C-based and circuit-level co-simulation using the Verilog Procedural Interface (VPI)

Nitasha Jugessur, Emma Sosa Morales
NXP Semiconductors, Research - Design Methods and Solutions

Abstract
This presentation will describe the co-simulation methodology using the Verilog Procedural Interface (VPI), which has been applied during the design phase of an Ultra Wide Band (UWB) Radio system.
Co-simulation of circuit descriptions (at transistor or behavioural level) with system descriptions (at C or SystemC level) has been used to investigate the interaction between the digital corrections and compensation techniques implemented in the baseband and the analogue mixed-signal transmit chain.
The C language has been used for the functional description in the baseband, doing numerical and digital signal processing. For the transmitter circuit, a behavioral language such as Verilog-AMS is used to describe its functional model and to allow detection and generation of timing events.

Unfortunately there is no "plug & play" solution available to integrate C/C++ functions in the Cadence mixed-signal design environment to perform mixed-abstraction level, mixed-signal simulations. Therefore a novel method has been studied: co-simulation through the Verilog Procedural Interface (VPI).

The Verilog Procedural Interface is a C-programming interface for the Verilog Hardware Description Language (HDL), and standardized as part of the IEEE 1364.
An overview is given of the most important access and utility functions, using the standard C programming language. A summary of the steps that are necessary in order to be able to use the VPI for this purpose will be presented.

This approach will be compared with the SystemC import flow proposed by Cadence [1]. This flow envelops the C code into a SystemC module, which is embedded into a Verilog-AMS wrapper, which is then available as a library component in the Cadence design environment.

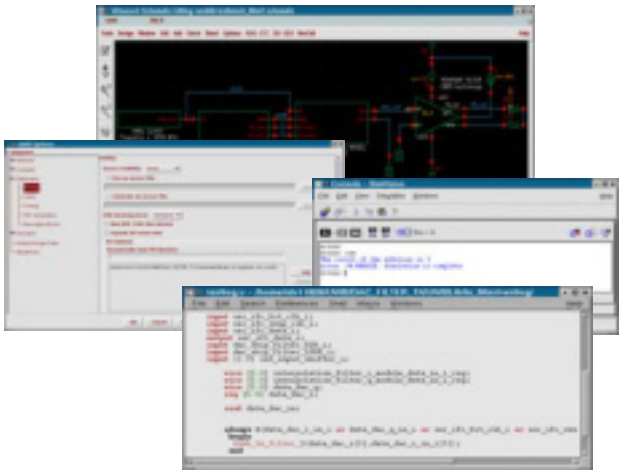


Fig 1. Co-simulation using the Verilog Procedural Interface (VPI)

A simple test case has been chosen which allows to quickly check any difference in the results when using the C function stand-alone and when using the same C description within the Cadence design environment.
A basic example including the digital-to-analogue converter implemented in C and a circuit implementation of a transmit filter is presented.
Based on this example, the co-simulation capabilities using the Verilog Procedural Interface will be compared with the SystemC import approach in terms of the usability in the available flow, the accuracy of the results and the simulation performance. The presentation will show both the advantages and the drawbacks of these methods, and will highlight the inconveniences encountered during the implementation.

References
[1] W. Hartong, P. Birrer, *SpeAC: Leverage from the System Level Success-Story in IC Design*, Cadence Design Systems, MEDEA+ DAC 2006

