On Behavioral Model Equivalence Checking for Large Analog/Mixed Signal Systems

Amandeep Singh and Peng Li
Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX 77843
{amandeep.singh, pli}@tamu.edu

Abstract—This paper presents a systematic, hierarchical, optimization based semi-formal equivalence checking methodology for large analog/mixed signal systems such as PLLs, ADCs and I/O's. We verify the equivalence between a behavioral model and its electrical implementation over a limited, but highly likely, input space defined as the Constrained Behavioral Input Space. Further, we clearly distinguish between the behavioral and electrical domains and define mappings between the two domains to allow for calculation of deviation between the behavioral and electrical implementation. The verification problem is then formulated as an optimization problem which is solved by interfacing a SQP based optimizer with commercial circuit simulation tools. The proposed methodology is then applied for equivalence checking of a PLL as a test case.

Keywords—Analog Circuits, Formal verification, Equivalence Checking, System Verification;

I. INTRODUCTION

The recent advances in semiconductor technology and continued transistor scaling have allowed designers to integrate increasingly more functionality on the same chip. This has resulted in development of complex mixed signal system on chip (SoC) designs. Further, increasingly variable manufacturing process, limited voltage headroom and limited power budgets has lead to increasingly complex analog circuits. This increased design complexity necessitates the development of efficient verification methodologies for mixed signal systems to prevent costly design errors and reduce development time.

The current state-of-the-art verification methodologies and tools have enabled efficient verification of complex digital circuits with millions of gates; however, the same cannot be said for analog or mixed signal circuits. The verification of analog circuits is still largely done manually using SPICE level simulations and is highly dependent on the skills and intuition of the designer. This is a time consuming task. Also, SPICE level verification for large systems involving a number of big mixed signal components involves huge computational complexity. The lack of formal verification for analog/mixed signal blocks often results in non-detection of functional errors in the design leading to re-spins and increase in time to market. While being a significant present research challenge, automated design verification for analog/mixed signal systems is crucial.

Several methods have been proposed for formal verification of analog circuits [1-9]. Most of these methods can be categorized into two categories, equivalence checking and model checking. Equivalence checking compares the output of two different models for a given set of input conditions [1]. For analog circuits, the exact same magnitudes of current and voltage may not be attained, hence, an error bound is defined and the models are said to be equivalent if the error lies within this bound. In [1] the authors provide a good summary of the equivalence checking methods proposed till date. Model checking [1] involves representing the design to be verified in form of a transition system. The specifications of the design are translated to temporal logic formulas. State exploration algorithms are then used to verify if the specifications are satisfied or not. However, model checking algorithms [8] [9] have also achieved limited success for formal verification of analog circuits. As illustrated in Fig. 1, most of these existing methods often require the conversion of a high-dimensional continuous state space to a large discrete equivalent so as to apply Boolean-like verification [4][5][7]. The resulting state explosion limits the application of these methods to toy circuits of very low dimensionality. Further, the inherent approximations in discretization can render these methods practically “informal”. Also, many of the methods proposed have limited practicality as they assume a linear behavior for the circuits under consideration [6].

Recently an interesting boolean-satisfiability based approach has been proposed [3]. The methodology cleverly leverages recent advances in SAT engine for analog verification. However, it also suffers from scalability issues, as it is difficult to solve the satisfiability problem for large complex systems like phase locked loops. Further, the run time increases exponentially as the granularity of the discretized device I-V tables used to formulate the satisfiability problem decreases.

