A VHDL-AMS Modeling Methodology for Top-Down/Bottom-Up Design of RF Systems

Torsten Maehne, Alain Vachoux
Laboratoire de Systèmes Microélectroniques (LSM)
École Polytechnique Fédérale de Lausanne (EPFL)
WWW: http://lsm.epfl.ch/
E-Mail: {torsten.maehne,alain.vachoux}@epfl.ch

Frédéric Giroud, Matteo Contaldo
Microelectronics Division
Centre Suisse d'Electronique et de Microtechnique SA (CSEM)
WWW: http://www.csem.ch/
E-Mail: {frederic.giroud,matteo.contaldo}@csem.ch

***Abstract*—This paper presents a modelling methodology for the top-down/bottom-up design of RF systems based on systematic use of VHDL-AMS models. The model interfaces are parameterizable and pin-accurate. The designer can choose to parameterize the models using performance specifications or device parameters back-annotated from the transistor-level implementation. The abstraction level used for the description of the respective analog/digital component behavior has been chosen to a good trade-off between accuracy, fidelity, and simulation performance. These properties make the models suitable for different design tasks such as architectural exploration or overall system validation. This is demonstrated on a model of a binary FSK transmitter parameterized to meet very different target specifications. The achieved flexibility and systematic model documentation facilitate their reuse in other design projects.**

I. Introduction

Portable, battery-powered electronic devices (cell phones, PDAs, notebooks, mice, ...) nowadays routinely contain wireless functionality to communicate among each other using standards like Bluetooth™, Wi-Fi™, or GSM™. This became possible due to the integration of full RF transceivers into a System-on-a-Chip (SoC), built of these main parts (Fig. 1):

- **Frequency synthesizer:** based on a Phase-Locked Loop (PLL) consisting of a Voltage Controlled Oscillator (VCO), DIViders (DIVs), Phase/Frequency Detector (PFD), Charge Pump (CP), LooP Filter (LPF), Quartz Oscillator (QO) for the reference frequency, Σ-Δ modulator for fractional frequency division and Multi-stAge-noise-SHaping (MASH) to generate the RF carrier signal required for demodulation and modulation.
- **Receiver (RX) chain:** Channel Selector (CS), Low-Noise Amplifier (LNA), MIXer (MIX), Intermediate Frequency Amplifier (IFA), Poly-Phase Filter (PPF), and Digital Signal Processing (DSP).
- **Transmitter (TX) part:** modulation of the Pulse-Shaped (PS) data bit stream on the RF carrier using direct modulation of the fractional PLL and its amplification with a Power Amplifier (PA).
- **POWer management (POW):** linear regulators, step-up converters.

This work has been funded by the Hasler Stiftung under project № 2161.

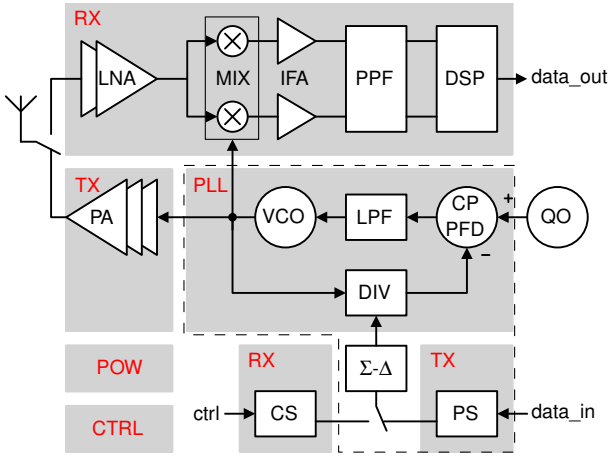


Fig. 1. A global view of the RF transceiver

- **Digital ConTRoL (CTRL)** of the transceiver through some communication interface (e.g., SPI or I^2C bus) or any other digital system (e.g., microcontroller).

The design of such a complex system is highly demanding since its analog/RF and digital parts require different design flows, methodologies, and tools.

During the *top-down design phase*, the overall system specifications must be properly distributed and assigned to the components defined by the selected architecture. Then, the result has to be evaluated and checked against constraints such as the feasibility with the selected technological process. The distributed specifications will then be used for the detailed block design until their layout is obtained. While Hardware Description Languages (HDLs) such as VHDL or Verilog are routinely used for the design of digital parts, mathematical tools such as MATLAB/Simulink or spreadsheets are preferred for deriving analog/RF specifications and performing system-level studies. However, mixed-signal HDLs such as VHDL-AMS or Verilog-AMS do offer additional capabilities that can support the top-down design and more specifically the architectural exploration step. Anticipating the real component behaviors with models at proper levels of abstraction using appropriate parameters is a considerable help in that phase. Their reusability for evaluating various possible architectures

is also an advantage [1].

During the *bottom-up verification phase*, the implemented components must be individually verified and then assembled to form the complete system under design. Assuming that each component has been successfully verified, a full top-level verification is necessary for detecting interface issues such as wrong signal behavior or incorrect timing. As full system verification at transistor level is becoming impracticable due to the huge amount of data to handle, models at proper levels of abstraction using appropriate parameters again offer the required accuracy/verification performance trade-off [2].

