Automatic Behavioural Model Calibration for Efficient PLL System Verification

Ayman Mounir Ahmed Mostafa Maged Fikry
Mentor Graphics Egypt
51 Beirut St. Heliopolis, 11341, Cairo, Egypt
ayman_mounir,ahmed_mostafa,maged_fikry@mentor.com

Abstract

Behavioural models selected from a predefined library are automatically calibrated against transistor-level blocks from a gigahertz-range PLL undergoing verification. The calibrated behavioural models simulate at 10 to 200 times the speed of the target blocks with insignificant loss of accuracy. The technique shrinks the overall simulation time of the assembled PLL by a factor of 120. We rely on a set of carefully qualified, detailed behavioural models, written in VHDL-AMS, each with a custom calibration plan.

1. Introduction

Design verification now dominates the cost of system-on-chip, SoC, design. It is not uncommon on large design projects for verification engineers to outnumber designers in a ratio of 3 to 1 [1]. Devices combining analog and digital elements are particularly challenging. Simulation of detailed transistor level models of a large SoC consumes so much time that it is becoming impractical to provide adequate test coverage. Often digital and analog portions are simulated in isolation, then assembled with very little integration testing. The result is expensive failures and multiple fabrication turns.

A mixed-signal behavioural model of a transistor-level block is an abstract, simulatable representation that exhibits the characteristics of most interest to the designer while suppressing irrelevant physical detail. Suppressing detail speeds simulation. The designer can accelerate simulation runs by replacing one or more transistor-level blocks with behavioural model equivalents. The test plan that judiciously incorporates behavioural models will allow more simulation runs and hence superior coverage. We call this test strategy bottom-up verification [4,5,6].

In spite of the potential benefits, these barriers slow the adoption of behavioural modeling in the verification flow:

1. Although it is easy to develop idealized system-level behavioural models, it is far more difficult to create models that exhibit an appropriate set of physical effects and suppress the rest. System-level models are useful in a top-down design flow, but the more detailed models are required for bottom-up verification.

2. A detailed mixed-signal behavioural model must be carefully qualified if the verification engineer is to rely upon it for critical testing. Qualifying a model is different in important ways from the normal task of a verification engineer. A separate set of skills is required.

3. A detailed mixed-signal behavioural model will have many generic parameters to specify such matters as pole positions, loading impedances, and operating frequencies. Each instance of the model is particularized by specifying fixed values for the parameters. An instance of a model used in a bottom-up verification strategy must be tuned, or calibrated, by the appropriate selection of parameter values to match the transistor-level block it is intended to replace.

The construction and qualification of a suitable library of ideal and detailed behavioural models, is addressed in our earlier work [2,3,8]. This paper addresses the third problem.

Section 2 discusses the problem of calibration and two possible solutions. In section 3, we illustrate the overall strategy by application to a PLL in the design of a production chip. The selection and calibration of the VCO and Charge Pump in the PLL are described in sections 4 and 5. Section 6 concludes with an evaluation of the accuracy and speed of the calibrated model and a discussion of the trade-offs involved.

2. Behavioural Model Calibration

Behavioural model calibration, BMC, is the process deployed to quantify the values of the parameters of a behavioural model, in order to render a model response close to the actual circuit performance, within acceptable
tolerance, in specific operating ranges [9]. We will refer to the transistor level block which is the calibration standard as the “design-under-test” (DUT). The main goal of developing an automatic calibration system is to accelerate the deployment of behavioral models in the bottom-up verification of AMS chips.

Seeking a tool to automatically calibrate an arbitrary behavioral model, without knowing enough details about the model’s internal equations and modelling techniques, should somehow or another be based on parameters-search and fitting techniques, or in other words should be based on optimization techniques.

Optimization based calibration treats the behavioral model as a “black box”. It is based on an iterative search of the space defined by the parameters. The objective of the search is to minimize the distance, measured by some appropriate metric, between each output of the transistor-level block and the corresponding output of the behavioral model. The search is conducted using repetitive simulations of a suitable selection of test cases. The advantage of the iterative search is that it can be conducted without detailed information concerning the internal organization of the behavioral model. There are several disadvantages. An iterative search can be very time consuming. There is no guarantee it will find a best fit. The blind search may attempt to use a parameter vector that causes the behavioral model to fail to converge [12], or to issue error messages.

The approach we advocate here depends on detailed knowledge of the model source code of each element of a library of qualified behavioural models. Each parameter is extracted with a purpose-built simulation and analysis plan using techniques similar to those an analog designer might use to characterize a transistor block.

3. System Description

A block diagram of the PLL system at hand is shown in Fig. 2. This PLL was designed in a 0.13µm CMOS process. The voltage controlled oscillator (VCO) produces a low jitter output signal of 2.5GHz. Two successive dividers producing a divide-by-100, allow the VCO to operate at 100 multiples of the reference clock provided by a chip crystal oscillator running at 25MHz. The dividers output signal is compared to that of the oscillator by a phase frequency detector (PFD). A Charge Pump (CP) is used to convert the digital outputs of the PFD into an analog signal which is fed to the VCO after being filtered by a loop filter (LF), for filtering out the noise of the CP and for stabilizing the loop.