In this paper we propose an optimization based, hierarchical behavioral model equivalence checking methodology that is not necessarily completely formal, but yet systematic and applicable to large designs such as PLLs, ADCs and I/O’s. We use behavioral modeling (e.g. Verilog AMS) as a system verification vehicle. The proposed methodology facilitates feasible behavioral model equivalence checking under the following system context. We assume that the desired system behaviors are “encoded” in a set of block-level behavioral models, or the reference system behavioral model (RSB). Hence, the desired system performance specifications are also reflected in the simulated performances of the RSB. A given detailed electrical (circuit) implementation, e.g., represented by a set of (extracted) block-level SPICE netlist, is checked (verified) against the RSB on an individual block basis. Either, the implementation is deemed as “equivalent”, or the check is inconclusive due to the conservative nature of the check. In addition to the aforementioned equivalence checking against a given “golden” RSB, the proposed work also serves an intrinsically related purpose: compare an existing electrical-level design implementation against its corresponding behavioral...
model so as to provide guidance for behavioral modeling. The proposed methodology has several key characteristics:

- System-level behavioral simulations are used as a basis to derive a limited but sensible set of input stimuli for verification. Inherent abstraction in behavioral modeling, which contributes to the deviation of the behavioral model from its electrical counterpart, is specifically targeted in our verification; such modeling abstraction is mathematically characterized by defining two signal domains and mapping functions between them.

- Equivalence checking is formulated as a constrained optimization problem and solved by interfacing behavioral and SPICE-level simulators that contrast the behavioral model with the SPICE netlist.

- System equivalence checking is broken into individual block-level checks, and hence performed hierarchically; this makes the approach scalable for large designs.

The rest of the paper is organized as follows: Section II gives the problem description; Section III provides the detailed description of the mapping functions and signal domains used; Section IV formulates the verification problem as an optimization problem. Finally, Section V shows the results of our verification methodology using a Phase Locked Loop as a test case.

II. PRELIMINARIES

The proposed semi-formal, hierarchical, optimization based equivalence checking methodology aims at verifying equivalence between the system behavioral model called the ‘reference system behavioral model’ (RSB) against detailed electrical, i.e. transistor level implementation.

A. Problem Definition

We view that the input and output signals to/from each block of the reference system behavioral model, hitherto referred to as the Behavioral Signals, belong to a behavioral signal domain $\Omega_B$. Similarly, we define an electrical signal domain $\Omega_E$ which contains the input and output signals to/from each block of the electrical transistor level implementation. To enable verification of large analog/mixed signal designs we also define a limited, but most likely, input behavioral signal space for the behavioral models called the Constrained Behavioral Input Space (BIS). The mechanics of generating the BIS for each block are discussed in section II.B. The equivalence check is then performed not over the universe of all possible inputs in the behavioral signal space, instead, only w.r.t. the chosen set of sensible input stimuli as defined by the Constrained Behavioral input space.

For each block-level behavioral model, and a given behavioral input and the resulting behavioral output, we perform equivalence check by asking the essential question: does the corresponding block-level electrical model (spice netlist) retain the same (behavioral) input and output correspondence?

The above question would have been trivial to answer if both models operate in the same signal domain. However, the fact that such equivalence check has to be conducted across two signal domains introduces complications. As such, we define two mapping functions $f_{B,E}$ and $g_{E,B}$ to map the signals from the behavioral signal space to the electrical signal space and vice-versa. The function $f_{B,E}$ is a one to many mapping while the function $g_{E,B}$ is a many to one mapping. The generation of these mapping functions is dependent of the module being verified and is explained in Section IV.

The obtained behavioral BIS are mapped to $\Omega_E$ using the mapping $f_{B,E}$ which is then used to drive the verification on an individual block basis as shown in Fig. 2. Each behavioral input in the BIS is mapped into a set of detailed electrical inputs which are then used to simulate the electrical transistor (Spice) level circuit. The resulting electrical outputs are mapped back to the behavioral domain to compare with the reference behavioral output of the behavioral model. The maximum discrepancy of the two is used as metric to judge the equivalence.

