Get to ASICs Faster - A Novel Mixed Signal Design Methodology

Dr. Greg Tumbush[1], William Gonnason[2], Dustin Griesdorf[3], Alaa El-agha[3],
Gareth Weale[3], Marc Matthey[3], Andreas Drollinger[4], Holger Meiners[5]

(1) Tumbush Enterprises, Colorado Springs, CO/USA, greg@tumbush.com
(2) ON Semiconductor, Colorado Springs, CO/USA, william_gonnason@amis.com
(3) ON Semiconductor, Waterloo, Ontario/Canada, {alaa_el-agha, dustin_griesdorf, gareth_weale}@amis.com
(4) ON Semiconductor, Marin, Neuchâtel/Switzerland, {marc_matthey, andreas_drollinger}@amis.com
(5) Self, holger_meiners@yahoo.de

***Abstract*-This paper introduces a novel methodology used to design the critical RF Subsystem ASIC of a wireless hearing aid. A single, high level model of the entire mixed signal system is developed which creates an "executable specification" of both the digital and analog functionality. The model can be simulated to drive architecture development as well as software development. The model is further leveraged throughout the design process for design, verification and FPGA validation. This allowed a small international team to go from architecture to GDSII for the ASIC in less than 4 months. The methodology presented applies to any mixed signal system as well as pure analog or digital.**

I. INTRODUCTION

Attempts to create a system level model of a mixed-signal system utilize a number of methodologies. The tradeoffs with each methodology involve model development effort, accuracy, tool cost, and simulation speed[1][2]. Another very important consideration is the ease of architecture exploration.

The digital portion of the system is typically represented in either Verilog[3] or Vhdl[4] and the analog design is represented as transistor schematics or Verilog-AMS[5] or VHDL-AMS[6]. Less accurate analog models may also be created in Verilog or Vhdl.

At one end of the simulation speed spectrum is representing the entire system in Verilog or Vhdl. The mixed signal system is simulated in a digital simulator which is very fast and inexpensive compared to mixed signal simulators. However, the trade-off is poor accuracy and the difficulty of creating Verilog or Vhdl models of analog circuits.

At the other end of the simulation speed spectrum is representing the digital portion in Verilog or Vhdl and the analog models in transistor schematics. The system is simulated in a mixed signal simulator which is quite slow and expensive. With this methodology the tradeoff is great accuracy and elimination of the effort to create high level analog models. One point that should not be overlooked is the analog schematics simulated are the final schematics used for layout so no further translation is required.

In the middle of these two methodologies lies representing the digital portion in Verilog or Vhdl and the analog design in VHDL-AMS or Verilog-AMS. A costly mixed signal simulator is still required but the simulations will be much faster due to the analog models being higher level. Also, the accuracy is high because the analog circuit can be modeled accurately. However, modeling the analog in VHDL-AMS or Verilog-AMS and verifying that they match the final schematics is a large effort.

Architecture exploration is very difficult with any of these methodologies due to the high cost of creating new circuits.

In the sections that follow a novel methodology to design mixed signal systems is presented. This methodology improves on the current state of the art in a number of ways. First, it represents the RF portion with both circuit level and system level analog models and the digital portion with SystemC. A mixed signal simulator is then used to simulate the entire system. The mixed signal simulator and RF design environment chosen was the Agilent Advanced Design System, or ADS, but this capability is available from other vendors. The ADS and SystemC models are easy to create and the design can be simulated quickly for true end-to-end simulations. In this way, architecture exploration is possible earlier in the design process.

First, we introduce the system that this novel mixed signal design methodology was applied to.

A. The RF Subsystem ASIC

The RF Subsystem is an integral part of a wireless hearing aid product. The RF Subsystem is a stand alone RF

ASIC/ASSP that forms the RF Subsystem Chip. See Figure 1 for a partial breakdown of the System. The final decomposition of the Digital Subsystem and the Analog RF Front End can be seen in Figure 2 and Figure 3 respectively.