This paper reports on a work that has the goal to develop and provide a library of VHDL-AMS models called `RF_TRX` that supports both the top-down architectural exploration and the bottom-up verification of complex RF systems such as the RF transceiver. VHDL-AMS models of components of a Frequency-Shift Keying (FSK) transmitter (indicated by the dashed line in Fig. 1) have been developed following a particular modeling methodology, as presented in Section II. Special care has been taken to model the proper non-idealities, while conserving acceptable simulation performances, and to define consistent parameter sets that support the top-down (specification) and bottom-up (back-annotation) phases. Similar model libraries have been already presented, e.g., in [3]–[5], but mostly focus on top-down design only.

The paper is organized as follows. Section II presents the modeling methodology that has been used for the PLL in the frequency synthesizer. The development of a model of the VCO is described in detail, from the designer's specification to the actual VHDL-AMS model. Other component models needed to model the frequency synthesizer are only outlined as the space is limited. Section III presents the usage of the developed VHDL-AMS models for the architectural exploration of a FSK transmitter in two different design cases using carrier frequencies of 868 MHz and 2.45 GHz, how the developed structural transmitter model can be reused during bottom-up verification, and how modeling decisions impact the simulation performance. Finally, Section IV includes some conclusions and an outlook on future steps.

II. Modelling Methodology

In this section we outline the modeling methodology and illustrate its application using the frequency synthesizer and particularly its VCO component as an example.

The first step involves the *definition of the requirements the model library shall meet*. This includes the definition of the components that will be modeled, the level of detail, at which the component behaviors shall be described, the parameters that shall be available to the model user, and the parameters that shall be derived internally. The definition of these two kinds of parameters has to be done carefully as the same model shall be used for both top-down architectural exploration and bottom-up verification. In the top-down phase, the interface (generic) parameters are used as specifications and internal parameters are used as nominal values for the design of the component at the transistor level. In the bottom-up phase, the interface parameters are back-annotated with values from the transistor-level implementation and internal parameters have in turn values that correspond to extracted specifications. This also means that models shall be pin-accurate for easy replacement with equivalent transistor-level models, when required. The second step involves the *specification of the behaviors to model, their interfaces (signals and parameters), and their internal parameters* as discussed above. This depends largely on which specifications need to be validated with the help of the (sub-)system model and requires a preliminary understanding of the overall system function to derive the principal component parameters that have an influence on the system performance. The specification is largely prepared by the domain experts, i.e., in our case the RF designers, with feedback given by the model developers, e.g., asking for clarification or restructuring of certain specifications. Then, the *agreed specifications have to be translated into legal VHDL-AMS models*. In this step, the capabilities offered by the modeling language and, unfortunately, the incomplete support of the language in existing EDA tools, have to be balanced to achieve the intended accuracy/fidelity and simulation performance. Finally, the *models shall be thoroughly tested using elaborate test benches* that include as much as possible the verification of assertions about the modeled components.

As these steps imply a non-negligible implementation effort, the models shall be as flexible/modular as possible to be *reusable in other design projects* to accelerate and optimize the design process using an over time growing base of well documented and individually reviewed and verified component models, in which validity other designers can trust. The particular requirements for such a model library are outside the scope of this article, but described in more detail in [6].

A. Specifications for the Frequency Synthesizer

A frequency synthesizer [7] is a system that allows to enslave an oscillator (VCO) at a specific frequency. It is most of the time based on a PLL (Fig. 1). For the RF application described here, the VCO is designed to give a signal at high frequencies, like those specified for the ISM or SRD bands. The frequency of the VCO will change until its frequency, divided by the ratios of the frequency dividers, is exactly equal to the frequency of the quartz oscillator. The equation expressing the relation between VCO frequency, the division ratio, and the quartz reference frequency is:

$$f_{\mathrm{vco}} = f_{\mathrm{REF,PLL}} \cdot N_{\mathrm{div}} \tag{1}$$

Variables with uppercase subscripts will be considered later as model constants (either generic parameters or local constants) and variables with lowercase subscripts will be considered later as model quantities or signals.

As for every looped system, a *stability problem* occurs, if the phase margin of the PLL open-loop gain is too small. For calculating this open-loop gain, we need to define the principal parameters (Fig. 2) of each block:

- **VCO:** K_{vco} in Hz/V, expresses the relationship between the input voltage v_{tune} and the output VCO frequency.

