Nanometer Wireless Transceiver Modeling using Verilog-AMS and SystemC

Martin Hujer, Radek Manasek, Jerry O'Mahony, Patrick Feerick, Mark Barry, Brendan Walsh
Silicon & Software Systems Ltd.
mark.barry@s3group.com

ABSTRACT

This paper presents an original methodology for use of Verilog-AMS and SystemC to model a highly integrated wireless transceiver. Novel structures allow the model to be reconfigurable for extensive early architectural analysis and easy re-mapping to new wireless standards and applications. Matlab plots are used to demonstrate the usefulness of the model simulations in the analysis of complex combined digital and analog functionality such as automatic gain control and DC offset correction. Performance results for the simulations as well as development effort are also presented thus showing how this methodology is well suited to the modeling of integrated nanometer wireless transceivers.

Keywords

Integrated circuit modeling, Mixed analog-digital integrated circuits, Simulation

1. INTRODUCTION

With the advent of nanometer technologies, more and more analog and digital functionality is being integrated on to a single die [1]. In the area of wireless transceivers, designs are moving towards either System In Package [2] or towards the integration of RF onto CMOS as it begins to perform as well in CMOS as in bipolar [3]. This allows the RF to be integrated with the analog and digital baseband to provide a low-power and cost-effective solution [4]. In addition, since digital circuits scale much better than analog ones with shrinking geometries and are more easily moved to new processes, there is a trend to reduce analog and RF circuitry and compensate with increased digital complexity [5]. Even in what were traditionally pure analog designs, there is a tendency to use complex digital logic for calibration [6]. Given that there is an increasing amount of digital and analog integration on a single die, it is necessary to verify the interaction of these complex systems before fabrication. Waiting until silicon is available to validate system interactions and find bugs would be a costly exercise, with respins in nanometer geometries reaching millions of dollars [7] and time to market being of prime importance.

In this paper we present an original method for co-simulation of analog and digital systems in wireless transceivers using Verilog-AMS and SystemC. This methodology allows rapid prototyping of these systems for early implementation-dependent architectural analysis, fast simulation and an excellent path to implementation with potential reuse for RTL and schematic verification. In section two we describe the previous work related to modeling of mixed analog and digital systems with emphasis on transceivers. In the third section we present a transceiver architecture and our reconfigurable top-level hardware model of this transceiver. In the fourth section we describe the type of analysis that may be obtained from co-simulation, with Matlab plots to capture performance metrics of the transceiver design. In the fifth section we show simulation performance and effort results for the methodology. In the final section we present our conclusions and our experiences with this methodology.

2. PRIOR WORK

The increasing need for co-simulation of analog and digital systems has been approached in different ways. Co-simulation of analog sections with SPICE and digital parts with an HDL would be too simulation-intensive for the large transceivers that we see now. In addition, the transistor schematics and RTL are not available until very late in the design process. At this stage, architectural issues found through co-simulation are more difficult to correct and can have a time-to-market delay impacting revenue streams.

Particularly with wireless transceivers, system design is mostly done with tools such as Matlab and SPW to identify algorithmic issues [8]. However these high level simulations are not good at modeling implementation effects such as signal transients and parasitics and are often far removed from the actual design implementation [8]. There have been attempts to improve upon this gap with implementation by co-simulation of Matlab and SystemC [9], but this is appropriate only for the digital parts of these designs. SystemC-AMS language extensions are being developed by OSCI to allow mixed-signal systems to be modeled more effectively in SystemC [10]. However this

