Bluetooth Transceiver Design with VHDL-AMS

Rami Ahola, Daniel Wallner
Spirea AB
Stockholm, Sweden
rami.ahola@spirea.com
daniel.wallner@spirea.com

Marius Sida
Mentor Graphics
Munich, Germany
marius_sida@mentor.com

Abstract

This paper describes the design challenges of BlueTraC™, a low-cost, low-power radio transceiver and the usage of mixed-signal/mixed-mode techniques and behavioral modeling with ADVance MS (ADMS) from Mentor Graphics to address and solve them. BlueTraC™ from Spirea is a Bluetooth 1.1 compliant Class 2 radio transceiver. In addition to all the required RF and analog functions, the chip also includes a complete digital GFSK modem, making it a very complex mixed-signal (MS) system-on-chip (SoC). VHDL-AMS, the mixed-signal IEEE 1076.1 standard modeling language, was used to describe the SoC building blocks at different levels of detail and complexity. This permitted us to perform top level functional verification and debugging, as well as detailed subsystem simulations throughout the design process. We are presenting the concept and the results we obtained, in terms of performance and accuracy. The methodology that we deployed increased the confidence in silicon success and allowed on time delivery.

1. Motivation and description of the case study

BlueTraC™ is a complex RF Mixed Signal (RFMS) IC. It contains approximately 50,000 gates and over 5,000 transistors for the RF and analogue baseband part. Part of the verification strategy was to simulate the complete SoC, the power-up, the power-down sequences and the state machine. Another important target in the verification strategy was to couple two BlueTraC™ systems to "talk" to each other. We aimed to simulate the bit error rate that was one of the most important parameters of the system.

Another important decision was to choose a top-down approach for the design of BlueTraC™ and to use hardware description languages (HDL's) for both the analogue and digital parts. Through effective usage of the HDL's we have been able to create complex and effective test benches and test strategies. Also the usage of HDL's have permitted the simulation of an RFMS system of this size. We will describe in the next chapters, the modelling approach, test bench creation and the verification results.

2. Top-down analogue and mixed-signal design methodology

A top-down analogue and mixed-signal (AMS) design methodology allows simple and rapid proof of concept, chip assembly and verification throughout all project phases. This concept has been successfully used in digital design for long time.

In the development of BlueTraC™, we used the Mentor Graphics' IC Flow that enabled us to use a top-down approach. We started with simple models already before the real transistor level design started. We performed simple architectural exploration and proof of concept. In later design phases, some of the models have been reviewed and refined, as we will exemplify in Chapter 4. We have been using VHDL-AMS language (1076.1 IEEE standard) to describe the analog and digital behavior of all the blocks.

The top-level schematic has been used to provide the design team the block level specifications in a simulatable form. Gradually, the transistor representation of the individual block replaced the VHDL-AMS models. This concept permitted us to check each block at chip level context and avoided simple errors, like block synchronization on the wrong edge of control signals, etc.

Design Architect IC (DA-IC), which is Mentor Graphics' schematic capture and simulation control environment, has been used for analogue and RF block design, as well as for top level verification, including co-simulation with the digital parts of the ASIC. Once all blocks have been designed at transistor level, an exhaustive verification of the specification through simulation has been done. We used different combinations of VHDL-AMS and transistor description to achieve a good accuracy and simulation speed. The Model Selector allowed us easy switching between the different simulation views, like VHDL-AMS, SPICE, schematic, etc.

Another critical piece in the design flow is the simulator. It needs to accommodate the different viewpoints we have selected. We used for simulation ADVance MS (ADMS) from Mentor Graphics. ADMS is a multilingual mixed-signal mixed-mode simulator. It simulates, simultaneously, VHDL-AMS, Verilog-A(MS) and SPICE described circuitry, in any combination.

The input netlist is a single hierarchical structure (ASCII file), which is automatically created and updated from DA-IC. The analog-digital, digital-analog or bi-directional data conversion is automatically performed and inserted.