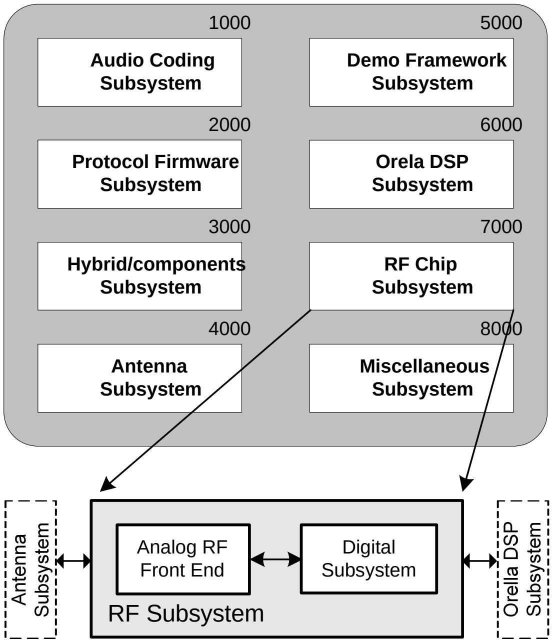

Figure 1: Wireless System

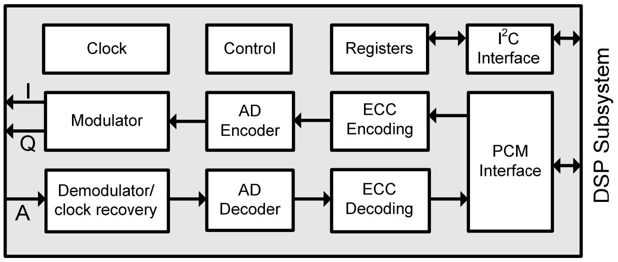

Figure 2: Digital Subsystem

The Digital Subsystem transmits and receives data from the DSP to the analog portion of the RF Subsystem. In transmit mode it accepts audio data from the DSP via the PCM interface, performs channel encoding, and digitally generates an FSK modulated signal composed of In-Phase (I) and Quadrature (Q) components. In receive mode it receives the limited IF signal, performs channel decoding, and delivers the baseband data with the appropriate clock to the DSP Subsystem via the PCM interface. The DSP Subsystem reads and writes data through the I^{2}C interface which it uses in addition to I/O lines to set the ASIC to power-down, standby, or active mode.

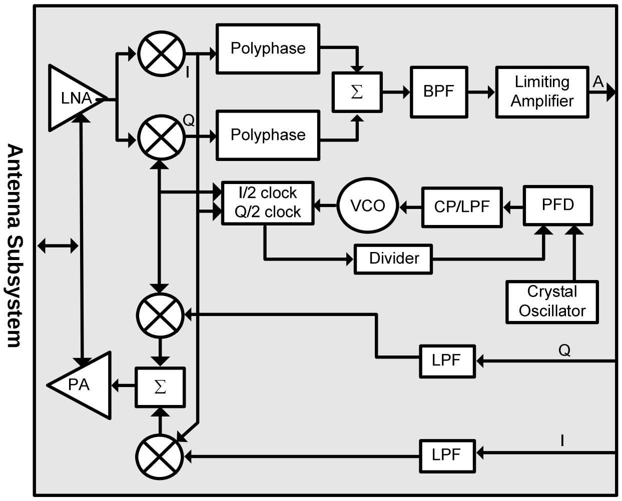

Figure 3: Analog RF Front End

The Analog RF Front End fundamentally transmits and receives the RF signal. It utilizes quadrature mixing architecture for both transmitter and receiver. On the transmit side, it accepts the digitally FSK modulated I and Q waveforms from the MODEM module. The I and Q waveforms are filtered by low pass filters, up-converted by quadrature mixers, and then amplified by the Power Amplifier (PA) to produce the RF signal. On the receive side, the received RF signal is amplified by a Low Noise Amplifier (LNA), and down converted to IF by quadrature mixers. The resulting I and Q signals are passed through poly-phase filters before being combined to produce a single IF signal. The IF signal is then filtered by a band-pass filter which is amplified and limited. The output of the limiter is an FSK modulated square waveform version of the transmitted signal. The RF Front-End delivers this square waveform to the MODEM module for demodulation and further processing. Table 1 shows a summary of the high level system specifications.