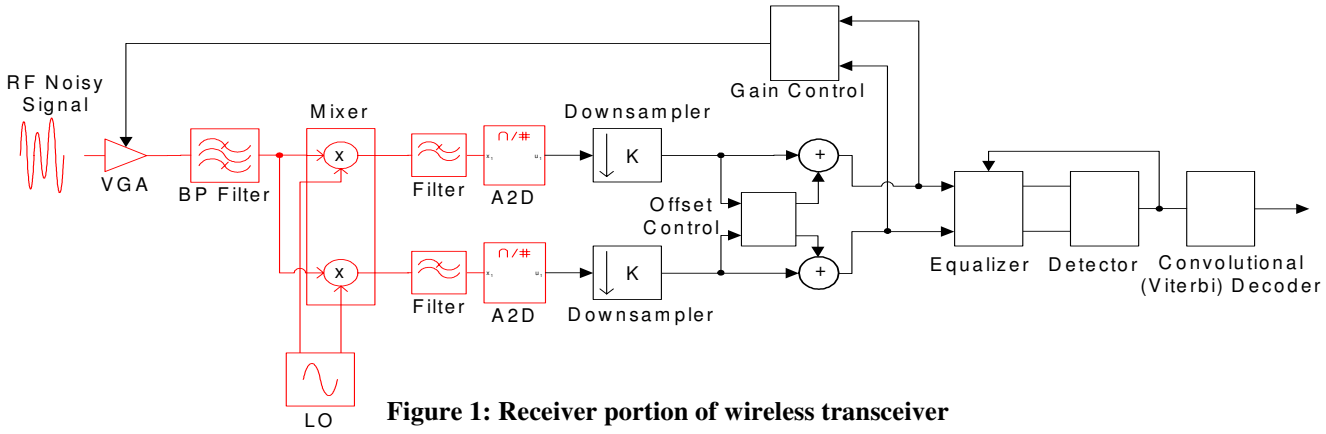

Figure 1: Receiver portion of wireless transceiver

has not been completed and standardized and so is not supported by commercial simulators.

Where large portions of analog or RF have been moved to the digital domain, it is possible to simulate transceivers using an HDL such as VHDL with behavioral models for the analog blocks [11]. For designs where large amounts of functionality are still in the analog domain, as is the case for most transceivers, a combination of Verilog-AMS to model analog blocks and Verilog HDL for digital sections has been used [1]. Even though this methodology may allow reasonably rapid simulation, it is still necessary to wait for digital RTL to be available before co-simulation can occur, again increasing the potential of a time-to-market delay.

In our model we apply an original methodology of using Verilog-AMS to model the RF or mixed-signal components and SystemC for the digital parts. This potentially allows higher simulation speeds than in the previously described methods. More importantly though, it allows co-simulation of the digital and analog portions well in advance of the implementation views being available giving a considerable time-to-market saving. In addition to this, novel structures enable high levels of reconfigurability during architectural exploration.

3. TRANSCEIVER MODELING

3.1 Wireless Transceiver Architecture

To demonstrate our methodology, a wireless transceiver is used as a test case. It is a good example of a system with a very complex analog front end and a large digital section, requiring significant functional interaction between both parts. Although the transmit portion of the design as well as the air interface channel are modeled in our simulations, we will focus on the more interesting receive portion of the transceiver in the analysis sections of the paper.

3.2 Model Requirements

The main requirement of our model was to enable fast co-simulation of complex analog and digital systems allowing early architectural analysis. Another requirement was to use standard languages that were supported by commercial simulators which would allow reuse of the model for verification of the implementation. We also wanted the model to be highly reconfigurable so that we could do extensive architectural tradeoff analysis and to allow it to be easily adapted to various wireless standards.

All simulations were carried out using Cadence AMS Designer. This tool also supports HDL and SPICE simulations through a unified simulator kernel [16], which means that the abstract model can be reused as a test harness and golden reference model for verification of the RTL and transistor-level implementations.

On the receive side everything up to and including the A2D on the left of Figure 1 (the channel model, variable-gain amplifier, band-pass filter, mixer, local oscillator, low-pass filters and ADCs) represent analog and RF blocks that are modeled in Verilog-AMS. All the other post-ADC blocks on the right of Figure 1 are digital blocks and are modeled in SystemC.

Verilog-AMS can model continuous-time signals in the analog blocks and has many analog processing functions available. Because the SystemC-AMS mixed-signal extensions to SystemC are still in development, modeling of analog blocks in SystemC was not considered.

3.3 Verilog-AMS Modeling

We chose Verilog-AMS for modeling the analog parts because it is the language of choice when modeling a mixed-signal system at a high level of abstraction [14], [15]. It allows the designer to create rapidly a top-level infrastructure which is usable throughout the project to verify the connectivity and functionality of the system [17]. Using this infrastructure the transistor level schematic implementation can be calibrated against the Verilog-AMS model. The language is very powerful for representing complex analog behavior including non-idealities such as converter DNL and circuit offset in a concise manner [15]. Thus Verilog-AMS models can be written quickly and help

a designer get simulations running before transistor-level schematic designs have begun. When writing Verilog-AMS models large simulation speed gains can be achieved by writing efficient code [17].