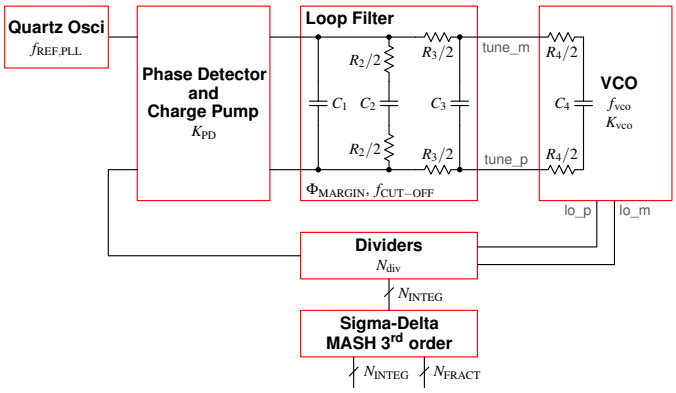


Fig. 2. A global view of the frequency synthesizer

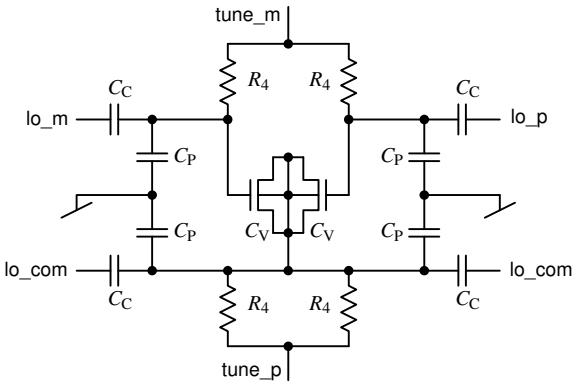


Fig. 3. Wiring of the varicap inside the VCO structure

- **Frequency divider:** N_{div}, modulated by the Σ-Δ modulator, is the division ratio between the VCO and the quartz oscillator frequencies. In an implementation N_{div} might be obtained by cascading several dividers.
- **PFD and CP:** K_{PD} in A/rad, expresses the gain between the phase error at the input of the PFD and the instantaneous current flowing at the output of the CP.
- **LPF:** C_1, R_2, C_2, R_3, C_3 are the circuit device parameters of the loop filter, which need to be adjusted to get a good phase margin Φ_{MARGIN} for a specific cut-off frequency $f_{\text{CUT-OFF}}$. Note that a fourth order R_4, C_4 (impedance at the VCO input) may be taken into account.

Using these parameters, the open-loop transfer function of the PLL is given by the following equation:

$$H(j\omega) = \frac{\frac{K_{\text{vco}} \cdot K_{\text{PD}}}{N_{\text{DIV}} \cdot j\omega} \cdot Z(j\omega)}{\left(1 + j\omega \cdot R_3 \cdot \left(C_3 + \frac{C_4}{1+j\omega \cdot R_4 C_4}\right)\right) \cdot (1 + j\omega \cdot R_4 C_4)} \quad (2)$$

where $Z(j\omega)$ is the impedance seen by the charge-pump:

$$Z(j\omega) = \frac{1}{j\omega \cdot \left(C_1 + \frac{C_2}{1+j\omega \cdot R_2 C_2} + \frac{C_3 + \frac{C_4}{1+j\omega \cdot R_4 C_4}}{1 + j\omega \cdot R_3 \cdot \left(C_3 + \frac{C_4}{1+j\omega \cdot R_4 C_4}\right)}\right)} \quad (3)$$

By calculating the phase value, for which the magnitude of (2) is equal to one, the *system phase margin* can be derived [8].

Different implementations can be used to build the *VCO* by satisfying the gain and phase conditions needed to ensure steady-state oscillation [9]. When used in a frequency synthesizer, the oscillator has to cover a precise output frequency range determined by the target application. A tunable ideal oscillator with continuous output phase and control voltage can be represented by the following equation:

$$v_{\text{osc}}(t) = V_{\text{AMP}} \cdot \cos\left(2\pi\left(K_{\text{vco}} \int_{-\infty}^{t} v_{\text{tune,eff}}\, dt + f_{\text{vco},0} \cdot t\right)\right) \quad (4)$$

The oscillator taken into account for the modeling is an *LC* oscillator with band switching capabilities. It is composed by an inductor and an equivalent capacitor needed to fix the oscillator frequency, while some active devices compensate for the losses in order to sustain the oscillation.

The frequency tuning is performed by adjusting the capacitance value using at the same time an analog and a digital control. The analog control is achieved by means of a varicap allowing continuous frequency tuning over a certain range. The digital control is performed by driving a bank of switchable capacitances, with the result of shifting in a discrete way the center frequency $f_{\text{vco},0}$ of the range covered with the analog control. The final oscillation frequency is thus given by:

$$f_{\text{vco}} = \frac{1}{2\pi \cdot \sqrt{L_{\text{TANK}} \cdot (C_{\text{tank}}(band) + C_{\text{v,tot}}(v_{\text{tune,eff}}))}} \quad (5)$$

with L_{TANK} the oscillator inductance, $C_{\text{tank}}(band)$ the bank capacitance for the selected *band*, $C_{\text{v,tot}}(v_{\text{tune,eff}})$ the variable capacitance for the effective tuning voltage $v_{\text{tune,eff}}$.

The previous equations show that the primary parameter needed for a correct VCO model is K_{vco}. It is present in the time- and frequency-domain equations (by means of the variable capacitance) affecting not only the VCO behavior but also the PLL stability (2). The value of K_{vco} is directly related to the characteristic of the varicap, which is the circuit element used to perform the analog frequency tuning. Therefore, a detailed study of the varicap and its usage in the VCO circuit has to be done.