Table 1: High Level System Specifications

Parameter	**Specification**
Current Consumption	Less than 1.5mA
Range	At least 40cm
Data rate	At least 128kbps
Modulation	Binary FSK
RF BW	300kHz
BER	10^{-5} effective
Size	Fits in BTE and ITE hearing aids

B. The Problem

The team was tasked with completion of the RF Subsystem ASIC, including layout, in 4 months. This was a hard constraint because of limited shuttle runs and a customer requirement to demo a working system at the next American Academy of Audiology (AAA) Convention & Expo. The RF System ASIC is based on an earlier test chip but its functionality has been greatly expanded. Except for a PCM and I^{2}C interface all blocks were original designs.

A major challenge facing the team was the fact that the system requirements, as set by the customer, were only defined at a very high level. Other than current consumption, distance and data rate, nothing was defined. The team was tasked to define the architecture for the whole system that will not only meet these requirements, but also investigate several options to make the RF chip a good fit for wireless hearing aid applications. For example, for wireless devices, link distance is set by defining the maximum acceptable BER. However, raw RF link BER is not sufficient since error detection and correction, ECC, could be used to increase the distance through coding gain. This issue is more significant in wireless hearing aids than in normal wireless applications as the ECC coding scheme does have an impact on audio quality measures for hearing aid algorithms (i.e. BER and delay). That is, an ECC code should be designed or chosen in the context of hearing aids applications. Another question that needed an answer early on was what RF front-end architecture should be used that fits the above three requirements, yet is low risk. For example, a direct VCO FSK modulator is a higher risk architecture than an IQ-based FSK modulator. To complicate the case further, the FSK modulator along with the frame synchronization scheme were new concepts and needed to be designed from the ground up and evaluated.

The above examples simply mean that the team was faced with the challenging task of doing architecture definition, system design, specifications development, RF/analog and digital chip design and verification in a four month period.

C. *The Solution*

We realized early-on that a traditional design flow as detailed in Figure 4 would not get us to tape-out on schedule. There are two problems with this flow as pertains to this project. First, this flow assumes that at least the system architecture is defined and one can proceed with developing detailed block specifications. Second, the system cannot be fully evaluated until all digital logic is designed in VHDL/Verilog (i.e. RTL) and the analog schematics are complete. If problems are discovered the iteration loop is huge and the schedule impact is immense. We abandoned this flow in favor of the design flow in Figure 5. This flow is unique in that the architecture was created using a high level modeling environment.

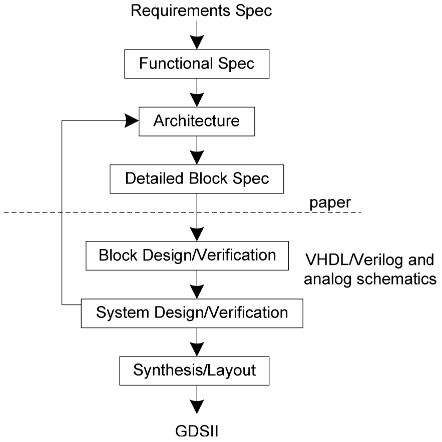

Figure 4: Traditional Design Flow

SystemC was chosen to model the digital subsystem while the Agilent Advanced Design System, or ADS, was chosen to model the analog RF Front End. These two high level modeling environments allow us to achieve two goals. One is to enable quick "what if" scenarios to make informative decisions regarding both the RF/analog front-end and digital baseband architectures. The other is to create an "executable spec" to evaluate the system architecture. Data is processed through the system model and the output evaluated. The architecture is refined if necessary. In this way, if problems are discovered, the schedule penalty is small. Many costly design iterations were eliminated using this methodology.

