CMOS Sigma-Delta Converters – From Basics to State-of-the-Art

Systematic Design Methodology

Rocío del Río, Belén Pérez-Verdú and José M. de la Rosa

{rocio,belen,jrosa}@imse.cnm.es

KTH, Stockholm, April 23-27

OUTLINE

1. Introduction
   • Digital vs. Analog/Mixed-signal design
   • Simulation approaches
   • Hierarchical synthesis approach
2. Top-down design methodology
   • Top-down/Bottom-up approach
   • Optimization engine
   • Behavioral simulation
3. Introduction to SIMSIDES
   • Description of the toolbox
   • Behavioral modeling of building blocks
4. DEMO & Tutorial examples
Introduction: Tendency to System Integration

From PCB (Printed-on-Board) To SoC (Systems-on-Chip)
- Portability and robustness increase
- Reduce cost
- Benefits from technology evolution
- Increase programability

To SiP (Systems-in-Package)
- Design for System
- Design for Assembly
- Design for Test

Introduction: Digital vs. Analog & Mixed-Signal design

(P.R. Gray 1992)

Signal sources: Digital VLSI, Analog
Transmission media: Fiber, Cables, Wireless

Low Power, Low Cost, High-Speed

DIGITAL ON-DIE TRANSCEIVERS & PORTABILITY
Introduction: Digital vs. Analog & Mixed-Signal design

Digital Circuits
- Robustness, programmability and adaptability to different standards and protocols
- Increased performance with technology scaling
- Simplicity of the design and testing
  - Clear hierarchy leading to well-defined levels of abstraction
  - Circuit blocks are largely independent of each other
- Automatic top-down design from specs to silicon

A/D/A Interface = Bottle Neck of the Design Process
- Technology evolution demands for faster, precise and low-power A/D/A Interfaces
- Analog Circuits in Technologies dedicated to Digital Design (standard ULSI CMOS)
  - Continuously decreasing supply voltages
  - Reduced channel length and large threshold voltage in MOS transistors
  - Poor matching properties and linearity
  - Incomplete models for the analog design needs
- Proximity of noisy digital circuits
- Need of Design Automation (DA)

Introduction: Digital CAD tools

ARCHITECTURAL DESIGN FLOOR-PLANNING

LOGIC DESIGN LOGIC SIMULATION (FUNCTIONAL DESCRIPTION)

CIRCUIT DESIGN CIRCUIT SIMULATION (PARAMETRICAL DESCRIPTION)

PHYSICAL DESIGN LAYOUT LIBRARY CELLS, ROUTING (GEOMETRICAL DESCRIPTION)

FABRICATION AND TESTING TEST VECTORS

COMMON DATA BASE CAD TOOLS

- At each design-flow level:
  - Generation of a solution
  - Simulation and verification
- Properly supported by CAD tools:
  - Simulators, verification tools, ....
  - Graphical interfaces: schematic and layout editors, ....
Introduction: Analog CAD tools

SPECIFICATIONS
PRELIMINAR DESIGN
ELECTRICAL SIMULATION
No
OK?
LAYOUT
PARASITIC EXTRACTION AND VERIFICATION
ELECTRICAL SIMULATION
No
OK?
FABRICATION AND TEST

Until a few years ago: only SPICE-like simulators and post-layout verification tools

Schematic capture
Introduction: Analog CAD tools

- Clearly inefficient for the design of complex systems
- Excessive consumption of computational resources

Introduction: Simulation approaches

Design Methodology of ICs

- Electrical simulation of relatively simple discrete-time (SC, SI) circuits is **COMPLETELY INEFFICIENT**
  - Need of time-domain analysis
  - Several thousands of clock cycles required

Example: A 2nd-order $\Sigma$-ADC

- Several weeks of CPU time to obtain SNR
- Need of searching other simulation approaches
Introduction: Simulation approaches

- **Electrical simulation with macromodels**: Uses numerical algorithms to solve the differential equations that result from the analysis of a circuit, thus resulting in long CPU times.
- **Multi-level simulation**: The critical parts of the system are simulated numerically while behavioural models are incorporated to emulate the rest of the system. (ELDO, SABER...)
- **Behavioural simulation (event-driven)**: The system is broken up into a set of subcircuits, often called building blocks or basic blocks. These blocks are described by explicit equations that relate the outputs in terms of the inputs and the internal state variables. Thus, the accuracy of the simulation depends on how precisely those equations describe the real behaviour of each block. The transient evolution of the voltages and currents is not important, only the final value. (TOSCA, ASIDES, MATLAB-BASED SIMULATORS...)