For our model we created highly parameterized Verilog-AMS blocks. Let's take the fifth order Bessel low-pass filter as an example. The starting point was the low-pass transfer function.

$$H(s) = \frac{1}{(1+\frac{s}{p1})(1+\frac{s}{p2})(1+\frac{s}{p3})(1+\frac{s}{p4})(1+\frac{s}{p5})} \quad (1)$$

The poles of the transfer function are represented by *p1*, *p2*, *p3*, *p4* and *p5*. This is modeled in Verilog-AMS using the *laplace_np* function. The coefficients of the *laplace_np* function are defined as parameters.

Listing 1. Low pass filter model

```
module ms_tlm_rxlpfilter ( voutm, voutp, …);
    parameter real f0  = 1000;  // Cutoff frequency
    parameter real rp1 = -1.3851; // Real Pole1
    ...
    analog begin
    ...
      voutVar = laplace_np(V(vinp, vinm),
          {1},
          {rp1*`M_TWO_PI*f0, ip1*`M_TWO_PI*f0,
           ...
           rp5*`M_TWO_PI*f0, ip5*`M_TWO_PI*f0});
      ...
    end
  endmodule
```

The filter cutoff frequency can be changed by passing a single parameter to the block. To change the cutoff frequency of a transistor-level schematic could take several weeks of effort.

3.4 SystemC Modeling

We chose SystemC for modeling the digital parts of the design because it is a standardized abstract language which allows fast creation and simulation of models at high levels of abstraction as well as allowing high reconfigurability [12], [18], [20]. It is excellent for modeling hardware and is well suited for reuse in verifying RTL implementations [13], [19].

We can see this ability to abstract in the syntax used to describe the instantiation and connectivity of components. Rather than use a pin-level description a more abstract and generic syntax can be employed – as shown in the following code example. This has the benefit of being both concise and less impacted by low-level implementation changes later in the design cycle.

Listing 2. Connection syntax excerpt

```
rxgainctrl -> out_port (*analog); // 'abstract' connection
rxequalizer -> out_port (*rxdetector); // Equalizer connection
```

Furthermore, and unlike an HDL environment, these instantiations need not be static but can be deferred to simulation run-time by virtue of the abstract description for device connectivity. This has significant modeling advantages, in that device configurations can quickly be created and explored without actually changing the design code description. For example, as will be seen in the next section, a simple text file can define the configuration.

3.5 Top-Level Model

The top level of the simulation model is written in SystemC. It controls the simulation and contains instances of all the main blocks as shown in the following diagram.

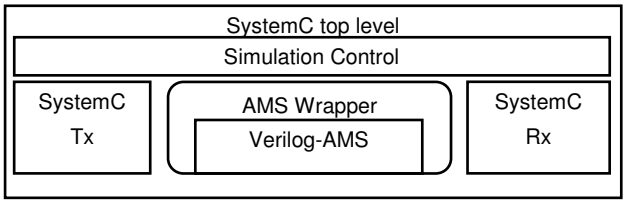

Figure 2: Simulation environment hierarchy

Configuration occurs at simulation startup. A configuration file is read and this is used to replace the generic instances with specific architectures as shown in the code segment below:

Listing 3. Configuration excerpt

```
while (config_file >> field) { // read text file
  if (field == "START_RXDETECTOR"){
    config_file >> field;
    if (field == "16QAM")
      rxdetector = new ms_tlm_rxdetector_16qam
                  ("DETECTOR",config_file,
                   global_parameter_reader);
    else if (field == "…")
      … // alternative detector scheme
  }
}
```

The analog part is positioned in the system via a SystemC wrapper that ensures the right binding of Verilog-AMS signals ports and parameters to SystemC variables. This

module is also configurable as the Verilog-AMS parameters are defined in the configuration file and passed in through the wrapper interface. The essential code for the wrapper is shown below.