Simulation of C-based functional models and circuit implementation

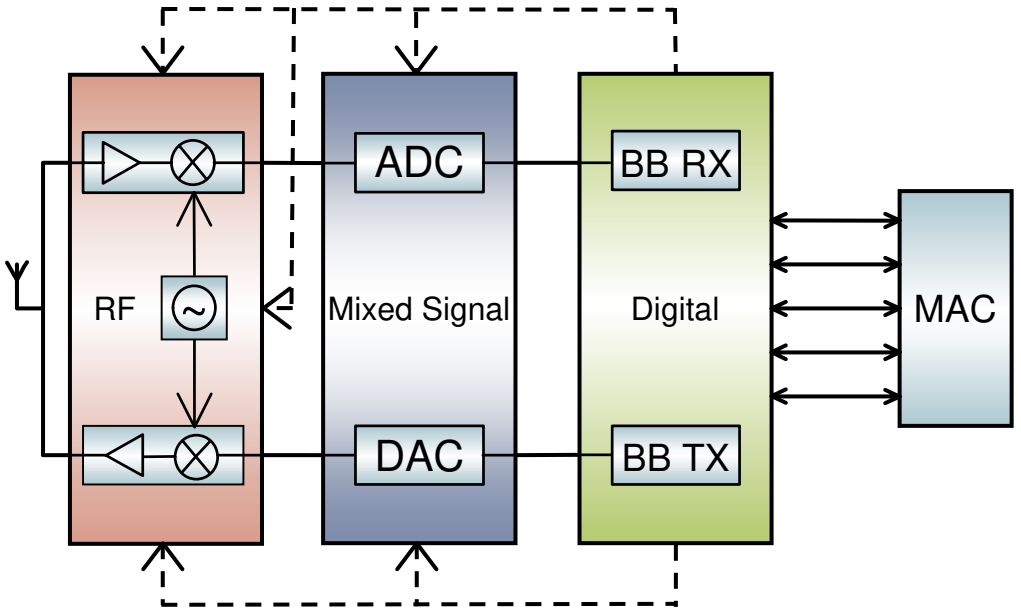
Nitasha Jugessur, Emma Sosa Morales, Giorgia Zucchelli
Design Methods and Solutions – System Design Methods
June 25, 2007

Outline

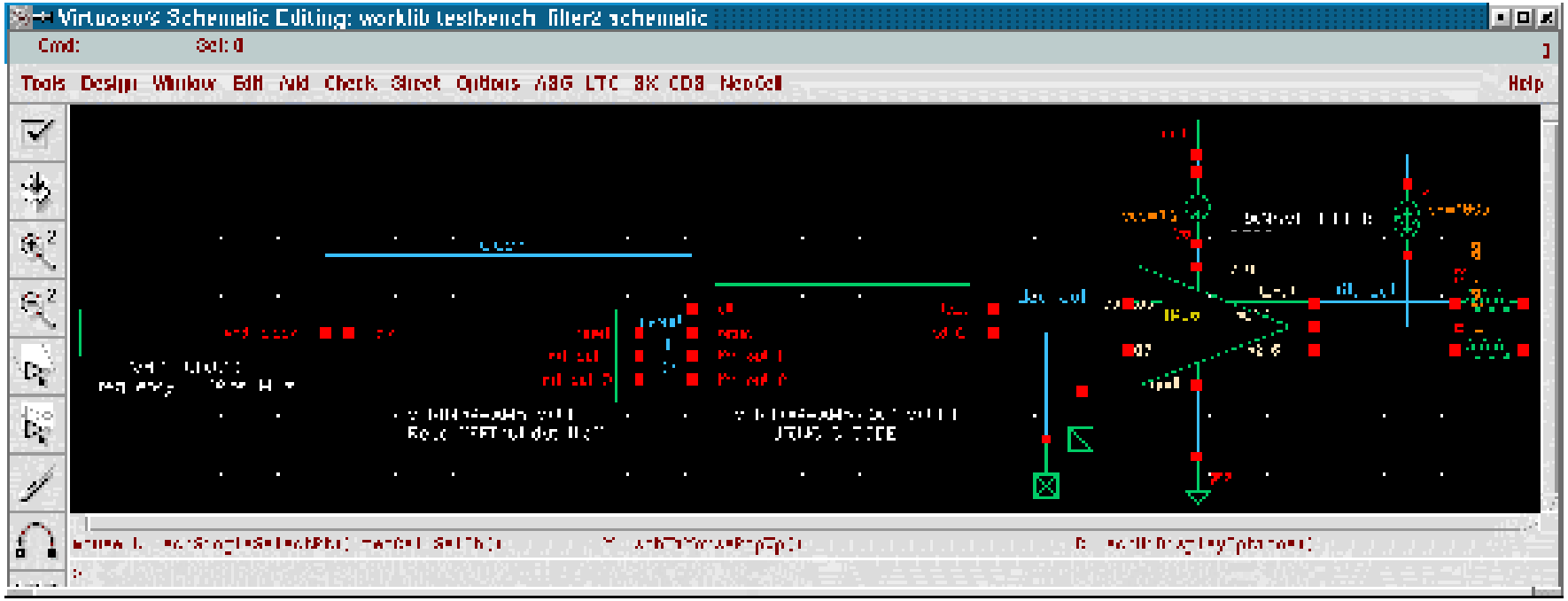
- Introduction
  - Challenges of wireless systems
  - Standard flows in NXP
- Study activities
  - Simple test case
  - Verilog Procedural Interface (VPI)
  - SystemC Import Flow from Cadence
- Conclusions