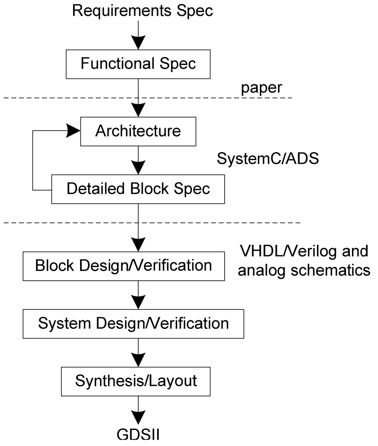

Figure 5: Design Flow using High Level Models

In this paper the SystemC and ADS modeling languages used in the RF Subsystem are described and how they were leveraged to create and verify an ASIC. Furthermore, the tradeoffs made in creating the analog and digital models are explored to illustrate where detailed modeling makes sense and where it does not.

This methodology can also be applied to entire systems like more complex RF chips and others consisting of multiple ASICs, processors, firmware, and software tools. Software tools, firmware design, and testing can begin as soon as the first SystemC/ADS executable spec is complete. In a traditional design flow the software tools and firmware are designed from a paper specification and the results of these tasks can only be tested when an FPGA or ASIC is available.

II. AN EXECUTABLE SPEC OF THE RF SUBSYSTEM

In this section the use of SystemC and ADS for system design to create an “executable spec” is explored further.

A. What is SystemC?

SystemC is simply a class library that can be included in a C++ program[7]. SystemC provides hardware centric data-types such as fixed width variables, 4-valued logic, and operators such as concatenation, bit selection, and bit insertion. While possible in native C++ this functionality is difficult to implement. SystemC also offers the notion of concurrency and time, essential to designing hardware systems. A SystemC design can vary in abstraction from register transfer level (RTL) to a very high level behavioral description. The SystemC Verification library (SCV) provides advanced verification features such as generation of constrained random stimulus.

B. What is ADS?

ADS is a software tool by Agilent Corp for designing RF ICs. It supports both circuit level and system level simulations and modeling, with co-simulation capabilities to mix both circuit level and system level (behavioral) models of system blocks in the same simulation run. In addition, it is capable of RF/Analog and digital co-simulations.

Since the version of ADS that we used did not yet support intrinsic co-simulation with SystemC, co-simulation between ADS and SystemC was accomplished through file I/O. However, a new version of ADS has been released that removes this requirement to provide true end-to-end simulation capability.

C. SystemC/ADS Create an Executable Spec

SystemC provided an ideal environment for developing the system architecture for the digital section of the RF Subsystem. Critical system components such as the modulation technique, forward error correction coding, demodulation and frame and clock recovery were designed without prior designs to work from. SystemC provided a means to quickly prototype these blocks at the functional level without dealing with the intricacies of RTL. Each block was then evaluated for function and performance either as part of the system or independently. Since SystemC is based on a standard C++ run-time environment a simple test bench was created that allowed testing various configurations by simply changing command line parameters. Internal state and diagnostic information useful in evaluating the design were easily output to either text files or the screen for analysis. Further, based on the results of the functional and performance testing, the blocks were quickly refined as required and retested to optimize the design.

SystemC allowed the design to be built using a modular approach. Specific modules could be built independent of preceding or following modules and later integrated into the larger project. This modular approach minimized the inter-process dependencies and sped up the design process. For example, once a SystemC model of the modulator was complete, even though other SystemC models preceding the modulator were still under development, work could start on the ADS model that used the modulator as its input.

Later in the design process SystemC proved advantageous in that the functional models could be refined to bit and cycle true models providing a clear path towards the RF Subsystem RTL implementation. Bit and cycle true models also allowed for easy co-simulation between the SystemC and RTL.