ADMS was released in 1999 and is a mature and easy to use simulator. ADMS uses proven solvers, which are themselves (as standalone simulators) references in the specific design context. The individual simulators participating as solvers in ADMS are

- ModelSim: VHDL (VITAL), Verilog and "C", optional with SDL back-annotation.
- Mach: high speed and high capacity, transistor level timing and power analysis, including DSPF parasitics.
- Eldo: analogue transistor level simulation.
- Eldo RF: high capacity and high speed simulator, Harmonic Balance based, for analysis of RF IC circuitry like VCO's, LNA's, mixers, etc. It supports true multi-tone steady-state and noise analysis, including modulated signals.
- ADMS : above-mentioned simulators glued in one including the A/MS extensions of VHDL and Verilog.

The usage of ADMS and its solvers on different partitions of the design can be best explained by Figure 1. A closer look at the picture shows that all blocks/subsystems can have different "parallel" descriptions from high level HDL abstract representation, to detailed transistor level, including parasitics. This flexibility of ADMS has been a key factor in the success of the top-down design approach.

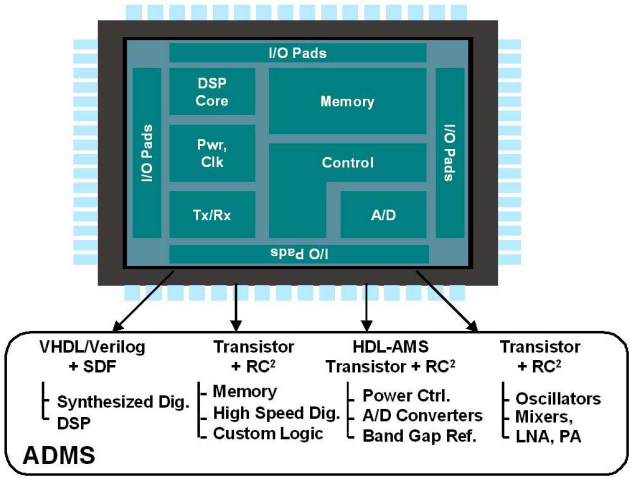

Figure 1: Partitioning of the design.

3. BlueTraC™ - A Bluetooth transceiver

BlueTraC™ is a true radio-on-a-chip Bluetooth transceiver. The design is based on an efficient architecture incorporating Spirea's CMOS design expertise making the solution both reliable and area efficient. From the architecture and system design levels down to the block and circuit design levels, BlueTraC™ incorporates dynamic power management and adopts a mixed signal CMOS design strategy achieving the highest level of integration. It is a cost effective Bluetooth radio solution requiring no external passive components, and features a low supply voltage and low power consumption.

To achieve the above features, many of the traditionally domain. This approach has several obvious advantages, for example benefiting from technology scaling, robustness against process, supply, and temperature variations, and design reusability. The main challenge in this approach is verifying the increased number of digital control interfaces and mixed-signal loops accurately enough, while keeping the simulation time of the system at a level where simulation can be used as a debugging and iteration tool and not only as a single-shot final verification. The number of devices – over 50,000 digital gates together with over 5,000 transistors in analogue and RF functions – gives an idea of the complexity of the task.

Figure 2 shows the block diagram of the BlueTraC™ transceiver. The design has been successfully verified with a power consumption of less than 50 mW in both RX and TX modes. More information, along with a complete datasheet, can be found in [3].