3.1 The System Model

The PLL system is structurally modelled: each transistor level block in Fig. 2 is replaced by its appropriate behavioural model from the library. The following subsections briefly describe the modeling of each block of the system.
3.2 PFD

Fig. 3. Structure of the PFD model

The library model chosen for the PFD is a structural representation of the actual circuit, which is composed of a pair of D-flip flops and a NAND gate. These circuit elements are replaced by their equivalent library models, which account for the propagation delays from input to output. These are purely digital VHDL models. Analog-to-digital and digital-to-analog converters are inserted at their interfaces.

3.3 Charge Pump

Fig. 4. Charge pump model when transistors are in saturation

The Charge Pump model chosen from the library is a controlled current source. It takes into account the mismatch between the source and sink currents. The rising and falling characteristics of these currents are accounted for as well. Fig. 4 illustrates the modelling approach of the Charge Pump model. It is modelled as two transistors, one pumping current and the other sinking it following the UP and DN controls, generated by the PFD. These behavioural transistors exhibit linear, saturation and off regions of operation. In the linear region the transistor behaves as a linear resistor. In the saturation region it behaves as a current source in parallel with a resistor, while in the off region its behaviour approximates that of an open circuit.

3.4 Loop Filter

The loop filter can be kept on its transistor-level description or replaced with its equivalent discrete RC components. Its transfer function representation is not adequate for the behavioural simulation of this type of PLL [10]. For the system at hand, the loop filter is kept on its transistor-level. It is a golden merit to have the possibility to mix abstraction levels within a single simulation session [11].

3.5 VCO

The library model chosen for the VCO mainly describes the relationship between the output frequency and the control voltage through the following equation:

$$\omega_{vco} = K_v(V_{in} - V_{f0}) + K_{vv}(V_{in} - V_{f0})^2 + K_{vvv}(V_{in} - V_{f0})^3 + 2\pi f_0$$

(1)

Where, $\omega_{vco}$ is the output frequency from the VCO in rad/s, $V_{in}$ is the input control voltage, $f_0$ is the VCO centre frequency, $V_{f0}$ is the control voltage generating $f_0$. $K_v$, $K_{vv}$ and $K_{vvv}$ are the coefficients of the third order polynomial that ties the VCO frequency to its control voltage.

Fig. 5. Input/Output impedances in the VCO model

Another important effect modelled is the input/output impedance of the VCO. As shown in Fig. 5 the input resistance and capacitance of the model are shunt to the model’s input. Current to voltage relationship at the model’s input is governed by the following ordinary differential equation:

$$I_{in} = C_{in} \frac{dV_{in}}{dt} + \frac{V_{in}}{R_{in}}$$

(2)

Where $V_{in}$ is the voltage across the input terminal to ground. The output resistance is in series with the differential model outputs and the output capacitance is shunt as shown above. Current to voltage relationships at the model outputs are as follows:

$$I_{R_{out}} = 2 \times \frac{V_{R_{out}}}{R_{out}}$$

(3)

$$I_{C_{out}} = 2 \times \frac{dV_{C_{out}}}{dt}$$

(4)
Where \( V_{R_{out}} \) and \( V_{C_{out}} \) is the voltage across the output resistance and capacitance, respectively.

The library model provided a sine wave output although the circuit had a square wave output. The availability of the model’s source code facilitated its modification to reflect this variation. It is evident and is encountered in many cases that AMS models usually require slight modifications before being deployed in a real-life design. This is due to the nature of analog circuit design which have infinite variations, with each instance having its own identity and speciality among others.

3.6 Divider

The divider is another purely digital block which was described completely in VHDL. It is an important feature of an SoC simulator to have the possibility of mixing analog and digital blocks and tightly solving them simultaneously within a single kernel [11]. The single kernel architecture alleviates the time consumed in inter-process communication, encountered when a co-simulation architecture is deployed.

4. VCO Model Calibration

The VCO model calibration process starts by performing steady state analysis on the VCO transistor level circuit. This analysis should be hosted by an RF simulator [7]. The test bench is shown in Fig. 6, where the control voltage of the VCO (V\(_{\text{centr}}\)) is swept all over its valid range and at each voltage the steady state analysis is used to calculate the respective oscillation frequency.

![Fig. 6. VCO test bench](image)

After the simulation sessions are over and the results are gathered in a pre-defined location, the post processing engine starts analysing the simulation output. A specific fitting technique is deployed to calculate the coefficients of (1), \( K_y, K_{yy} \) and \( K_{yvy} \), to best fit the relationship between the control voltage and the output frequency. The generated third order polynomial (1) was sufficient to approximately fit this relationship, as shown in Fig. 7.