Fig. 3 depicts the varicap configuration inside the VCO structure with biasing resistances R_4, coupling capacitances C_C, and parasitic capacitances C_P. The equivalent capacitances seen from the filter and from the oscillator terminals are:

$$C_4 = 2C_{\text{v}} + C_{\text{C}} + C_{\text{P}} \quad (6)$$

$$C_{\text{v,tot}} = \left(\frac{2}{C_{\text{C}}} + \frac{2}{C_{\text{v}} + C_{\text{P}}}\right)^{-1} \quad \text{with} \quad \frac{C_{\text{P}}}{C_{\text{C}}} \cong 1\,\% \quad (7)$$

The nonlinear dependency of the varicap density from the effective tuning voltage $v_{\text{tune,eff}}$ at its terminals can be accurately described using a hyperbolic tangent function:

$$D_{\text{cv}}(v_{\text{tune,eff}}) = D_{\text{CV,AMP}} \cdot \tanh\left(\frac{\frac{dD_{\text{CV},0}}{dv_{\text{tune,eff}}}}{D_{\text{CV,AMP}}} \cdot (v_{\text{tune,eff}} - V_0)\right) + D_{\text{CV},0} \quad (8)$$

with the following technology-dependent parameters (Fig. 4):

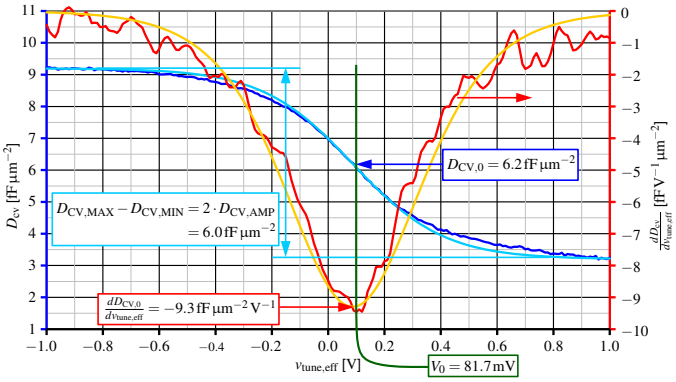


Fig. 4. Fitting of varicap density and its derivative for TSMC 0.18 µm

- V_0: Offset tuning voltage for the point with the steepest slope $\frac{dD_{cv}}{dv_{tune,eff}}$ of the measured varicap density
- $D_{CV,0} = D_{cv}(V_0)$: Varicap density measured for V_0
- $D_{CV,AMP}$: Varicap density modulation amplitude
- $\frac{dD_{CV,0}}{dv_{tune,eff}} = \frac{dDcv}{dv_{tune,eff}}\Big|_{V_0}$: Slope of D_{cv} measured at V_0

By deriving the varicap density function with respect to the tuning voltage $v_{tune,eff}$, we obtain:

$$\frac{dD_{cv}}{dv_{tune,eff}} = \frac{dD_{CV,0}}{dv_{tune,eff}} \cdot \left(1 - \left(\frac{D_{cv}(v_{tune,eff}) - D_{CV,0}}{D_{CV,AMP}}\right)^2\right) \quad (9)$$

Then, by defining the area A_{CV} of the varicap, it is possible to calculate its absolute value and derivative:

$$C_v = A_{CV} \cdot D_{cv} \qquad \frac{dC_v}{dv_{tune,eff}} = A_{CV} \cdot \frac{dD_{cv}}{dv_{tune,eff}} \quad (10)$$

To calculate K_{VCO}, we still need the derivative $\frac{dC_{v,tot}}{dv_{tune,eff}}$ of the total equivalent varicap $C_{v,tot}$ seen by the VCO between the terminals lo_p and lo_m taking into account the coupling and parasitic capacitances present in the circuit (Fig. 3):

$$\frac{dC_{v,tot}}{dv_{tune,eff}} = \frac{1}{2} \cdot \frac{1}{\left(1 + \frac{C_v}{C_C} + \frac{C_P}{C_C}\right)^2} \cdot \frac{dC_v}{dv_{tune,eff}} \quad (11)$$

Using (5) and (11), we can finally calculate K_{vco}:

$$K_{vco} = \frac{df_{vco}}{dv_{tune,eff}} = 2\pi^2 \cdot f_{vco}^3 \cdot L_{TANK} \cdot \left(-\frac{dC_{v,tot}}{dv_{tune,eff}}\right) \quad (12)$$

Using the technology constants introduced for the varicap density given by (8), and choosing the ratios $\frac{C_C}{C_{V,0}}$ and $\frac{C_P}{C_C}$ for the capacitances, the varicap area A_{CV} and its related capacitances C_C and C_P can be calculated to optimize the VCO to have a maximal K_{vco} at a certain target frequency $f_{VCO,OPT}$. This optimal frequency is supposed to be at the inflection point of $D_{cv}(v_{tune,eff})$, at V_0. This maximal K_{vco} is used for the PLL stability calculation with the help of (2) and (3). For accurate simulation, the instantaneous value of K_{vco} is always used.