B. Generation of Constrained Behavioral Input Space

To allow for a scalable verification methodology we recognize that the inputs to a specific circuit block are constrained by the structure of the entire design, i.e. the inputs to each block in the model cannot be any arbitrary input, and instead, only a subset of them (Figure 3). This constrained behavioral signal space for the behavioral model forms the BIS. To generate the BIS for each block in the behavioral model, the RSB is simulated using a set of typical system-level simulation stimuli, such as the ones that are used to measure system design specs (e.g. lock-in time for PLL etc.). Upon the completion of each system-level simulation, the behavioral input (as well as the corresponding behavioral output) is retained for each circuit block. The complete set of such behavioral inputs defines the BIS for the block. In this case, Equivalence Checking essentially checks the electrical implementation against the RSB under the typical input excitations that are employed to measure system design specs. If the equivalence check succeeds, the corresponding design specs of the RSB would be deemed as reflecting those of the actual implementation. The use of the verification allows efficient determination of achieved system performance specifications without resorting to expensive flat (SPICE) simulations of the design. A more complete input space BIS can also be obtained by simulating the RSB with a more comprehensive set of system-
level input stimuli and record the corresponding behavioral inputs appearing at the input to each circuit block. In practice, these system-level inputs can be obtained by using design knowledge or by introducing pseudo-random variations to typical inputs. In this case, a higher coverage in verification will be resulted as a larger set of input excitations are included in the verification process.

III. SIGNAL DOMAINS & MAPPING FUNCTIONS

As described in the previous section, we use system-level behavioral simulations to generate a behavioral input set (BIS) for each circuit block. Then for each behavioral input $I^\prime_B$ (in the BIS) and the corresponding behavioral output of the block, $O^\prime_B$, the electrical implementation or a SPICE-level transistor model of the block is checked against the behavioral block model for equivalence. As illustrated in Fig. 4, such equivalence check is performed across two different signal domains: behavioral ($\Omega_B$) vs. electrical ($\Omega_E$). In this section we highlight the key differences between the behavioral and the electrical domains. The mapping functions used to transform the signals from one domain to another are also explained.

A. Behavioral vs. Electrical Domains

The behavioral domain ($\Omega_B$), characterized by the behavioral signal space, is essentially an abstract form of the actual electrical domain ($\Omega_E$). The signals in the behavioral domain are abstract versions of the electrical signals and are generated by removing some details from the electrical signals. For example, let us consider two models, an electrical model and a behavioral model. We apply a sinusoidal input waveform to both the models. Further, let us also assume that the behavioral model output only depends on the frequency of the input signal and the time instants at which the waveform pulse crosses the origin. Then in principle, any signal with an arbitrary waveform shape but identical frequency and zero crossing time should produce the same behavioral output as the sinusoidal waveform. However, the same shall not be true for the electrical output. This difference between the behavioral output and the electrical output comes from the fact that while the electrical input is a sinusoidal waveform the actual behavioral input signal simply abstracts away the waveform shape information while only preserving the frequency and zero-crossing times.

To further illustrate the differences between electrical and behavioral domains, especially in relation to analog/mixed signal systems, let us consider a behavioral model for a voltage controlled oscillator (VCO) as shown in Figure 5. The behavioral output of the module only depends on the time instants at which the phase changes, the low and high output voltage levels. No information about the precise waveform shape is present in the behavioral signal, whereas the same information content is present in the electrical domain output of a VCO.

B. Signal-Domain Mapping Functions

To link the two domains together we define two mappings, $f_{B,E}$: $\Omega_B \rightarrow \Omega_E$ and $g_{E,B}$: $\Omega_E \rightarrow \Omega_B$. With the inherent abstraction in behavioral modeling, $f_{B,E}$ is one-to-many mapping and maps a behavioral signal waveform to a set of electrical realizations; $g_{E,B}$ is many-to-one mapping and abstracts away non-behavioral details from an electrical waveform. Using $f_{B,E}$ we map a single (behavioral) input $I^\prime_B$ to the behavioral model to a set of electrical inputs, $S_{B,E} = \{I_{E1}, I_{E2}, \ldots\} = f_{B,E}(I^\prime_B)$, which are used to exercise the SPICE model in $\Omega_E$ (Fig. 4). The resulting multiple electrical outputs $S_{OE} = \{O_{E1}, O_{E2}, \ldots\}$ are mapped back to $\Omega_B$ via $S_{OB} = \{g_{E,B}(O_{E1}), g_{E,B}(O_{E2}), \ldots\}$ to compare against the reference output of the behavioral model $O^\prime_B$. Note that for a single behavioral input $I^\prime_B$, $I_{E} = f_{B,E}(I^\prime_B)$ defines the electrical input space over which the electrical implementation needs to be checked for equivalence. On the other hand, since the reference behavioral output, $O^\prime_B$ is behavioral, the outputs of the electrical implementations are mapped back to the $\Omega_B$ via $g_{E,B}$ for comparison.