![Fig. 7. Voltage-Frequency characteristics for transistor-level and behavioural VCO](image)

The maximum difference between both curves is less than 0.1% occurring approximately at \( V = 0.7 \), at which \( f_{\text{trans}} = 2.481\,\text{GHz} \) and \( f_{\text{behav}} = 2.479\,\text{GHz} \).

The rest of the parameters of the VCO are extracted using dedicated post processing functions. These functions make use of the internal equations in the VCO model to calculate:

1. \( f_{\text{min}} \), the minimum frequency the VCO can produce, is extracted as the minimum value of the F-V characteristics, which in general can be as shown in Fig. 8.
2. \( f_{\text{max}} \), the maximum frequency the VCO can produce, is extracted as the maximum value of the simulation data.
3. The VCO centre frequency, \( f_0 \), is supplied to the system and the corresponding control voltage \( V_{f_0} \) is extracted accordingly.

![Fig. 8. Extracted parameters of VCO model](image)

It is important to emphasize that the above calibration scenario is automatically carried by the BMC system with minimum designer intervention.
5. Charge Pump Model Calibration

BMC handles the differential Charge Pump shown in Fig. 9. In this topology the output NMOS and PMOS transistors alternate charging and discharging a capacitive load or loop filter.

The Charge Pump is characterized using transient simulation while the PLL is in lock, which means that the PFD detects no phase error, or a constant phase difference, between the reference signal and the feedback signal.

\[
I_{o_p} = I_p - \frac{(V_{hi} - V_{out})_p}{r_{DS_p}} \quad \text{(5)}
\]

\[
I_{o_n} = I_n - \frac{(V_{out})_n - V_{lo}}{r_{DS_n}} \quad \text{(6)}
\]

Where:
1. \((V_{out})_p\) and \((V_{out})_n\) are extracted from the output voltage curve, as shown in Fig. 11.
2. \(V_{hi}\) and \(V_{lo}\) are the voltage rails, as shown in Fig. 9.
3. \(r_{DS_p}\) and \(r_{DS_n}\) are the parallel resistances to the constant current sources \(I_{o_p}\) and \(I_{o_n}\), respectively.

The rest of the model parameters are measured as follows:
1. \(Tr_{ise_p}\) and \(Tr_{ise_n}\) are extracted as the time taken for the capacitive load to charge and discharge, respectively.
2. \(r_{ON_p}\) and \(r_{ON_n}\), which are the resistors modeling the behavioural transistors in the linear region, they are extracted using the equations (7) and (8):

\[
r_{ON_p} = \frac{r_{DS_p} \times (V_{hi} - V_{out})_p}{(I_{o_p} \times r_{DS_p} + V_{out})_p - V_{hi}} \quad \text{(7)}
\]

\[
r_{ON_n} = \frac{r_{DS_n} \times (V_{out})_n - V_{lo}}{(I_{o_n} \times r_{DS_n} + V_{lo} - V_{out})_n} \quad \text{(8)}
\]

The simulation objective sought to check the PLL performance before lock. The acquisition time as well as the overshoot are two performance indices to be calculated [10]. This necessitated a transient simulation for the full PLL for 10\(\mu\)s till it locks on the 25MHz input signal. This simulation on the transistor-level consumed over a week (~172 hours). BSIM3V3 (level 53) transistor models were used.

The behavioural simulation was performed using the calibrated behavioural models of the VCO and the Charge Pump.
Pump. These are the most sensitive blocks of the system, which, together with the loop filter, primarily affect the start-up characteristics of the PLL. The divider and the PDF models were used as simple purely digital VHDL blocks, whose parameters were supplied by the designer. This simulation was completed in 1h 30min: a speed gain of about 120 times. The above simulations were carried on dedicated\textsuperscript{1} SunSPARC10 stations.

6.1 Control Voltage Comparison

![Fig. 13. Control Voltage Comparison](image)

In the above figure it can be clearly observed that the two results (transistor vs. behavioural) are very close, having nearly the same maximum overshoot and acquisition time. The error in the maximum overshoot was less than 4%. After lock, the error in the control voltage is within 0.6%. The following table compares the values of the calculated maximum overshoot at the Charge Pump output and the loop filter output.

<table>
<thead>
<tr>
<th></th>
<th>Charge Pump Output (V)</th>
<th>Loop Filter Output (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavioural</td>
<td>1.030</td>
<td>0.973</td>
</tr>
<tr>
<td>Transistor</td>
<td>1.047</td>
<td>1.007</td>
</tr>
</tbody>
</table>

Table 1. Comparison of the maximum overshoot of the control voltage

6.2 FFT Results

Fig. 14 shows a comparison between the frequency content of the output signal for both the behavioural and the transistor simulation. The FFT for these signals is performed after reaching steady-state, in which the PLL has locked on the reference input signal. It is clear that the two signals have locked to almost the same frequency. The above figure has a peak of 68.1 dB for the transistor-level simulation and a peak of 69.4 dB for the behavioural one, both occurring at almost the same frequency of 2.5 GHz.

7. References


\textsuperscript{1} A special technology was used to insure complete dedication of the machine processor.