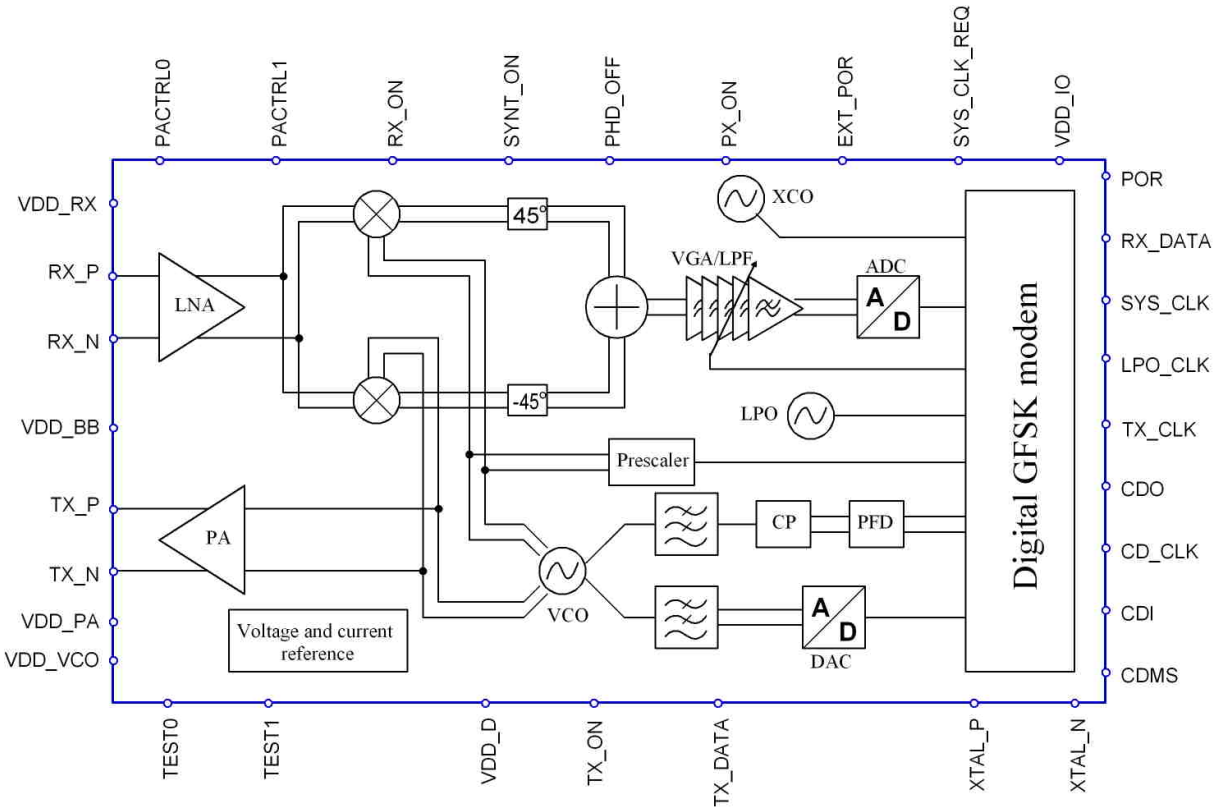

Figure 2: Block diagram of the BlueTraC™ radio transceiver.

4. Modelling approach

The modelling approach used in this project, as well as some of the key structures of the VHDL-AMS language will be explained with the help of an example. For this purpose we have chosen the analogue-to-digital converter (ADC) in the receiver chain (see Figure 2).

The source code of the ADC model starts with the library declaration. Please note the `DISCIPLINES.electromagnetic_system.all` library, which contains the analogue and mixed-signal definitions of `PORT TERMINAL`, data types as well as analogue-specific operations and attributes.

```
    library IEEE, DISCIPLINES;
    use IEEE.math_real.all;
    use IEEE.std_logic_1164.all;
    use IEEE.std_logic_arith.all;
    use IEEE.numeric_std.all;
    use
DISCIPLINES.electromagnetic_system.all;
```

Following the library declaration, we have the entity declaration describing the interface of the model to the outside circuitry. Specifically, it describes the analogue and digital ports as well as the parameters that are used in the behavioural description of the model and are influencing its electrical performance.

```
    entity adc is
      generic (
        ENOB: real := 5.3;
        DNL: real := 0.0;
        INL: real := 0.0;
        Cin: real := 500.0e-15;
        t_powerup: time := 10 us;
        t_powerdn: time := 10 us;
        bias_current: real := 10.0e-6;
        refp_voltage: real := 1.1;
        refn_voltage: real := 0.6
        );
       port (
      terminal inp: electrical;
      terminal inn: electrical;
      terminal vrefp: electrical;
      terminal vrefn: electrical;
      terminal vddan: electrical;
      terminal gndan: electrical;
      terminal vddd: electrical;
      terminal gndd: electrical;
      terminal vbody: electrical;
      terminal ibias: electrical;
      signal en: in std_logic;
      signal clk: in std_logic;
      signal d:out std_logic_vector(5 downto
0)
      );
    end entity adc;
```

After the entity declaration we have the architecture declaration, containing the actual description of the function of the model. It can be a mathematical description describing the physics of the model or it can contain so called structural constructs. Also the language allows a mixture of the two.