To illustrate how the two mapping functions are generated in practice, let us consider part of the behavioral model of a phase locked loop comprising of the charge pump and a module containing the filter and a VCO (Fig. 6). The behavioral output of the charge pump may contain only idealized current pulses which act as behavioral inputs for the filter & VCO module. Note that these output signals are in the behavioral domain and only have essential modeled behavioral characteristics of the output signal. $f_{B,E}$ basically maps the behavioral output signal to the electrical domain by adding the un-modeled electrical details, say in this case, the rise time and the fall time of the output current pulse. Note that for each behavioral input signal multiple electrical signals are produced. Similarly, the reference behavioral output signal of the filter & VCO module, $O^\prime_B$, only contains the essential behavioral characteristics that are modeled in the output function of the VCO, which for a model shown in Figure 5 shall be the level crossing time points. To compare with this reference $O^\prime_B$, $g_{E,B}$ basically maps the detailed electrical output waveforms produced by the SPICE-level block model to the behavioral

---

**Figure 5**: Part of behavioral model for a VCO.

module vco (in,out)
    ...
    analog begin
        freq = (V(in)-Vmin)*\alpha+Fmin // Simple Linear model for VCO frequency
    //phase calculation
        phase = 2*M_PI*idmod(freq,0.0,1.0,-0.5)
    //generation of output voltage
        V(out) <+ transition(n? Vlo: Vhi, td,tr,tf)
    end

---

**Figure 4**: Signal mappings between the behavioral and electrical domains.
domain. In the present example, the electrical outputs of the corresponding SPICE-level VCO net-list shall be simply mapped back to the behavioral domain by extracting the level-crossing time stamps. In general, $g_{E,B}$ is many-to-one since multiple electrical signals can have the same extracted behavioral features.

In principle, mapping functions $f_{B,E}$ and $g_{E,B}$ are module dependent. In particularly, as illustrated in Fig. 6, $f_{B,E}$ for the block under check shall be constructed to reflect the behavioral abstraction embedded in the output function of the preceding (driver) behavioral model. On the other hand, $g_{E,B}$ effectively extracts from an electrical output the behavioral characteristics that are specified in the output of the behavioral model under check.

IV. OPTIMIZATION BASED EQUIVALENCE CHECKING

As described in previous sections our verification methodology involves generation of system level behavioral stimuli, mapping each behavioral input stimulus to a set of detailed electrical inputs which are then used to simulate the SPICE level transistor net-list. At the output, we map the set of electrical signals produced to the behavioral domain, which are then compared with the corresponding behavioral outputs from the RSB to verify equivalence between the two implementations. In this section we formulate the above comparison as a maximization problem. The optimization problem may be solved using any ‘simulation based optimizer’, i.e. any available optimization solver which does not necessarily require a closed form expression for calculating the objective function. In this paper we used DONLP2 [10][11], a sequential quadratic programming (SQP) based optimization engine for the same. DONLP2 was interfaced with CADENCE Spectre to allow computation of the objective function using actual spice level simulations.

For a given behavioral input $I^*_B$, the behavioral model produces $O^*_B$ at the output (Figure 4). To verify whether or not this input-output correspondence is retained in the electrical implementation, we ask the following question: for all electrical input signals that have the behavioral characteristics specified by $I^*_B$, will the corresponding electrical outputs maintain the same behavioral characteristics specified by $O^*_B$? For every circuit block, we perform the above equivalence check for each behavioral input in its BIS. An electrical implementation is deemed as equivalent to the system behavioral model if and only if all such checks are passed. We formulate the above as a maximization problem.