Listing 4. SystemC wrapper

```
class ms_tlm_analog : public ncsc_foreign_module
   public:
{
   sc_in < bool> sampling_clk;
   sc_in < sc_lv <4> > modulator_data;
   …
   ms_tlm_analog(
         sc_module_name nm,
         double carrier_freq_mhz,
         …
   ) : ncsc_foreign_module(nm),
         sampling_clk ("sampling_clk"),
         modulator_data ("modulator_data"),
         …
   {
   ncsc_set_hdl_param( // pass in ams parameter
         "CARRIER_FREQ",1000000*carrier_freq_mhz);
         …
         }
}
```

A section from a typical configuration file is shown below. It illustrates how analog parameters such as carrier frequency, signal-to-noise ratio, cutoff frequencies of filters and various architectures for blocks can be specified. The digital blocks are also highly configurable, such as choice of Viterbi algorithm and the length of its window. Other settings include bit widths of variables and the total simulation time.

Listing 5. Configuration file excerpt

```
START_ANALOG
AMS_TRANSACTOR
CARRIER_FREQ_MHZ 200 // parameter
CHANNEL_SNR 20
…
RX_PGA_CUTOFF_FREQ_MHZ 3000
…
START_RXDECODER
TYPE_A // new architecture
L 8
```

Matlab can be launched directly from SystemC allowing results to be processed as the simulation runs. An example would be successive plotting of signal points into the constellation diagram.

4. ANALYSIS

The model can be used to generate waveforms for functional analysis of the system. Figure 3 shows signal waveforms from the model for the low pass filter and ADC. The interaction between the analog and digital sections can be observed and analyzed by the designer.

The rest of this section concentrates on the interaction between the analog and digital domains. This task is the most important feature of the simulation, as without co-simulation of implementation-specific effects such as signal transients and block delays, you cannot be sure that your combined analog and digital systems function correctly.

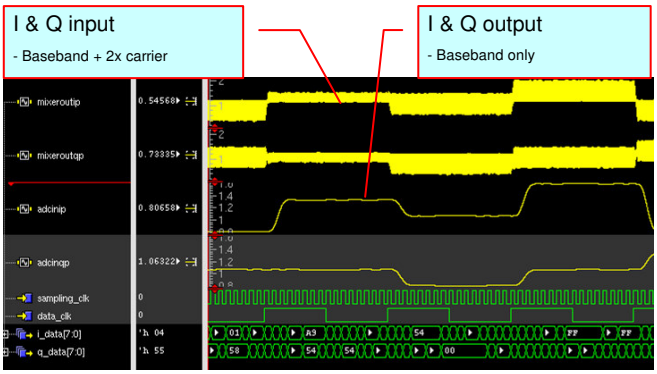

Figure 3: Snapshot of waveforms

Simulation of the automatic gain-control loop involves both the Verilog-AMS and SystemC models. The gain controller evaluates the power of the received signal. The actual values are averaged and then compared with a defined threshold. If the estimated power is low, the gain of the amplifier at the input of the receiver is increased. If the estimated power is too great, the gain is decreased.

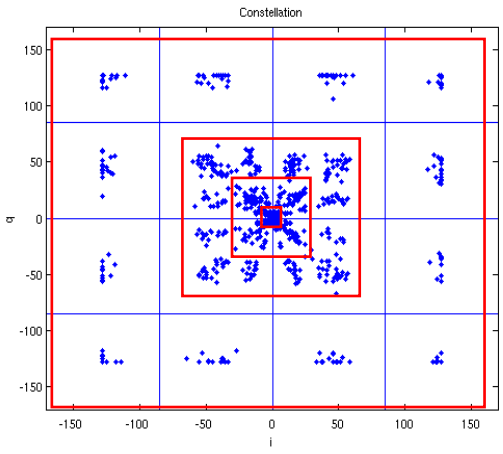

Figure 4: Progress of constellation with AGC loop

Figure 4 illustrates the impact on the point locations in the constellation. Their shift from the center to their correct

quadrant positions is clearly seen, showing that the basic function of the gain control is working as expected.

DC offset is a source of error in the received signal caused in part by implementation effects such as circuit offset in the analog blocks. The offset controller estimates the current DC offset of the in-phase and quadrature components of the received signal. It then generates the value which is removed from the baseband signal, leaving a signal without DC offset. Normally this DC offset is removed in the analog domain prior to the ADC, but in our system we decided to test our methodology by removing the offset in the digital domain.