The model can be described at any level of abstraction. This is chosen by the user to achieve the best trade-off between speed and the amount of electrical effects incorporated into the model. The multiple architecture support was utilized throughout this project; most of the blocks have a simple architecture intended for top level verification and a more complex architecture intended for subsystem validation.

In this example, we use the simple algorithmic description of the ADC with first level electrical effects incorporated. In addition to the basic algorithmic description of the electrical behaviour, the model includes supply voltage level and bias current checks, input and output impedances, and a limited dynamic range.

Firstly, we limit the input signal to within the dynamic range of the ADC. The quantity `v_in` is the differential input voltage and `v_ref` is the voltage difference between the `vrefp` and `vrefn` electrical connections that define the conversion range of the ADC. The `if use` conditional statement is used to model a linear transfer from `v_in` to `v_conv` inside the conversion range and to clip the input signal outside the `v_ref` range as visualized in Figure 3.

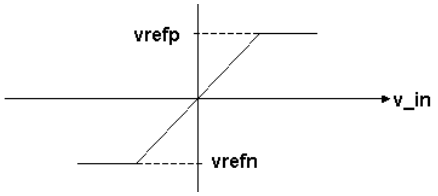

Figure 3: Limiting the dynamic range.

```
   architecture simple of adc is
      quantity v_in across
            I_in through inp to inn;
      quantity v_ref across vrefp to
vrefn;
      quantity v_conv: real;
   begin

     if v_in>v_ref use
       v_conv == v_ref;
     elsif v_in<(-v_ref) use
       v_conv == (-v_ref);
     else
       v_conv == v_in;
     end use;
```

Next, we model the input impedance. The input capacitance, Cin is created using following equation:

```
     i_in == Cin * v_in'dot;
```

The power-up and power-down of the ADC are controlled by the enable signal `en`, and may have different associated delays, controlled by the parameters `t_powerup` and `t_powerdown`. This effect has been modelled by a process sensitive to `en`:

```
   powerup: process(en)
   begin
     if en = '1' then
       power_on <= 1.0 after t_powerup;
     else
       power_on <= 0.0 after t_powerdn;
     end if;
   end process;
```

Finally, the data conversion is performed on the rising edge of the `clk` signal:

```
   convert: process(clk)
   begin
     if (v_vddan>1.5) and (v_vddd>1.5)
  and (i_ibias<-0.5*bias_current)
  and (i_ibias>-1.5*bias_current) then
       if (clk = '1') then
         d <= transport
conv_std_logic_vector
(conv_unsigned(integer(power_on*floor(63.0*
(v_conv+v_ref)/(2.0*v_ref))),6),6) after
300ns;
       end if;
     else
       d <= "000000";
     end if;
   end process convert;
```

Please note the first `if` statement that allows the conversion to take place only if power-up conditions have been fully met. Through this we accurately model the power management and the model will "sense" any possible problem in the biasing of the ADC. Also this coding style makes the model fast and highly effective which is one of the key issues for top level verification.

The above effects were the ones that were used for the top-level verification. Another, more detailed model was used for subsystem validation. This model uses a structural approach, modelling each of the blocks inside the ADC individually. The block diagram of the converter is shown in Figure 4.

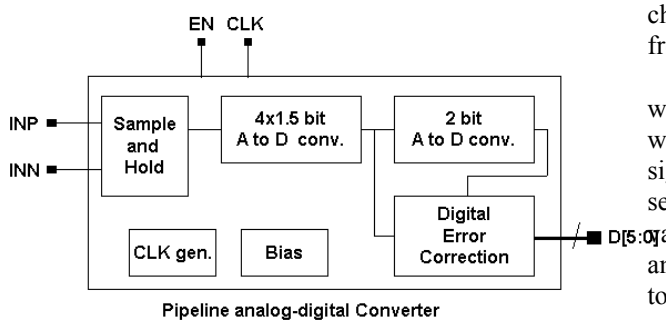

Figure 4: Block diagram of the ADC.