B. Design, Implementation, Test of the Frequency Synthesizer

During the design and implementation phase, the functional specification developed together with the RF designer is translated into an executable model. In the RF_TRX library, the PFD, CP, dividers and Σ-Δ modulator models have a single architecture, which satisfies the needs for the top-down design and bottom-up verification phases. Their top-down design parameters are also used during the bottom-up phase, which adds just additional details like delays or mismatches. The *Phase/Frequency Detector (PFD) model* is implemented as a digital behavior description (two D-flip-flops and one AND-gate) augmented by generics to specify the propagation delays of the gates back-annotated during the bottom-up phase. The *differential charge pump model* is implemented as ideally switched current sources with internal resistance. Their value is determined as a multiple of a measured input bias current, as they are implemented as current-mirrors on transistor level. Mismatches can be specified for the bottom-up verification. This level of precision is sufficient as, for a more detailed study of the charge pump's impact (especially the switching effects from the transistors) on the PLL performance, many simulators allow the mixed simulation of VHDL-AMS and SPICE component models.

The implementation of the *differential loop filter model* follows the stability analysis of the PLL. It models the *RC* network as branch equations. This allows to use simultaneous **if**-statements to implement the filter orders 1, 2, and 3 in the same model. For the top-down design phase, the RF designer can specify in the generics, besides the filter cut-off frequency $f_{CUT-OFF}$ and order, the phase margin Φ_{MARGIN} to be realized for the PLL characterized by the charge pump sensibility K_{PD}, the VCO frequency sensibility K_{VCO}, and the frequency division factor N_{DIV}. This information is used in the *top-down architecture* to calculate the R and C values of the filter taking into account the device ratios $\frac{R_3}{R_2}$ and $\frac{C_3}{C_1}$ and the (possibly back-annotated) VCO input impedance defined by R_4, C_4. The *bottom-up architecture* allows to directly specify the R and C values.

The *VCO model* constitutes to date the most complex model of the RF_TRX library. Five different interfaces (i.e., entities) provide the RF output signal at various levels of abstraction:

- **Instantaneous frequency:** is calculated using (5). This accelerates transient simulations as the integration step width of the analog solver can be increased up to one order of magnitude below the time constant of the PLL. This abstraction can be useful in combination with other base-band modeling approaches [10] to use the VCO output for up- and down-conversion in the frequency domain.
- **Single-ended and differential digital signal:** are generated completely in the Discrete-Event (DE) domain by sampling f_{vco} at the state change of the digital signal and calculating from it the delay till the next state change. This decoupling of analog and discrete solver allows again the analog solver to increase its integration step width. This abstraction is useful, if the VCO output is fed directly into digital blocks.
- **Sinusoidal quantity or differential voltage signal:** are generated by integrating the instantaneous f_{vco} over time to calculate the phase, which defines together with the specified amplitude the current value of the sinusoidal

waveform. This generation of the RF signal in the Continuous-Time (CT) domain slows down considerably the analog solver, which integration step-width needs to stay one order of magnitude below f_{vco}. It is still more efficient than using a complete *LC* oscillator model. This precise abstraction maybe necessary to interface analog blocks operating at RF and to do spectral analysis.

All five interfaces have in common that they use the instantaneous f_{vco} (5). That is why it was decided to reuse the VCO model with frequency output inside the models with digital or sinusoidal output. The *frequency to digital* and *frequency to sinusoidal conversion* has been factorized into separate models, which can be reused independently.

The VCO model implements the functional specification from Section II-A on three abstraction levels. The *ideal* architecture implements the band switching capability and uses a constant frequency sensitivity K_{VCO}. No input impedance is considered at the tuning terminals. Two *detailed* architectures (one for top-down design, the other for bottom-up verification) implement the full non-linear behavior of the VCO, as specified in Section II-A, including band-switching, non-linear varicap, non-linear input impedance depending on $v_{tune,eff}$, and calculation of the instantaneous K_{vco}. The *top-down variant* uses a subset of the generics defined in the entity, which represent the component specification (e.g., number of bands, frequency range, $K_{VCO,OPT}$ at $f_{VCO,OPT}$) and the implementation technology (fitting parameters of (8)) to calculate as secondary parameters in the architecture the internal device parameters (e.g., A_{CV}, C_C, C_P, and parameters of the tank capacitance bank). The *bottom-up variant* uses directly the device parameters passed through the generics.

Several programmable divider models have been implemented [11] with a focus on a flexible interface (size of control word, digital input or instantaneous frequency input) and parameterizable digital behavior (e.g., minimal/maximal division ratio, control word treated as absolute division ratio or offset to minimal division ratio). For a two-stage divider architecture, a divider model with integrated control logic for a prescaler has been implemented.

For fractional division ratio control and noise shaping, a Σ-Δ modulator model describing a MASH structure [7] of 1st to 3rd order on Register Transfer Level (RTL) (using **`generate`** statements) has been developed [11]. The number of bits representing the integer and fractional part can be adjusted making it easy to adjust the modulator resolution during architectural exploration to meet the frequency synthesizer specification and to reuse it in other projects.

The divider and Σ-Δ modulator models are primarily targeted to be used during the top-down design phase. During bottom-up verification, they can be replaced by the gate-level HDL models with detailed annotated delays exported by the synthesis and place & route tools.