Challenges of Wireless Systems

- RF, Mixed Signal and Digital in a single chip
- Digital control of analog front-end
- No off-the-shelf integrated simulation solution



Simple test case

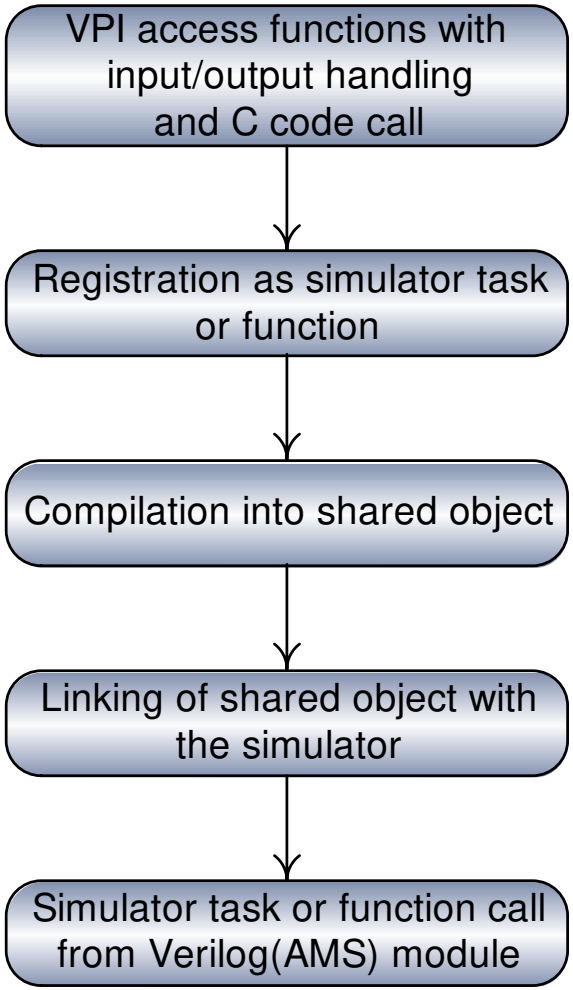


- VHDL Clock at a frequency of 1056 MHz
- Verilog-AMS module reading data from a file at clocked intervals
- Verilog-AMS module using C description of DAC
- Circuit implementation of the transmit filter

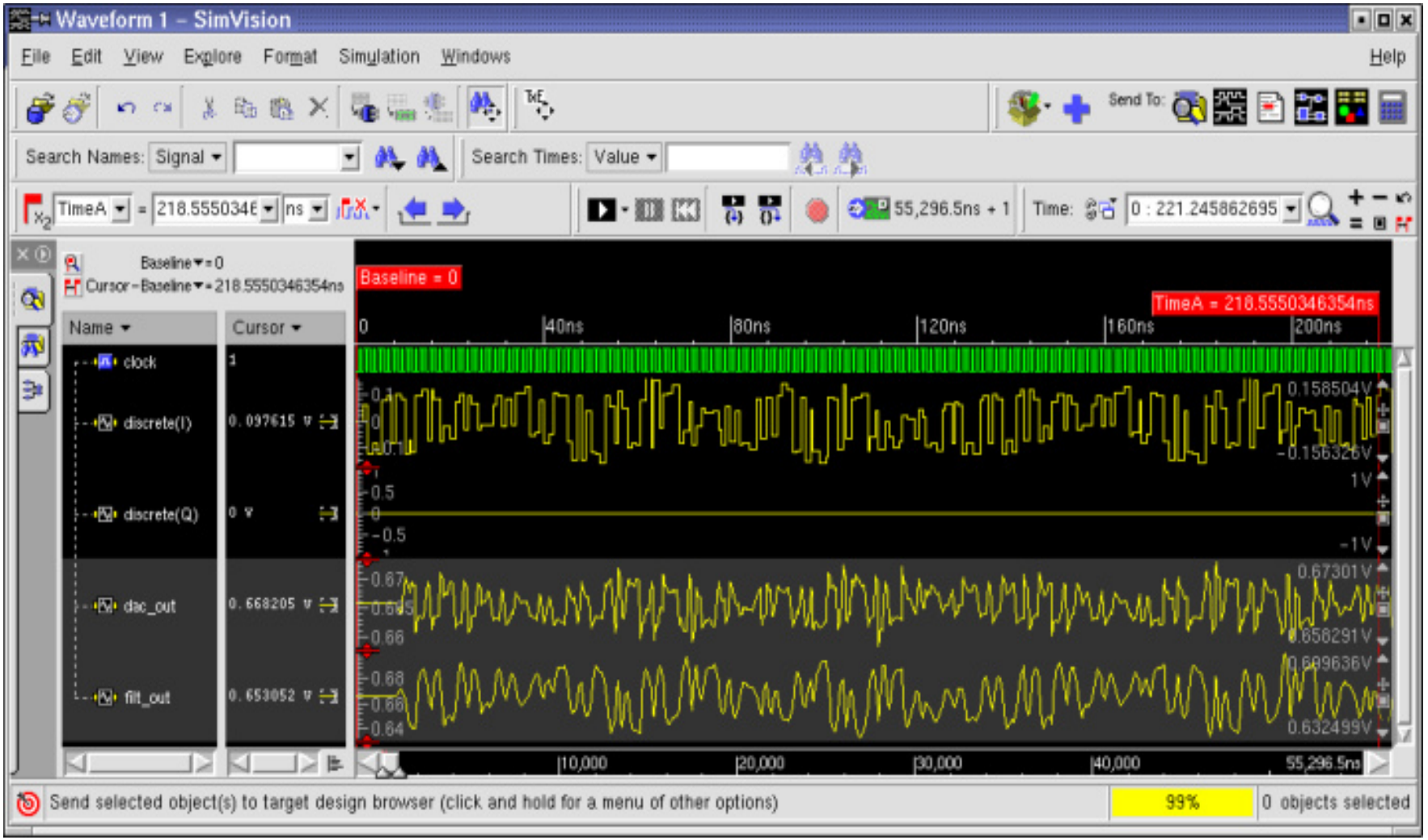
Verilog Procedural Interface (VPI)