5. Subsystem design and validation

The BlueTraC™ design includes a number of mixed-signal subsystems, such as digital calibration of analogue filtering functions, a phase-locked loop, and an automatic gain control algorithm in the receiver chain. To verify the performance of these subsystems in all operating conditions, the following model calibration flow was used. Firstly, very basic behavioural models for the analogue functions were used to validate the concept. In the next step, the block design at the transistor level was done. After that, a few selected test cases were used to refine the behavioural models to match the transistor level models (this iteration process is commonly referred to as model calibration). When an agreement between the two models was reached, exhaustive verification was performed on the behavioural model. As the following example shows, this would not have been feasible if only transistor level models were used.

The automatic gain control loop in the receiver chain of BlueTraC™ (see Figure 2) is as such rather complex, consisting of a variable gain amplifier merged with a lowpass filter (VGA/LPF), an analogue-to-digital converter (ADC), a digital channel selection filter, and a digital gain adjustment algorithm. Moreover, this loop also affects the performance of the demodulator, and thus the only reliable figure of merit was the bit error rate (BER) at the output of the entire receiver.

To get a reliable estimate of the BER, a minimum simulation of 100,000 payload symbols was required, resulting in an over 100 milliseconds long transient run. With the analogue blocks (VGA/LPF and ADC) modelled on transistor level and the digital blocks in synthesisable VHDL code, the simulation time was far too long to allow exhaustive simulation in all conditions. When accurate behavioural models replaced the transistor level models, the simulation time reduced significantly, allowing us to verify the design against all combinations of the Bluetooth specifications for in-band and co-channel interferers, carrier frequency offset, and carrier frequency drift [4].

Figure 5 shows an example of the simulation waveform window. The input power to the transceiver was changed randomly after every 300µs, and the output signal of the variable gain amplifier was plotted. We can see the gain control loop regulating the gain of the variable gain amplifier to provide optimum input amplitude (approximately 700 mVpp) for the analogue-to-digital converter.

In addition to fast and accurate models, the test bench plays an important role in the subsystem validation. Especially analogue designers tend to view the test bench only as a means to provide stimuli to the circuit. However, in the case of complex mixed-signal systems, this approach produces enormous amounts of data that is impossible to analyse effectively. Therefore, special attention was paid to developing a test bench that not only generates the stimuli, but also analyses the output of the simulation and reports it to the designer in a concise manner.

The test bench created for the automatic gain control loop validation makes extensive use of the ASSERT statement of the VHDL language. It prints out the number of errors in the packet header and in the payload on a packet-per-packet basis, as well as the total bit error rate and packet error rate at the end of the simulation. An example text output of the gain control loop simulation is shown in the following:

```
   Start of packet detected : 211.454ms
   Preamble correct.
   Errors in access code: 2
   Number of bits compared: 2740
   Number of errors: 1

   -> TOTAL # of packets: 10
   -> TOTAL # of payload bits compared:
27400
   -> TOTAL # of errors: 8
   -> TOTAL # of preambles with error: 3
   -> TOTAL # of lost packets: 0
```

Using this type of test benches that report the outcome of the simulation in plain text speeds up the analysing of the results enormously. Of course, the designer still has the freedom of checking the traditional waveform windows if that should be necessary.

6. Top level verification

The test bench used for top-level functional verification included two BlueTraC™ transceivers and a VHDL test bench controlling the radios and collecting the simulation results. One of the radios was configured to transmit data (real DH1 packets), while the other one operated in receive mode. The simulation started from

power down, so the power up sequence of the chip was also verified.