ADS was used primarily for two purposes. First, system level specifications for the RF/analog front-end were derived and verified in the ADS environment. Among these specifications are link budget, RF blocks performance requirements like LNA and PA gains, etc., PLL startup time, maximum acceptable PLL synthesizer phase noise, PLL stability, PLL loop filter requirements. The second purpose is to use the developed model as a test bench for the SystemC model of the digital section. Knowing that modulation and demodulation are digitally implemented, with further ECC encoding/decoding, one is only able to derive the front-end performance through a true end-to-end performance. This was accomplished by interfacing the executable SystemC model of the digital section (which includes modulator/demodulator, ECC encoding/decoding, etc.) with the front-end model in ADS. This allowed RF non-idealities to be investigated and specifications of both front-end and digital to be refined. An example that helps illustrate the usefulness of this use model is evaluating different PLL synthesizer phase noise levels on the end-to-end system BER using a model for the actual digital modem design.

In the ADS environment, the simulators that were used in the RF Subsystem are Ptolemy, Circuit-Envelope, Harmonic Balance, and Transient simulators. The combined use of Ptolemy and Circuit-Envelope simulators made it possible to simulate the end-to-end performance of RF Subsystem including PLL phase noise, AWGN and system non-idealities in a quick manner. In addition, simulating the PLL startup time and stability including phase noise effects was only possible due to the Circuit-Envelope simulator. On the transmitter side, the SystemC model saved the modulated I and Q waveforms into a file, which was read by ADS. The I and Q signals then go through the RF front-end inside ADS all the way to the output of the Limiting Amplifier on the receiver side. ADS will then save the Limiting Amplifier output into a file that is later read by the demodulator SystemC model for further processing. Figure 6 shows a screen shot of the ADS model.

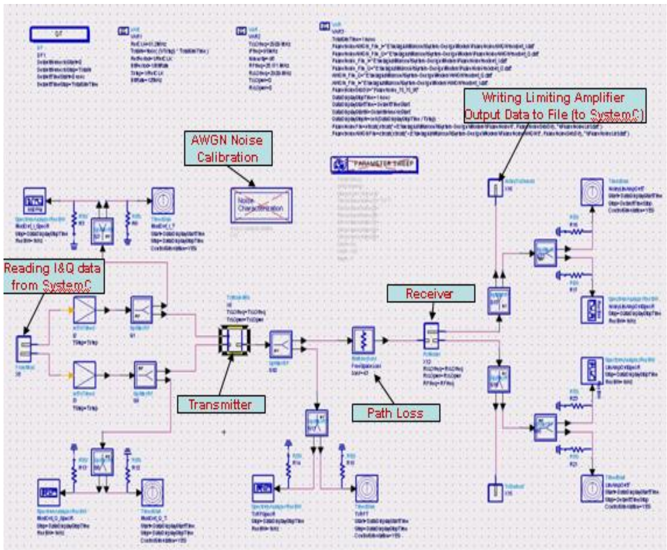
AWGN Noise Calibration
Writing Limiting Amplifier Output Data to File (to SystemC)
Reading I&Q data from SystemC
Receiver
Transmitter
Path Loss

Figure 6: ADS RF Subsystem Model

D. *Executable Spec Refinement*

SystemC was used to create the digital subsystem from the initial concept to a final cycle accurate design. The initial concepts had to be proven so the first versions consisted of proof-of-concept designs. These designs consisted only of top-level functionality such as an error correcting code and synchronous data flow. The design was continually refined based on analysis of the executable specification output. The transmitter output included I/Q waveforms that were analyzed for their spectrum characteristics. The receiver output included the received data as well as information related to receiver frame synchronization and clock recovery. The received data was compared against the transmitted data to compute a bit error rate (BER).

The SystemC model was driven with various test vectors such as simulated bit errors and output from the ADS model. As the ADS model was being continually refined the output was being fed into the SystemC model to evaluate the digital performance based on the latest designs.

First, refinements to the SystemC design were made to ensure that the digital subsystem functionality was correct. For instance, the demodulator technique was refined to ensure that the correct data was demodulated under various ADS simulation conditions (such as RF noise and phase noise). Second, further refinements to the SystemC design were made to improve the BER performance. These refinements included adding averaging and rounding to the demodulator, selecting proper error correction coding, improving the clock recovery technique, and adding retry thresholds when acquiring a signal lock.