Figure 5 illustrates the progression over time of the estimated offset which, as expected, begins to converge to a relatively stable offset after a certain time, showing that our offset calculation is working in principle.

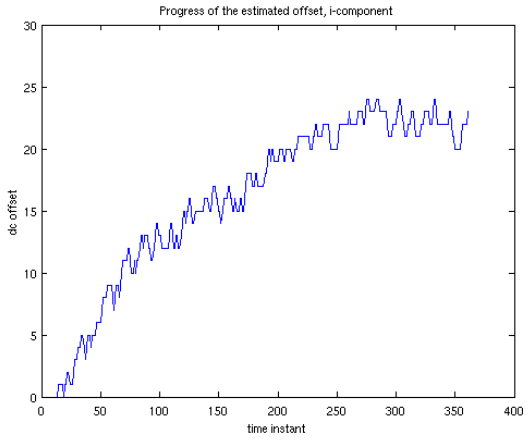

Figure 5: Progress of offset estimation

Figure 6 shows the constellation of the received signal. There is a notable shift of signal point locations over time towards their correct positions at the center in most cases. However, it can be seen that in the right-hand side of the constellation the signal actually moves away from its correct central location. This clearly indicates that the ADC is saturating. This arises because the gain is estimated from the signal after offset removal, but the offset is not removed before the ADC. The resultant saturation of the ADC also causes the convergence of the offset removal towards a non ideal point.

We deliberately remove the DC offset in the digital domain to test our methodology and we can see the effects of this, which would have a detrimental impact on the quality of the final received signal. It is thus clear that the methodology can be used to carry out architectural explorations and optimizations.

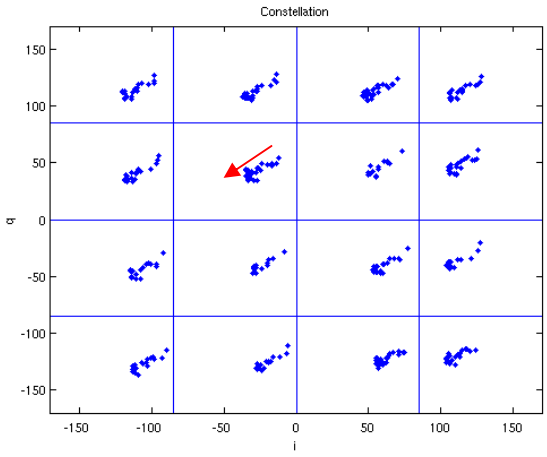

Figure 6: Progress of constellation with offset adjusted

5. RESULTS

Although simulation speed is not the most important benefit of this methodology, performance results are presented in this section along with development effort. All simulations were run on a Red Hat Enterprise Linux 3 machine with dual 64 bit 2.6GHz AMD Opteron™ processors and 7.6GB of memory.

Table 1 shows the almost linear relationship between CPU time and analysis time, corresponding to simulated transceiver time in which the designer is interested. It can also be seen from Table 1 that there is a strong relationship between carrier frequency of the transceiver RF and the simulation time, which is expected due to the larger number of simulation points needed within a certain transient analysis time. However the simulations are able to complete in a reasonable amount of time even for RF of 2.4GHz. This is important as CMOS RF circuits are now able to operate in the GHz range.

Carrier Frequency	Analysis Time	CPU Time
200MHz	10us	29s
2.4GHz	10us	171s
2.4GHz	50us	815s

Table 1: CPU times for run times and frequencies

In the simulations that we have run, we also found that in all cases the simulation time spent on the Verilog-AMS blocks and the interface between Verilog-AMS and SystemC consumed more than 99% of the CPU time. This gives a good indication that without significantly impacting CPU time additional digital complexity could be added to the simulation such as including the transceiver Medium Access Controller (MAC) and processors running embedded software. This would allow a transceiver system

designer to simulate the entire system with this methodology if desired.

In previous publications we have shown the benefits in terms of simulation and development time of Verilog-AMS versus transistor schematics and SystemC versus RTL [12], [17]. With this transceiver model we see that these simulation speed benefits allow complex analysis as was shown in section five. More importantly though, we were able to create the model within 2 months of development time while we would have been waiting 6 to eight months for initial RTL and transistor schematics. This allowed us to complete the architectural analysis much earlier than would have been possible if we had to wait for implementation to complete. In addition the models can be re-used for calibration and verification of the implementation. These factors can be translated into a risk reduction and potential time-to-market saving for transceiver programs.