The simulation time increases with increasing model accuracy. We used models accurate enough to give reasonably reliable bit-error-rate estimates, and still the simulation times were quite reasonable. Approximately 30 Bluetooth packets (20 milliseconds total) could be transmitted and received during an overnight simulation on a two-processor SunFire 280r workstation.

7. Conclusions

We have successfully simulated the complete transceiver using VHDL-AMS and ADMS. Our two main objectives were fully met. Firstly, we were able to simulate a complex mixed-signal SoC at the top level, completely verifying connectivity and power up and power down sequences, as well as the state machine controlling the RF and analogue parts. Secondly, we were able to perform a rather thorough verification of the performance of several mixed-signal subsystems. This would not have been possible with a traditional analogue simulator due to unreasonably long simulation times.

Currently, our analogue behavioural model only includes detailed descriptions of the low-frequency blocks. The descriptions of the radio frequency blocks are first order models. We have been basically verifying connectivity only. The next step in the project will be to create more detailed models for the RF blocks as well. After that, a more realistic channel model can be introduced, and the performance of the radio can be verified in a "real" communication situation.

8. References

[1] IEEE 1076.1 workgroup, Language Reference Manual, http://www.vhdl.org/analog

[2] ADVance MS™ Reference Manual, Mentor Graphics, http://www.mentor.com

[3] BlueTraC™ Datasheet, Spirea AB, http://www.spirea.com/

[4] Bluetooth Specification 1.1, http://www.bluetooth.org

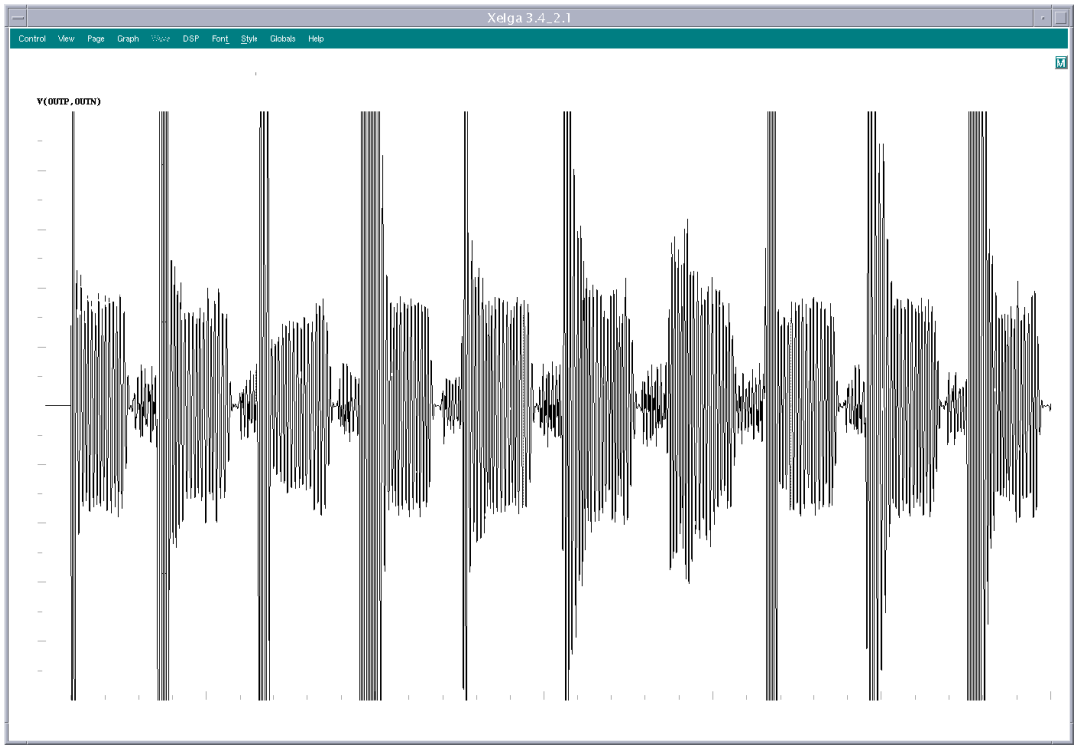

Figure 5: An example waveform at the output of the variable gain amplifier.