Finally each SystemC component that may have been designed and tested independently was refined to be integrated into the larger test project.

III. ASIC DESIGN USING A HIGH LEVEL MODEL

In this section the high level model expressed in SystemC and the ADS tool are leveraged for design of the ASIC.

A. *Using a SystemC Model for Digital Design*

Since the SystemC model of the system is the specification, the digital design can be developed directly from the SystemC code. This removes the vagueness that is inherent in a paper specification. Since the specification is executable, because it is just C++ code, both the RTL and SystemC description can be co-simulated in any commercial simulator.

B. *Using a ADS Model for Analog Design*

As the system and circuit designs are intertwined processes, the ADS model was often used during circuit development to refine the specifications, for example, tightening the specifications on one block and relaxing the specifications on another block such that the overall system performance (i.e. BER) is unaffected. In other situations where the circuit-level simulation time would be impractical, the ADS model was used. An example is the impact of VCO and PLL phase noise on the output of the limiting amplifier. In other cases, it was not possible to investigate the performance of certain blocks in the circuit-level environment. An example is investigating the PLL loop stability under various conditions like phase noise, actual VCO gain and loop filter variations. In such cases, the ADS model was used to investigate these issues.

The ADS model was also used to characterize the effects of VCO phase noise on the detection and also noise performance of the front end based on simulated noise performance of the blocks such as LNA and mixer. The effects of the VCO noise cannot be simulated in the Cadence environment because of the length of simulation time required, but ADS had that capability. The results showed the noise margin we have at the Limiting Amp output and the insensitivity to Phase Noise on the VCO, which may ultimately allow us to use an on-chip VCO tank for the 900MHz version.

C. *Appropriate Abstraction for a Model*

When developing models of a system a major question to be answered is the appropriate level of abstraction. High level models are simple to write and execute very quickly, possibly in real time, but they may not be sufficiently accurate. On the other extreme, a very low level model is a very good approximation of the system but the creation is similar in effort and time to the target design. Note that mixed levels of abstraction can co-exist in the same system simulator. Blocks requiring preciseness can be modeled at a low level while others can be modeled at a higher level.

The SystemC model of the digital subsystem modeled the encoding, decoding, and modem (modulator and demodulator) blocks at a very precise level. The PCM serial interface was abstracted to an interface where 32-bit words were simply read or written using a file. Input to the demodulator was read serially from a file. Similarly, the in-phase output only of the modulator was written to a file. System setup information, normally performed by the I^2C interface and stored in the register block, is passed in as command line arguments to the simulator. Due to the tight schedule, power-down and standby modes were not modeled. Control and clocking were not distinct blocks but inherent in the program flow.

D. *Refinement of the High Level Model*

The SystemC model was refined based on feedback from the RTL designers. The SystemC model abstracts many details about the required hardware to implement the functionality. Feedback regarding logic optimization was received. Changes that were agreed upon based on the feedback were integrated in the SystemC model and retested.

As the circuit-level design of the RF front-end was developed, the RF front-end model was refined based on feedback from the circuit designers. Such refinements included link budget redistribution, actual mixers' nonlinear characteristics, LPF and BPF characteristics, PLL characteristics like VCO gain and loop filter. In fact, the behavioral models of all filters including Tx LPFs, Rx BPF and PLL loop filters were replaced by their circuit schematics. ADS proved to be very useful in such gradual mix of behavioral and circuit level modeling. The decision regarding which blocks should be replaced by circuit models is dependent on a balance between confidence level in the model and simulation speed. However, for other behavioral blocks, refinements can always be made in terms of block specifications, i.e. IP2, IP3, S-parameters and port noise for a mixer. As part of circuit design, designers need to measure such parameters for RF/analog blocks in their simulation environment. These parameters are then fed back to system designers to refine the front-end models

IV. ASIC VERIFICATION USING A HIGH LEVEL MODEL

In this section the high level model expressed in SystemC and the ADS tool are leveraged for verification of the ASIC.