We parameterize the non-behavioral electrical features not modeled in a behavioral input, such as finite rise/fall times and signal shapes, by introducing additional electrical feature parameters. We denote these electrical feature parameters as $p_E$. Such parameterization mathematically constructs the mapping function $f_{B,E}$. The mapped electrical input set $S_E = \{ I_E^1, I_E^2, ... \} = f_{B,E}(I^*_B)$ defines a constrained electrical input excitation space over which the SPICE net-list needs to be compared with the behavioral model. We formulate this task formally as a maximization optimization problem in $S_E$ and seek to obtain the maximum deviation $\epsilon_{max}$ from the reference behavioral output $O^*_B$:

\[
\begin{align*}
\max_{p_E} \epsilon &= h_{err}(O_E(p_E), O^*_B) \\
\text{subject to:} \quad p_E &\leq p_E^* \\
I_E &= f_{B,E}(I^*_B, p_E) \\
O_E(p_E) &= g_{E,B}(O_E(p_E)) \\
O^*_B &= O_E(p_E^*)
\end{align*}
\]

Equation (1) defines the objective function, which is an error function $h_{err}$ specifying the derivation between the mapped electrical output and the reference behavioral output $O^*_B$. For instance, if $O^*_B$ and $O_E(p_E)$ are represented as vectors of sampled signal values, $L_2$ vector norm can be used to define the error function: $h_{err} = \| O_E(p_E) - O^*_B \|_2$. In practice, the definition of $h_{err}$ is model dependent and is dependent on the functionality of the block. For example, for the VCO behavior model shown in Figure 5, it may be the $L_1$ vector norm of level crossing time stamps, while for a charge pump it may be $L_2$ vector norm of the vector measuring the charge injected into the system at different time points. Equation (2) defines the bound on the electrical input parameters. Equation (3) maps $I^*_B$ to an electrical input signal by adding electrical features specified by $p_E$. Equation (4) maps an electrical input applied to the SPICE-level model to the corresponding electrical output; this mapping is realized by running circuit (SPICE) simulation. Finally, (5) maps the electrical output to the behavioral domain by using $g_{E,B}$.

The proposed optimization-based equivalence-check flow is shown in Figure 7, where an optimizer (DONLP2) is employed to search for the maximum deviation $\epsilon_{max}$. If $\epsilon_{max}$ is less than a user-defined tolerance, the equivalence check is deemed as passed; otherwise, a failure is reported. At the inner loop of the
optimization, the circuit simulator, CADENCE Spectre, is interfaced to provide the mapping in (4).

To make the above methodology more robust and conservative in nature, we also implemented a slightly modified form of the above flow. The modification was done to allow the proposed verification flow to merge conveniently with the existing commercial circuit simulation software like CADENCE Spectre and HSPICE. Although the behavioral output signal at any point should/does not depend on the non-modeled electrical details in the behavioral input signal, practical circuit simulators do not distinguish between the behavioral signals and the electrical signals. The circuit simulator treats the behavioral input and output in the same way as they treat the electrical signals. This anomaly in simulators may sometimes lead to unexpected results. To avoid any such occurrences and enable a conservative check we modify the above flow shown in Figure 7 slightly. The modified flow is shown in Figure 8. As shown in figure 8, at inner loop of optimization, in addition to simulating spcie level net-list we also simulate the behavioral model for the block under verification with the same electrical domain input signal. The electrical outputs are then mapped back to the behavioral domain and maximum deviation between the electrical and behavioral domain outputs are computed in the same manner as before.