<table>
<thead>
<tr>
<th>Speed (CPU time)</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Days/Weeks</td>
<td>Electrical</td>
</tr>
<tr>
<td>Hours</td>
<td>Multi-level</td>
</tr>
<tr>
<td>Minutes</td>
<td>Behavioural</td>
</tr>
</tbody>
</table>

Introduction: Hierarchical synthesis approach

* A very popular strategy to synthesize complex blocks is to hierarchize the design

- The complex system is partitioned in simpler blocks with relatively independent functionality.
- In each level of the hierarchy the sizing process involves selecting the architecture and transmitting the specifications to the lower level.

Mapping specs. on design parameters

Specifications at level i

Architecture selection

Specification transmission

Verification of the performance

OK? N Y

Specifications at level i - 1

Analysis

Design (Sizing)
Introduction: Synthesis methods

- **Knowledge-based tools**: capture the knowledge of experienced designers
  - Short execution times
  - Not optimized: design procedures usually based on approximate equations and models
  - Closed tools
    - Limited to a reduced number of topologies
    - Addition of new ones usually restricted to the tool developers

- **Optimization-based tools**: based on an iterative optimization procedure
  - Cost function evaluation by numerical methods: *equations* or *simulations*
  - Two main optimization techniques:
    - **Deterministic**: parameter updating needs information on the cost function and on their derivatives
      - Optimization process may be trapped in a local minimum of the cost function
      - Useful for fine tuning of suboptimal designs
    - **Statistical**: parameters are changed randomly
      - Avoid local minima
      - Appropriate for global optimization
      - No good initial design point is needed
      - Requires larger computational cost

---

Introduction: Optimization-based synthesis approach

- **ΣΔM specifications**
- Initial design point
- Performance evaluation
- New design point
- End: Yes
- Specs OK? (Yes/No)

© IMSE-CNM ΣΔ Design Group
Introduction

Design Methodology: Top-down & bottom-up approach
**Design Methodology: Top-down & bottom-up approach**

1. **Modulator specifications**
2. **Architecture selection**
3. **Building blocks specifications**
4. **Simulation ok?**
5. **Cell architecture selection**
6. **Opamp**
7. **Quantizers**
8. **Switches**
9. **Simulation ok?**
10. **LAYOUT**

**IMSE ΣΔ Design Group**

**Design Methodology: Optimization**

**BASIC FEATURES**

- Combines adaptive statistical and deterministic optimization
- Cost function formulation very versatile
- Addition of knowledge is permitted

**Speed Requirements ~ s**

**Performance evaluator**

- **Modulator Topology**
- **Design Space Discretization**
- **Cost function evaluation**
- **Main Optimization**
- **Local Optimization**
- Update design parameters
- End? Yes
- Hundredths or thousands of iterations
- End? No
- Update design parameters
OPTIMIZATION-BASED ENGINE (FRIDGE)

- Interacts with any kind of performance evaluation approach
- Statistical + Deterministic techniques
- Designer’s expertise can be added through powerful tools (embedded C++)
- Between Optimization and Knowledge-based approaches, taking the best from both worlds
- Used both for spec transmission and for cell-level sizing
- In process of being complemented with Evolutionary Algorithms

- Intense internal use

ITERATIVE CYCLE

Initial position $x_0$  
$x_n = x_{n-1} + \Delta x_{n-1}$

Control of the tuning process

END

Specs OK?  
yes no

FORMULATION OF THE COST FUNCTION

Problem: 
\[
\text{minimize } \sum_{i=1}^{P} \psi_i(x_i) \quad 1 \leq i \leq P
\]
subjected to 
\[
y_{rk}(x) \geq y_{rk} \quad \text{or} \quad y_{rk}(x) \leq y_{rk} \quad 1 \leq k \leq R
\]