The power consumption of these PLL components is not yet explicitly modeled as it is less important for the evaluation of the RF specification. Still the power supply terminals are included in each model, which test that the supply voltage is applied with the correct polarity and remains in the specified range for the supply voltage. This is important to facilitate the bottom-up verification of the connectivity.

Common to all models is the systematic usage of assertions to check for consistent parameterization of the model and compliance of the model state with the modeling assumptions. To facilitate code comprehension and maintenance, a consistent coding style is enforced through peer-review establishing a naming convention, consistent commenting of each declared constant/quantity/signal regarding its purpose and physical unit, encouraging the usage of common coding patterns for digital behavior, keeping the implementation of each model effect as local and independent from each other as possible, etc. All architectures are validated with the same test bench by using configurations, which implement automated measurements and checks for the various model behaviors. Simulation control scripts and waveform display configurations are provided for each configuration to ensure reproducibility of simulation runs. As compilation and simulation of all models and test benches is handled through a common build system implemented for ADVance MS (ADMS) 2008.2, modification/simulation turnaround times are drastically reduced and code modifications can be checked more easily. The model library and its simulation environment are managed by a Subversion repository, facilitating its parallel development and usage.

III. Usage of the RF_TRX Library in the Design Process of a Binary FSK Transmitter

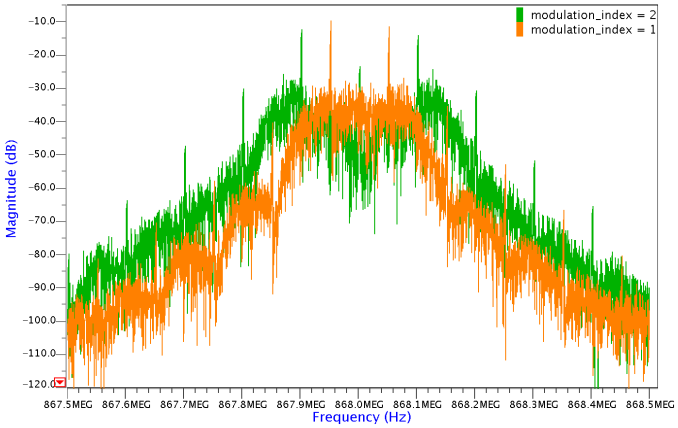
The binary FSK transmitter used in this section as an example is based on the frequency synthesizer architecture depicted in Fig. 2 with direct modulation done at the Σ-Δ modulator input. The Σ-Δ control words are generated by a bit-stream modulator model based on the bit stream sampled at its input, to which it can apply a pulse-shaping technique (none, ramped, or raised-cosine) [11]. The PA is not modeled in this example, as the focus is on the evaluation of the modulation process.

The transmitter model is realized as a structural model instantiating the necessary `RF_TRX` models not directly, but through **`component`** declarations provided in a **`package`**. All generic parameters of the component instances are forwarded through the **`generic`** declaration of the transmitter model. This allows us to parameterize the whole model hierarchy from the test bench. The constants used for this purpose are declared in an own package for each design case (target specification to implement). This has several advantages. The consistency during the evaluation of different architecture options is ensured, since constants, common to all architectures, are declared only once. Architecture-specific constants can be declared in the same package possibly based on the common once. Constants representing the back-annotated component parameters from transistor level are declared in this package, too. The stimuli generation (in our case pseudo-random bit-stream, reset, clock, and power supply) is factorized into an own model and parameterized in the test bench by the design case constants package. The test bench to be created for each design case and architecture to be evaluated gets thus

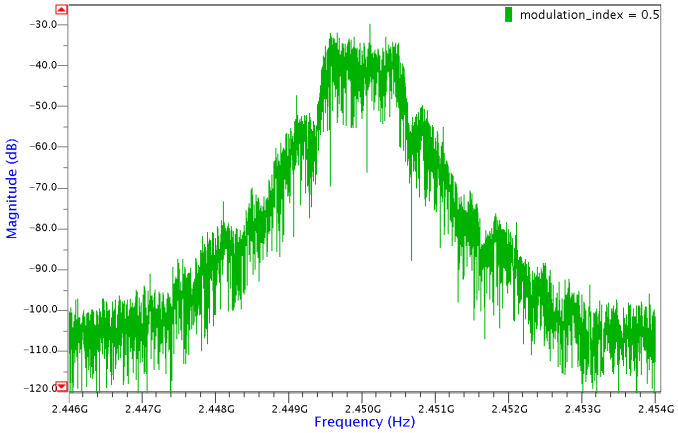
very simple, as it just includes the constants package for the design case, declares all signals to interconnect the instances created of the stimuli generator and transmitter architecture models, and carries out the constants mapping to the generic parameters. Finally, the mapping of the components instances is done in a **`configuration`** for the test bench. This allows us, e.g., to reuse the same structural model during the top-down design and bottom-up verification phases by selecting the appropriate architectures of the component models. As for the component test benches, the transmitter model is integrated into the automated build system with simulation control scripts and waveform display configurations for each design case model configuration. This careful model organization achieves a full orthogonalization of the aspects *model structure*, *parameterization*, and *abstraction selection* avoiding code duplication and making it easy to add new design cases and model configurations.