V. EXPERIMENTAL RESULTS

The proposed methodology was implemented using C language and was applied for equivalence checking between verilog AMS based behavioral model of a phase locked loop (PLL) and its electrical implementation (CADENCE Spectre net-list). The block diagram of the PLL used is shown in figure 9. To generate the behavioral input space (BIS) for each block in the behavioral model, the reference behavioral model was simulated using a typical system-level simulation setup used to calculate the lock-in time of a PLL. The reference input signal was a pulse of 10.9MHz and the voltage signal ‘Vcontrol’ was used to modify the divider ratio of the PLL from 150 to 100 at a time instant of 3μs. The verification of the entire system was performed in a hierarchical manner by dividing the system into three modules, loop filter and voltage controlled oscillator (VCO), charge pump and phase detector.

A. Loop Filter and Voltage Controlled Oscillator

The behavioral input to the block composed of the loop filter and the VCO consists of idealized current pulses from the charge pump. The electrical implementation of the block is shown in Figure 10. Figure 11 depicts the spectre simulation results of the current pulses generated by the charge pump from 0.32μs to 0.66μs. The results depict the presence of spikes in the output current whenever the current waveform amplitude changes suddenly. To map the idealized behavioral current pulse waveforms into electrical equivalent signals four electrical feature parameters, trise (t_r), tfall (t_f), peak_pos (p_pos) and peak_neg (p_neg) were defined. trise (t_r) and tfall (t_f) represent the rise time and fall time of the current waveforms, and peak_pos (p_pos) and peak_neg (p_neg) refer to the peak amplitudes of the current spikes generated in the output waveform. The behavioral output of the block, i.e. the VCO behavioral output, is dependent only on the level crossing time instants (Figure 5). Thus, the electrical output signals are mapped back to behavioral domain by simply extracting the level crossing time instants. Based on the above, the optimization problem was formulated as under:

\[
\max \epsilon = ||t - t_0^*||_2^2
\]

subject to:

\[1 \leq t_r \leq 50\text{ns}
\]

\[1 \leq t_f \leq 50\text{ns}
\]

\[3 \leq p_{- neg} \leq 20\mu A
\]

\[3 \leq p_{+ pos} \leq 20\mu A
\]

where \(t\) and \(t_0^*\) refer to the level crossing time instants obtained from the electrical and behavioral outputs respectively.

The above optimization problem was solved for three different behavioral models of the VCO. The output frequency versus control voltage plots for the three different VCO models are shown in Figure 12. Model A closely resembles the VCO characteristics across the entire control voltage range whereas models B & C are only linear approximations to the VCO output
frequency characteristics. Table 1 below shows the maximum error obtained for each behavioral model, values of the electrical parameters added at that instant, equivalence decision of the methodology, and the total runtime required for optimization. As expected, the maximum error is least for model A and is the only model for which equivalence test is successful.

Table 1: Equivalence check for VCO and filter block

<table>
<thead>
<tr>
<th>Model</th>
<th>Maximum Error</th>
<th>Rise Time</th>
<th>Fall Time</th>
<th>Peak_pos</th>
<th>Peak_neg</th>
<th>Equivalence</th>
<th>Runtime (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.5452e+02</td>
<td>1.999ns</td>
<td>50ns</td>
<td>3μA</td>
<td>20μA</td>
<td>Yes</td>
<td>8700</td>
</tr>
<tr>
<td>B</td>
<td>3.500e+03</td>
<td>1.999ns</td>
<td>50ns</td>
<td>3μA</td>
<td>20μA</td>
<td>No</td>
<td>14280</td>
</tr>
<tr>
<td>C</td>
<td>8.03734e+03</td>
<td>1.999ns</td>
<td>50ns</td>
<td>3μA</td>
<td>20μA</td>
<td>No</td>
<td>17690</td>
</tr>
</tbody>
</table>