Target Specification
P design objectives
R restrictions
Cost Function

EXAMPLE OF A COST FUNCTION

Weights used to give priority to the fulfillment of their corresponding specifications

\[
\Psi(x) = \begin{cases} 
\sum w_i \log(y_{ik}) & \text{if } x \in R_A \\
\max \left[ -w_i \log \left( \frac{y_i}{y_{rk}} \right) \right] & \text{if } x \notin R_A 
\end{cases}
\]
Simulation of $\Sigma\Delta$Ms:
- Strongly non-linear circuits
- Oversampling $\rightarrow$ long time-domain simulation

Techniques

- **Accuracy**
  - Electrical: days/weeks
  - Mixed-mode: hours
  - Behavioral: seconds

MATLAB-SIMULINK is a widely spread platform.
- Direct access to very powerful tools for signal processing and data manipulation.
- Provides a high-level language to create custom functions, structures, Graphical User Interfaces (GUIs), etc.

**Simulink block libraries**
*(Malcovati et al., IEEE Trans. CAS, 2003)*
- Limited to SC circuits.
- Limited accuracy:
  - Transient response in both clock-phases not considered.
  - Non-linear opamp DC gain, non-linear switch-on resistance, non-linear capacitors not included.
- Models based on MATLAB functions $\rightarrow$ excessive CPU time.
SIMSIDES: Description of the toolbox

- Graphical User Interface
- Collection of post-processing routines
- SC, SI and CT techniques
- Global and efficient optimization techniques
- Precise behavioural models
SIMSIDES: Description of the toolbox

IDEAL LIBRARIES

INTEGRATORS

Continuous-Time

Switched-Capacitor

Switched-Current

Gm-C

Gm-MC

RC

MOSFET-C

QUANTIZERS

REAL LIBRARIES

INTEGRATORS

Continuous-Time

Switched-Capacitor

Switched-Current

Gm-C

Gm-MC

RC

MOSFET-C

QUANTIZERS

DACs

RESONATORS

SIMSIDES: Behavioral modeling using S-functions

S-functions

• Precise models:
  • All major non-idealities included
  • Time-domain models
  • Models based on C-code ➔ fast!!

65536 points
All non-idealities

<table>
<thead>
<tr>
<th>S-functions</th>
<th>M-functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 s.</td>
<td>141 s.</td>
</tr>
</tbody>
</table>

© IMSE-CNM ΣΔ Design Group
SIMSIDE: Creating S-functions

- Steps to create an S-function \( \Sigma \Delta \)M Building Block

1. Behavioral Model
2. Generate C-Code
3. Create S-Function
4. Compilation (mos)
5. Create S-Function Block
6. Create Mask

- Example

Building Block
Model Flow graph
C. S-function file
SIMULINK Implementation

Jitter
Clock
Non-linearity, gain error, loop delay, offset
Quantizers
DACs
Hysteresis and Offset
Comparators
Gain and integrator errors
Resonators
Non-linear transconductance
Opamps
Integrators
Finite and non-linear open-loop DC gain
Incomplete settling error (both clock phases considered)
Output swing limit
Thermal noise
Switches
Thermal noise, switch-on (non-linear) resistance
Capacitors
Mismatch, non-linearity
Resonators
Gain and integrator errors

Circuit Tec.
Opamps
Switches
Capacitors
Resonators
Integrators
Integrators
Resonators
Clock
Comparators
Quantizers

Behavioral models included in the toolbox

- Non-ideality
- Intelligibility
- Non-linearities
- DC gain
- Transient response
- Gain and integrator errors
- Jitter
- Hysteresis and Offset
- Non-linearity, gain error, loop delay, offset
**SIMSIDES: Behavioral modeling of SC FE Integrators**