Three different design cases have been chosen to validate the FSK transmitter model and show its flexibility. The first two examples represent possible wireless sensor network applications. The targeted output frequency (868 MHz) is in the Short Range Device (SRD) activity band and the chosen modulator data rate (100 kbit/s) allows to reach the duty cycling requirement (< 1 %) for a low data rate application, e.g., a sensor data transfer. The two simulations differ in the modulation index, set respectively to 2 and 1, to analyze the influence of a frequency deviation modification on modulation. The third design case targets a transmission in the ISM band, for a compliant Bluetooth™ or Wibree™ application. The chosen carrier frequency and data rate are 2.45 GHz and 1 Mbit/s, respectively, suitable for applications like audio streaming. For this scenario, a modulation index of 0.5 has been used.

In order to have indications on the modulation quality, two different analysis are performed: signal spectrum and eye diagram evaluation. Depending on the targeted band (SRD, Bluetooth), different requirements have to be fulfilled in order to verify that the transmitted signal respects the standard specification and does not perturb adjacent channels. For this reason, the modulated signal spectrum has to be compared to the spectral mask, verifying that it never overcomes this reference curve. In the presented simulations (Fig. 5), only the contributions coming from the ideal PLL modulation determine the final signal spectrum. The noise generated by the PLL blocks and the PA non-linearities are thus not taken into account. Nevertheless the modulated signal at the output of the VCO already gives a fair indication of the occupied spectrum highlighting possible problems in the modulation process or in the PLL design. By applying an ideal demodulation on the same VCO signal and processing the result, the eye diagram can be built (Fig. 6). The study of the eye diagram allows to gather useful information both on the transmitted signal and the demodulation requirements. An example is the choice of the optimum sampling instant to minimize inter-symbol interference. Concerning the top-down design methodology, this analysis shows mainly the impact of the Σ-Δ quantization noise on the signal integrity, giving indications on the trade-



(a) f_c = 868 MHz, $MI = \{2, 1\}$, $DR = 100kbit/s$



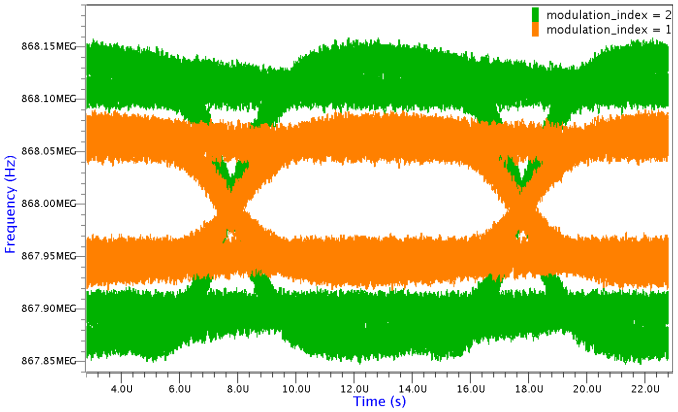
(b) f_c = 2.45 GHz, MI = 0.5, $DR = 1MHz$

Fig. 5. Frequency spectrum of the FSK transmitter output signal

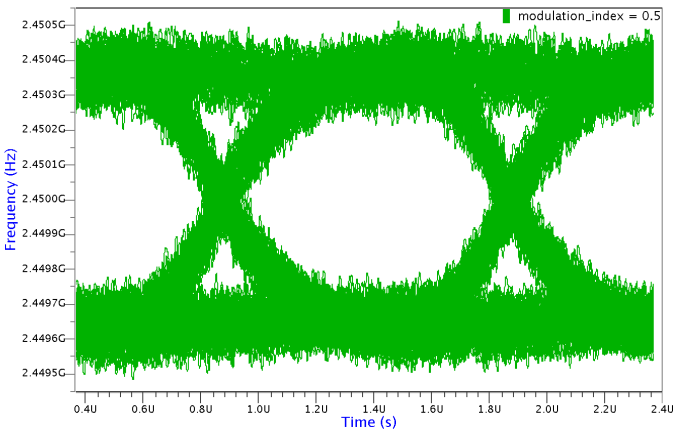
offs between the reference, the carrier, and the filter cut-off frequencies. For the bottom-up verification phase, it is very convenient that the `RF_TRX` models can be also parameterized with the back-annotated device parameters—especially as the device parameters calculated by the top-down architectures cannot be implemented usually exactly on the transistor level due to design rules limitations and other constraints.

TABLE I summarizes the simulation performances obtained for the first and third design cases[1] of the FSK transmitter. Three transmitter model variants were developed, in order to investigate the impact of the different VCO output modeling approaches on the transmitter model simulation performance. As expected, the variant with digital VCO output performs better than those with frequency output (variant "frequential") and with sinusoidal output (variant "sinusoidal"), even though much more discrete events need to be processed by the digital simulator. The presented simulation results (Fig. 5 and 6) were obtained with the variant "digital". The "sinusoidal" variant is more precise, but much slower with respect to the "digital" variant (e.g., more than an hour of simulation time compared to ca. four minutes for case (a)), while its spectra

[1]The second design case is, in its specification and thus its resulting simulation performance, close to the first one.



