td>Real+NF</td>
<td>110h</td>
<td>6067</td>
<td>0.034</td>
<td>64.85&quot;</td>
<td>Ts=1e-11</td>
<td></td>
</tr>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Attenuation</td>
<td>Real</td>
<td>Real</td>
<td>Real+NF</td>
<td>96h</td>
<td>5495</td>
<td>0.469</td>
<td>62.89&quot;</td>
<td>Ts=1e-11</td>
<td></td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>67&quot;</td>
<td>4333</td>
<td>0.0</td>
<td>0.928&quot;</td>
<td>Without RF</td>
<td></td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Sim.</td>
<td>Real</td>
<td>Real</td>
<td>11h37'</td>
<td>3133</td>
<td>0.0</td>
<td>13.34&quot;</td>
<td>Without NF</td>
<td></td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Sim.</td>
<td>Real</td>
<td>Real+NF</td>
<td>7h53'</td>
<td>2600</td>
<td>0.02</td>
<td>10.91&quot;</td>
<td>With NF</td>
<td></td>
</tr>
</tbody>
</table>

Simulation time decreases by 5 ~ 6 x ratio

Allows mixed SPICE / Behavior / Simulink description levels (and SysC...) → (next point)
Focus on Transmitter

Transceiver simulation results

Device 1

Device Master
DSP

SystemC
Simulink

Simulink

TX BB
Modulator

Hopping
Bits

Simulink

TX RF
IF + RF

Simulink
VHDL-AMS

VHDL-AMS
Verilog-AMS
SPICE

PA

RF_out
Focus on Transmitter

Transceiver simulation results

Bluetooth Emitter – Simulations comparison – 75 µs simulation time

<table>
<thead>
<tr>
<th>Software</th>
<th>Bit – Hopping</th>
<th>Baseband part</th>
<th>RF part</th>
<th>Power Amplifier</th>
<th>Simulation Type</th>
<th>CPU Time (sec)</th>
<th>Elapsed Time (sec)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Fixed Step</td>
<td>NC</td>
<td>2755</td>
<td>1</td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Transient – Dig.</td>
<td>309</td>
<td>473</td>
<td>5.82</td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>VHDL-AMS (Ours)</td>
<td>Transient</td>
<td>1171</td>
<td>1493</td>
<td>1.85</td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Verilog-AMS (Commlib)</td>
<td>Transient</td>
<td>1153</td>
<td>1397</td>
<td>1.97</td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>SPICE</td>
<td>Transient</td>
<td>1180</td>
<td>1431</td>
<td>1.93</td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>VHDL-AMS (Commlib)</td>
<td>MODSST (RF + PA)</td>
<td>3997</td>
<td>4386</td>
<td>0.63</td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Verilog-AMS (Commlib)</td>
<td>MODSST (RF + PA)</td>
<td>4301</td>
<td>5161</td>
<td>0.53</td>
</tr>
<tr>
<td>ADMS</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>SPICE</td>
<td>MODSST (RF + PA)</td>
<td>5022</td>
<td>6026</td>
<td>0.46</td>
</tr>
<tr>
<td>ADMS</td>
<td>SystemC</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Transient</td>
<td>29570</td>
<td>29843</td>
<td>0.09</td>
</tr>
<tr>
<td>ADMS</td>
<td>SystemC</td>
<td>Simulink</td>
<td>Simulink</td>
<td>VHDL-AMS (Commlib)</td>
<td>Transient</td>
<td>276</td>
<td>378</td>
<td>7.29</td>
</tr>
<tr>
<td>ADMS</td>
<td>SystemC</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Verilog-AMS (Commlib)</td>
<td>Transient</td>
<td>1108</td>
<td>1284</td>
<td>2.15</td>
</tr>
<tr>
<td>ADMS</td>
<td>SystemC</td>
<td>Simulink</td>
<td>Simulink</td>
<td>SPICE</td>
<td>Transient</td>
<td>1219</td>
<td>1400</td>
<td>1.97</td>
</tr>
<tr>
<td>ADMS</td>
<td>SystemC</td>
<td>Simulink</td>
<td>Simulink</td>
<td>Transient</td>
<td>Transient</td>
<td>3891</td>
<td>4084</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Conclusion (Simulink simulation basing)

- Simulink exported on ADMS $\rightarrow$ 6x faster
- Simulink exported on ADMS + SystemC $\rightarrow$ 7x faster
- Simulink exported plus behavioral languages and/or SystemC $\rightarrow$ 2x faster
- Simulink exported plus SPICE and/or SystemC $\rightarrow$ 1.4x slower but SPICE visibility (RF instructions)
Introduction
- Classical simulation support
- Proposed simulation support
- ... and EDA Softwares ?
- One solution with ADVance SME

RF Library presentation
- Noise Figure implementation
- Phase Noise implementation
- RF blocks modeling summary

Transceiver simulation results
- Specifications extraction
- Transceiver results
- Focus on Transmitter

Conclusion
Conclusion

**Combine description languages:**
- To reduce system complexity,
- To increase model accuracy,
- To reduce some EDA software limitations (as Phase Noise for ADMS),
- To develop model faster!

**Need a generic RF Library:**
- To reduce simulation time,
- To allow model re-use with generic / critical parameters,
- To mix EDA softwares, mix system and circuit designers (utopia?)

**Conclusion on Transceiver Work:**
- Simulink or SystemC exported on ADMS → 6 to 7x faster (Simulink to ADMS)
- Mixed Simulink/SystemC and SPICE on ADMS → 2x slower (ADMS)
- SPICE circuit simulation with MODSST algorithms → 34x faster than transient (Eldo RF)

Next steps: - Combines RF instructions for SPICE and abstraction (SystemC and Simulink) acceleration to reduce simulation time!
- Validate this methodology with a ZigBee Transceiver Circuit
Thank you for your attention. Any questions?
Transceiver results

Transceiver simulation results

Device - Master

Frequency synthesis

LO_1 MHz
LO_1 1.6 GHz
LO_2 0.8 GHz
LO_2_1
LO_2_2

Frequency Hopping Index

TX

D2A

GFSK Modulator

Frequency Hopping

I way

LO_2_1

Q way

Bandpass filter Power Amplifier Antenna Filter

Heterodyne transmitter

P/P

Fris

White Noise, Phase Noise, IQ imbalance, Receiver thermal noise

Device - Slave

Frequency Hopping Index

RX

GFSK Demodulator

IF Amplifier

Baseband filter

I way

Q way

A2D

IF

I Mixer

Q Mixer

IF Mixer

LNA

Heterodyne receiver

LO_2_1

LO_1

LO_2_2
Focus on Transmitter

**Transceiver simulation results**

**Simulink description**

Baseband Emitter

- Cyclic Encoder + Buffer
  - 625 samples @ 1.0 µs
- GFSK Modulator
  - 62500 samples @ 1.0 µs
- Frequency Hopping
  - 1 sample @ 625 µs

**DSP Part**

- Random Bit Generator
  - 10 samples @ 15 µs

**SystemC description**

Index Generation

**Simulink description**

LO_1

- LO_2_I
  - Re
  - freq
  - 0.8G
- LO_2_Q
  - Re
  - freq
  - 0.8G

**Verilog-AMS behavior**

Bandpass filter

**Power Amplifier**

Antenna

1 sample @ 625 µs

**Different descriptions but focused on key blocks:**

- **DSP Part as Simulink or SystemC**
- **Baseband Part as Simulink**
- **RF Transmitter: Simulink + VHDL-AMS**
- **Power Amplifier: Simulink / Verilog-AMS / VHDL-AMS / SPICE circuit**