- **Input Voltages**: $v_1, v_2$
  - Yes: Sampling phase
  - No: Transient Response
- **Memorized Voltage Input**
- **Calculate equivalent thermal noise**: $\Delta v_t$
- **Opamp finite and non-linear open-loop gain**
- **Non-linear capacitors**: $C_{nl} = C(1 + cnl_1 v + cnl_2 v^2)$
- **Transient response**
- **Output Saturation**
- **End**: No

**SIMSIDES: Model accuracy**

- **A 0.25μm CMOS 2-1-1 cascade $\Sigma\Delta$ Modulator for ADSL**
- **A 0.8μm CMOS 4th-order bandpass SI $\Sigma\Delta$ Modulator for digital radio**

---

**SIMSIDES: SIMSIDES**

- **Behavioral modeling of SC FE Integrators**
- **μm CMOS 2-1-1 cascade $\Sigma\Delta$ Modulator for ADSL**
- **μm CMOS 4th-order bandpass SI $\Sigma\Delta$ Modulator for digital radio**

---

**FS MHz**

<table>
<thead>
<tr>
<th>Frequency (f)</th>
<th>SIMSIDES</th>
<th>Measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 MHz</td>
<td>100</td>
<td>101</td>
</tr>
<tr>
<td>40 MHz</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>60 MHz</td>
<td>120</td>
<td>121</td>
</tr>
<tr>
<td>80 MHz</td>
<td>130</td>
<td>131</td>
</tr>
<tr>
<td>100 MHz</td>
<td>140</td>
<td>141</td>
</tr>
</tbody>
</table>

---

**Model accuracy**

- **A 0.25μm CMOS 2-1-1 cascade $\Sigma\Delta$ Modulator for ADSL**
- **A 0.8μm CMOS 4th-order bandpass SI $\Sigma\Delta$ Modulator for digital radio**
SIMSIDES: Simulating a CT 5th-order LP SL ΣΔM

7s. CPU time, 65536 points
P4@1.7GHz

Specifications
- 13bits@4.4Ms/s
- Minimum area and power consumption

Synthesis
All non-idealities 40.8 minutes
SIMSIDES: Sizing a SC 2-1-1 cascade ΣΔM

- **Input Signal**
- **Output Signal**
- **Frequency (Hz)**
- **Power Spectral Density (dB/Hz)**

### Synthesis Toolbox vs. Measured

<table>
<thead>
<tr>
<th>Specification</th>
<th>Synthesis Toolbox</th>
<th>Measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock frequency (MHz)</td>
<td>70.4</td>
<td></td>
</tr>
<tr>
<td>Digital output rate (Ms/s)</td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td>Oversampling ratio</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Reference voltage (V)</td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td>0.25μm <a href="mailto:CMOS@2.5V">CMOS@2.5V</a></td>
<td></td>
</tr>
<tr>
<td>Power consumption (mW)</td>
<td>-</td>
<td>55</td>
</tr>
<tr>
<td>Area (mm²)</td>
<td>-</td>
<td>2.78</td>
</tr>
<tr>
<td>Resolution (bits)</td>
<td>13.3</td>
<td>12.7</td>
</tr>
</tbody>
</table>

**SIMSIDES: Conclusions**

- SIMSIDES is a ΣΔM Synthesis Toolbox in the Matlab/Simulink environment.
  - It allows to efficiently map the modulator specifications into building-block specifications.
  - It deals with the synthesis of ΣΔMs using both DT and CT circuit techniques.
- The implementation platform brings numerous advantages with a relatively low penalty in computation time.

### Simulation Approach

<table>
<thead>
<tr>
<th>Simulation Approach</th>
<th>CPU Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using MATLAB-functions (Brigati, ISCAS99)</td>
<td>415</td>
</tr>
<tr>
<td>Using S-functions (This work)</td>
<td>4-5</td>
</tr>
</tbody>
</table>

**Notes:**

- SIMSIDES is a ΣΔM Synthesis Toolbox in the Matlab/Simulink environment.
- It allows to efficiently map the modulator specifications into building-block specifications.
- It deals with the synthesis of ΣΔMs using both DT and CT circuit techniques.
- The implementation platform brings numerous advantages with a relatively low penalty in computation time.
References