(a) $f_c = 868\,\text{MHz}$, $MI = \{2, 1\}$, $DR = 100 kbit/s$



(b) $f_c = 2.45\,\text{GHz}$, $MI = 0.5$, $DR = 1 Mbit/s$

Fig. 6. Eye diagram of the FSK transmitter output frequency

and eye diagram are very close to those of the "digital" variant. This sufficient precision and high simulation performance makes the RF_TRX library models very useful for the top-down architectural exploration and bottom-up verification phases. For comparison, a transistor level simulation of the whole PLL (excluding Σ-Δ, modulator, and quartz oscillator blocks) has been tried. For a 10 µs transient simulation, the overall elapsed CPU time exceeded 24 h and after termination it was not possible to access the simulation results.

IV. Conclusions and Outlook

This paper has presented a methodology for modeling a complex RF system, namely a transceiver including a frequency synthesizer and the modulation for the transmitter part. In a first step, the behavior of the principal system components has been specified. Then, the corresponding models have been designed, implemented, and tested individually. Finally, the full system model has been assembled, thoroughly tested, and the simulation results presented. The development of the VHDL-AMS model library RF_TRX helped to establish best practices regarding communication between model developers and RF designers as well as organization and documentation of the models. Their application ensures a maximum flexibility, reusability, validity, and maintainability of the models. This approach shows that a complex RF system can be simulated rapidly and precisely by doing the right abstraction choices. The results show that such VHDL-AMS system simulations can provide the RF designer essential information, e.g., related to the spectral density or the eye diagram that transistor-level simulations cannot give in such a short time.

Future working directions include the consideration of different noise forms and the modeling of power consumption in the models as well as the development of new component models for the library to increase its coverage in the long term on the whole transceiver chain.

TABLE I

Simulation performance comparison of different FSK transmitter model variants run with ADVance MS 2008.2 [12] ($eps = 10^{-6}$) on an Intel Core 2 Quad 2.66 GHz CPU, 4 MB cache, 4 GB RAM, Linux 2.6.9 x86_64.

(a) $f_c = 868\,\text{MHz}$, $MI = 2$, $DR = 100\,\text{kbit/s}$, $t_{stop} = 4\,\text{ms}$

VCO output	h_{max}	CPU time	Steps	Events
digital	1 µs	230 970 ms	2 637 297	44 173 736
frequential	4 ns	281 240 ms	3 795 161	6 154 945
sine	100 ps	4 598 380 ms	42 086 453	30 285 926

(b) $f_c = 2.45\,\text{GHz}$, $MI = 0.5$, $DR = 1\,\text{Mbit/s}$, $t_{stop} = 400\,\mu\text{s}$

VCO output	h_{max}	CPU time	Steps	Events
digital	100 ns	118 550 ms	911 153	14 935 101
frequential	500 ps	144 410 ms	1 889 599	4 145 715
sine	40 ps	1 232 120 ms	11 417 132	11 016 978

References

[1] G. Gielen, "CAD tools for embedded analogue circuits in mixed-signal integrated systems on chip," *IEE Proceedings – Computers and Digital Techniques*, vol. 152, no. 3, pp. 317–332, 2005.

[2] J. Ravatin, J. Oudinot, S. Scotti, A. Le-Clercq, and J. Lebrun, "Full transceiver circuit simulation using VHDL-AMS," *Microwave Engineering*, pp. 29–33, May 2002.

[3] N. Milet-Lewis, G. Monnerie, A. Fakhfakh, D. Geoffroy, Y. Herve, H. Levi, and J. J. Charlot, "A VHDL-AMS library of RF blocks models," in *Proceedings of the Fifth IEEE International Workshop on Behavioral Modeling and Simulation (BMAS) 2001*, Santa Rosa, CA, USA, 2001.

[4] R. Frevert, J. Haase, and R. Jancke, *Modeling and Simulation for RF System Design*. Berlin: Springer, Dec. 2005.

[5] *CommLib RF VHDL-AMS Library: Release AMS 2008.2*, Mentor Graphics Corporation, 2008.

[6] T. Mähne and A. Vachoux, "ModelLib: A web-based platform for collecting behavioural models and supporting the design of AMS systems," in *Advances in Design and Specification Languages for Embedded Systems: Selected Contributions from FDL'06*, ser. ChDL, S. A. Huss, Ed. Dordrecht, The Netherlands: Springer Verlag, Jul. 2007.

[7] J. Rogers, C. Plett, and F. Dai, *Integrated Circuit Design for High-Speed Frequency Synthesis*. Artech House, Jan. 2006.

[8] *An Analysis and Performance Evaluation of a Passive Filter Design Technique for Charge Pump PLL's*, National Semiconductor, Jul. 2001, application note 1001.

[9] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge University Press, 1998.

[10] J. E. Chen, "Modeling RF systems," The Designer's Guide Community, Tech. Rep., 2005. [Online]. Available: http://www.designers-guide.org/

[11] N. C. Bazungula, "Development of a high-level VHDL-AMS simulation model of an RF transmitter based on a direct modulation architecture," Master's thesis, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, Jan. 2009.

[12] *ADVance MS User's Manual: Release AMS 2008.2*, Mentor Graphics Corporation, 2008.