- C programming interface for the Verilog Hardware Description Language
- Described in the IEEE 1364 standard Verilog HDL Language Reference Manual
- Consists of a set of access and utility routines called from standard C programming language functions
- Allows applications that create new simulator system tasks and manipulate instantiated simulation objects

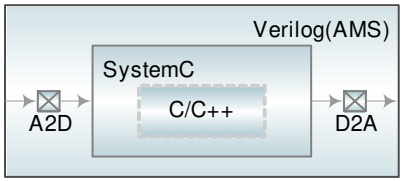
VPI – Work flow



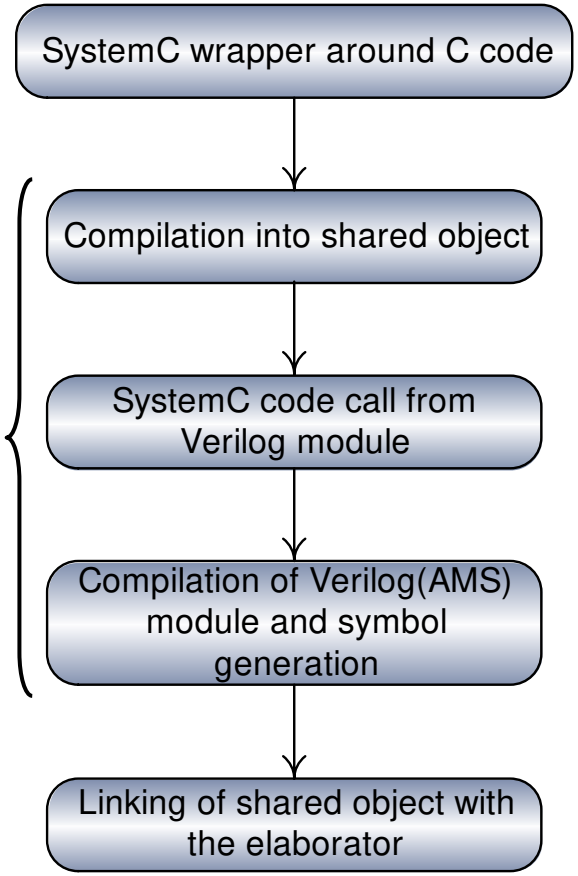
Simulation results



SystemC Import Flow from Cadence [1]