A. *Digital Subsystem Verification with a SystemC Model*

Since the high level model is executable a *co-simulation* environment can be developed that simulates the RTL and SystemC together in parallel. A stimulus generator drives the inputs of the SystemC and RTL simultaneously. The correctness of the RTL can be determined by comparing the response of the SystemC to the response of the RTL. See Figure 7 for a conceptual depiction of the verification environment. Using this methodology moves the verification emphasis from creating self checking tests and verifying results to generating more tests and corner cases, resulting in a better verified ASIC.

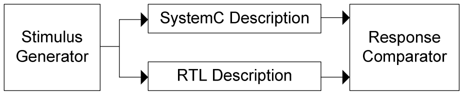

Figure 7: Co-Simulation Environment

B. *Analog RF Front-End Verification with an ADS Model*

From a user point of view, the BER performance of any communications system is the ultimate measure of the system performance. Having a complete system model that allows early verification of the true end-to-end system performance should be appreciated. Such a capability allows early verification of the system performance. This becomes even more crucial for the RF/analog front-end, as it enables capturing design errors and making informative design decisions that would otherwise require another costly tape-out.

For the RF Subsystem, with the refined RF/analog front-end model, end-to-end BER was simulated which verified that the front-end performance was satisfactory. Figure 8 shows the simulated end-to-end BER performance under AWGN noise and different phase noise levels. Figure 9 shows the startup/lock time of the PLL synthesizer using the actual loop filter, VCO gain curve, PFD/Divider/Charge Pump and phase noise.

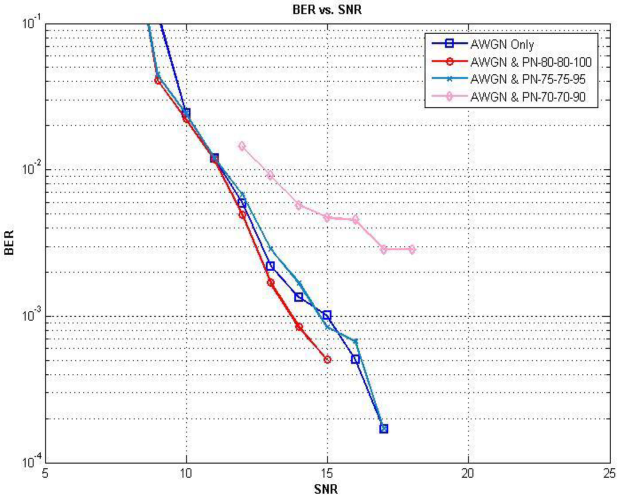

Figure 8: RF Subsystem End-To-End BER vs SNR

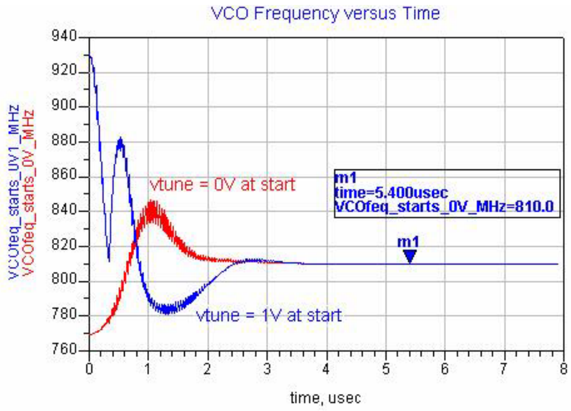

Figure 9: PLL Synthesizer Startup Time

C. RF System Verification

Three different methods were utilized to verify top level Digital Subsystem functionality. An FPGA, a digital co-simulation environment, and a digital full-chip simulation with analog and pad models were utilized. Each of these methods targets different classes of bugs.