Our novel code structures allowed the entire transceiver to be reconfigured within moments. So for example we were able to change the analog anti-aliasing filter cut-off frequency and number of digital equalizer taps by changing 2 lines in the configuration file instead of waiting weeks or months for new RTL and transistor level implementations.

6. CONCLUSION

The original methodology described allows co-simulation of complex analog and digital systems, particularly those found in highly integrated wireless transceivers. Abstract models in Industry-standard languages SystemC and Verilog-AMS can be reused for verification of the actual implementation RTL and transistor-level schematics. Models can be quickly written and simulated allowing analysis of implementation effects in mixed analog-digital systems to begin before schematics or RTL are available. Moreover, the models are highly configurable, allowing extensive early trade-off analysis leading to robust architectural definition.

REFERENCES

[1] A. Matsuzawa, "Mixed signal SoC: A new technology driver in LSI industry" Keynote speech, IEEE International Symposium on Circuits and Systems 2003

[2] M. Fitzgibbon, J. Mondal, D. O'Keeffe, D. McSwiney, D. Redmond and W. Waldie “A Single Package Transceiver for Quad Band EGPRS (GSM/GPRS/EDGE) Class 12 Applications,” IEEE European Conference on Circuit Theory and Design, 2005

[3] A. Abibi, “RF CMOS comes of age,” IEEE Journal of Solid State Circuits, Vol. 39, No. 4, April 2004

[4] C. Paillard, J. Wright, “RF Architectures: Past, Present and Future” Wireless Networking Seminar, Embedded Systems Conference, 2006

[5] R. Staszewski, J. Wallberg, S. Rezeq, CM Hung, O. Eliezer, S. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M-C. Lee, P. Cruise, M. Entezari, K. Muhammad and D. Leipold “All-digital PLL and GSM/EDGE transmitter in 90nm CMOS,” IEEE Solid-State Circuits Conference, 2005

[6] Y. Cong and R. Geiger “A 1.5V 14b 100MS/s self-calibrated DAC,” IEEE Solid-State Circuits Conference, 2003

[7] D. Harris, “VLSI Economics,” Lecture at Harvey Mudd College, Spring 2005

[8] M. OBrien, P. Pratt, “The Application of MathWorks tools in the Architecting of Mobile Phone Radio Frequency ICs,” Model Based Design Conference 2005

[9] J.F. Boland, M. Hemon, C.Thibeault, Z. Zilic “Using MATLAB and Simulink in a SystemC verification environment,” North American SystemC Users Group Meeting, 2004

[10] A.Vachoux, C. Grimm, K. Einwich “SystemC-AMS Requirements, Design Objectives and Rationale,” IEEE/ACM Design Automation and Test in Europe Conference, 2003

[11] R.-B. Stazewski, K. Muhammad and D. Leipold “Digital RF processor (DRP) for cellular phones,” IEEE/ACM International Conference on Computer Aided Design, 2005

[12] M. Barry, P. Feerick, B.Walsh, “Architectural analysis of an eDRAM arbitration scheme using Transaction Level Modeling in SystemC,” CDNLive Silicon Valley, 2005

[13] D. Mitchell and D. Notestein, “Creating SystemC and HDL testbenches with SCV,” Chip Design Magazine, May 2004

[14] K. Kundert and O. Zinke, “The Designer’s Guide to Verilog AMS,” published by Kluwer Academic Publishers

[15] Verilog-AMS Language Reference Manual, Accellera, Available at http://www.designers-guide.org/VerilogAMS

[16] Cadence AMS designer simulator datasheet, available from http://www.cadence.com/products/custom_ic/ams_designer/

[17] T. Blake and D. Breathnach, “Design and Verification of Nanometer SoCs using AMS Designer,” CDNLive Silicon Valley, 2005

[18] F. Ghenassia, “Transaction-Level Modeling with SystemC,” Published by Springer, 2005

[19] A. Clouard, “TLM concepts that are successful at ST,” ECSI Institute workshop on Efficient Transaction Level Modeling, June 2005

[20] OSCI Standard for SystemC TLM, available from http://www.systemc.org