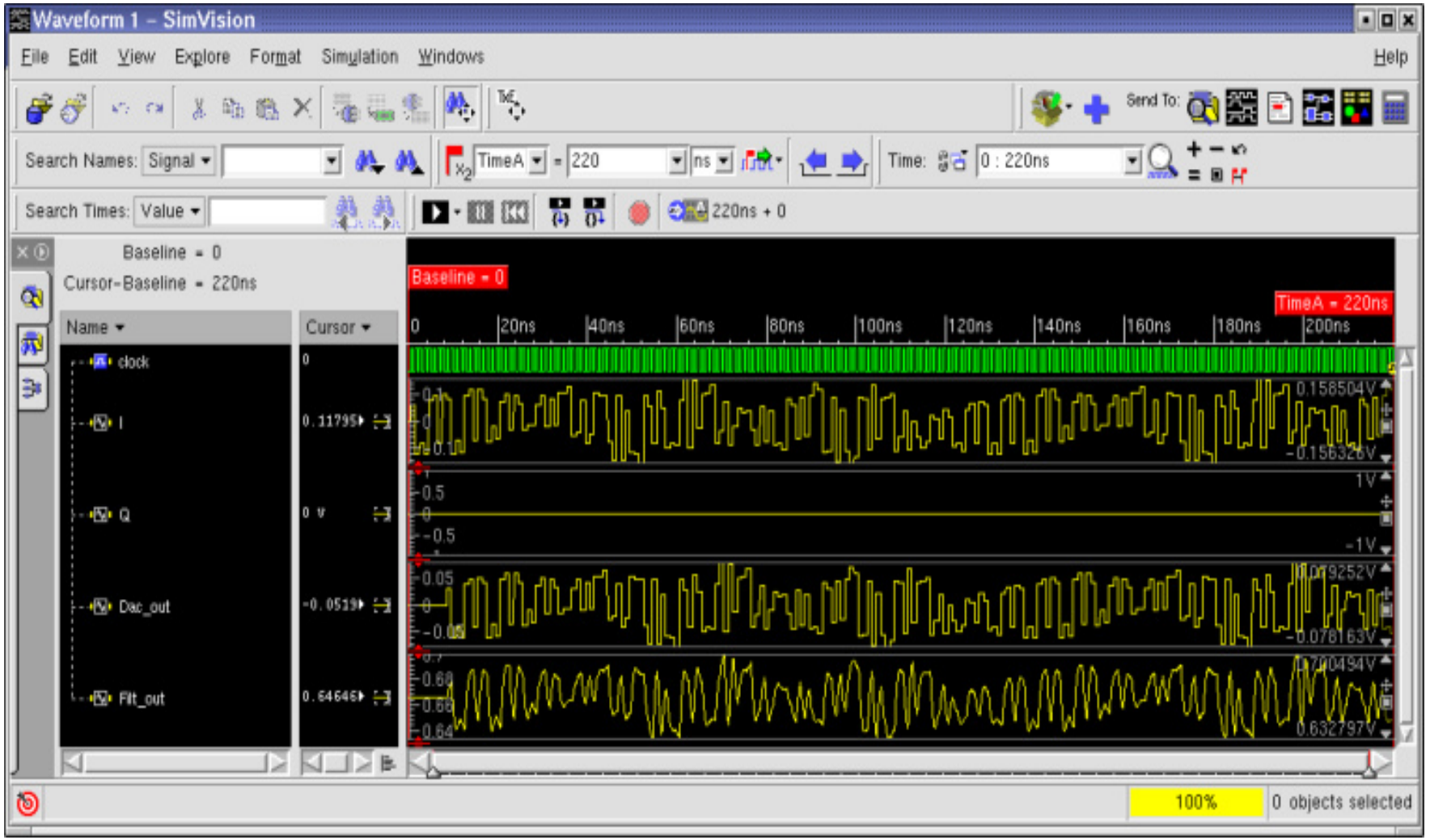


Automated by set of Perl Scripts provided by Cadence



[1] W. Hartong, P. Birrer, SpeAC: Leverage from the System Level Success-Story in IC Design, Cadence Design Systems, MEDEA+ DAC 2006

Simulation results



Comparison of the 2 methods

- Work flow
  - GUI and command line
  - Basic C and SystemC knowledge required
- Accuracy
  - Similar accuracy of results for the 2 methods
- Performance
  - 55296 samples (I and Q)

Method	Memory Usage	CPU Usage
VPI	88.4M	4786.8s
SystemC Import flow	109.0M	10818.9s

Conclusions

- Novel System Design Methodology
  - To bridge functional, architecture and implementation abstraction levels
  - Targeted at the system architect or system integrator
- Unified Simulation Environment for RF, Mixed Signal and RF SoCs
  - For architecture exploration without subsystem over-specification
  - For sub-circuit validation and early verification
- CAD/EDA tool vendor independent solution
  - Verilog Procedural Interface standardized under IEEE 1364
  - Functional blocks based on standardized description languages