The FPGA consisted of an FPGA motherboard and an FPGA daughterboard. The FPGA motherboard was an off-the-shelf product that provided several I/O headers to connect to expansion boards. An FPGA daughterboard of the system was designed consisting of two distinct and independent signal paths – one for the transmitter and one for the receiver. Each path contained an oscillator, DSP, I^2C bus, level translators and LEDs. The FPGA was loaded with synthesized RTL. The DSPs were loaded with firmware previously developed using the high-level SystemC models. The FPGA setup was able to run in real-time. The FPGA implemented two RF Sybsystem designs – one for transmitter and one for receiver. The connection between the two designs was made through a header. A jumper connected the modulated output from the transmitter to the demodulator input on the receiver.

The FPGA was able to provide real-time audio pass through by using the DSPs on the daughterboard. Audio was input to the transmitter DSP. The firmware transferred the audio signal to the RF Subsystem transmitter instance over the PCM interface. The RF Subsystem transmitter processed the data and produced a modulated signal. The RF Subsystem receiver instance used the modulated signal as an input. The demodulator demodulated the signal and processed the data, providing a decoded data stream to the DSP over the PCM interface. The receiver DSP received the data stream through the firmware and played the audio on the audio output.

The digital subsystem was evaluated to ensure that the external interfaces such as the PCM interface worked properly with DSP. The internal components such as the demodulator were verified by ensuring that audio was being output properly from the receiver DSP. By disconnecting the jumper between the modulator and demodulator the design was verified to ensure receiver signal lock could be reestablished after being lost.

The FPGA was also used to test system performance under non-ideal conditions. A jitter generation unit was developed in SystemC and then implemented in RTL for the FPGA. The jitter generator was situated between the modulator and demodulator. When active, pseudo-random jitter was added that simulated RF phase noise. The entire digital subsystem design was then evaluated for performance under various jitter conditions. Since the jitter generation was developed under SystemC, detailed analysis can be performed through the SystemC model to understand the internal workings of the design under these non-ideal conditions. Such analysis would be extremely difficult or impossible to do in real-time.

Full-chip simulation with basic analog models allows validation of the interfaces between the analog part and the digital part. Analog models were simple high-level functional descriptions of the analog components in Verilog. Only critical functions of the analog modules were modeled: clock and reset generation, direction control of pads and pull-up enabling. The correctness of the each analog model was checked by the analog designer in charge of the design of the component. In addition to the normal modes, mode switching, power-on-reset, low-power and test modes have been simulated. Several incorrect signal polarities (i.e. active low enable signals), initial value incoherency or a deadlock in low-power mode state machine were found with the help of these models.

V. CONCLUSION

This paper introduced a novel methodology for designing entire mixed-signal systems, including firmware, software, digital, and RF/analog. A single, high level SystemC & ADS model is developed which creates an "executable specification" which fully represents the system and can be simulated very quickly. The model is further leveraged throughout the design process for design and verification and FPGA validation. This allowed a small design team to go from architecture to GDSII for the RF Subsystem ASIC in less than 4 months. The ASIC has been found to be a first pass success.

REFERENCES

[1] R. Singh and J.M. Cohn, Leading-Edge Computer Aided Design Solutions for Advanced Digital and Mixed-Signal Systems-on-Chips, *Proceedings of the IEEE*, vol. 95, issue 3, pp 465-466, March 2007.

[2] Ken Kundert, Henry Chang, Dan Jefferies, Gilles Lamont, Enrico Malavasi, Fred Sendig. Design of mixed-signal systems on chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1561-1571, December 2000.

[3] *IEEE Standard Verilog Hardware Description Language*, IEEE Standard 1364-2001.

[4] *IEEE Standard VHDL Language Reference Manual*, IEEE Standard 1076-2002.

[5] *Verilog-AMS Language Reference Manual*, Version 2.2 Accellera.

[6] *IEEE Standard VHDL Analog and Mixed-Signal Extensions*, IEEE Standard 1076.1-1999.

[7] D.C. Black and J. Donovan, SystemC *from the Ground Up*, Springer Science+Business Media, 2006