B. Charge Pump

The behavioral input to the charge pump (CP) consists of two digital like voltage pulses ‘up’ and ‘down’ controlling the output current. The electrical implementation of the charge pump is shown in Figure 13. To map the behavioral input signals into equivalent electrical waveforms, two electrical feature parameters, trise (t_rise), and tfall (t_fall) were defined. The electrical output of the charge pump consists of the short duration current pulses (Figure 11). Since the total charge injected into the system is the most important parameter for the charge pump, the output current was integrated to calculate the net charge introduced by the charge pump. The total charges introduced by the behavioral and electrical implementations at different instants of time were then compared to calculate the error function. Based on the above, the optimization problem was formulated as under:

\[
\max \varepsilon = ||q - q_B^*||^2
\]

\[
subject\ to:
\]

1. \[1 \leq t_r \leq 50\,\text{ns}\]
2. \[1 \leq t_f \leq 50\,\text{ns}\]

where \( q \) and \( q_B^* \) are vectors consisting of the total charge injected into the system by the electrical and behavioral models at different time instants. To verify the methodology for charge pumps, two different behavioral models were used, with one closer to the electrical implementation than the other (Figure 14). Behavioral model ‘A’ took into account the current mismatch between the ‘up’ and ‘down’ current while model ‘B’ simply neglected this difference and modeled both the current sources identically. The maximum deviation between the electrical and behavioral models was calculated by solving the optimization problem in (11). Table 2 below shows the maximum deviation between the electrical and behavioral signals and the equivalence decision of the proposed methodology. As expected, the maximum deviation in model ‘A’ is less than the maximum deviation in model ‘B’ and equivalence test is successful only for model A.

Table 2: Equivalence Results for the Charge Pump

<table>
<thead>
<tr>
<th>Model</th>
<th>Maximum Error</th>
<th>Rise Time</th>
<th>Fall Time</th>
<th>Match</th>
<th>Run Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.43297e+01</td>
<td>50ns</td>
<td>2ns</td>
<td>Yes</td>
<td>31.6</td>
</tr>
<tr>
<td>B</td>
<td>2.14668e+01</td>
<td>50ns</td>
<td>50ns</td>
<td>No</td>
<td>30.4</td>
</tr>
</tbody>
</table>

C. Phase Detector

The behavioral input to the phase detector (PD) consists of input reference voltage and the output voltage waveform (behavioral) of the frequency divider. The behavioral input voltage was mapped to electrical domain by adding the rise and fall time parameters to the input behavioral waveforms. Further, similar to the VCO output waveforms, the electrical domain voltage waveforms were mapped back to the behavioral domain as follows:

Figure 11: Spectre simulation: charge pump output current.

Figure 12: Frequency versus control voltage for the three VCO models.

Figure 13: Electrical implementation of charge pump.
by simply extracting the time instants at which the output voltage crossed the \( V_{dd}/2 \) value. The optimization problem for the phase detector was formulated as under:

\[
\max \varepsilon = \| (t_{up} - t_{up,B}^*) * 1e6 \|^2_2 + \| (t_{dn} - t_{dn,B}^*) * 1e6 \|^2_2
\]

subject to:

\[
1 \leq t \leq 50ns
\]

\[
1 \leq t \leq 50ns
\]

where \( t_{up}, t_{up,B}, t_{dn} \) and \( t_{dn,B} \) refer to the \( V_{dd}/2 \) crossing time instants obtained from the electrical and behavioral output for the ‘up’ and ‘down’ voltage respectively. The above optimization problem was solved for the phase detector and the maximum deviation of 1.48383e+01 was achieved for \( \text{trise}=\text{fall}=50\text{ns} \). Since, the deviation was less than the maximum threshold, the models were deemed equivalent.

D. Verification of the proposed methodology

To verify that the proposed methodology correctly identified the equivalence between the behavioral models and their electrical implementations, system level properties of the PLL such as locking time were calculated from both the behavioral and electrical implementations. The equivalence between the behavioral models and their electrical counterparts. A locking time of 1.03 \( \mu s \) was achieved.

Similarly, model A was used to calculate the above properties as model A was deemed as equivalent by the verification methodology. Similarly, model A of the charge pump block (Table 2) was used in the above calculations. Figure 15 shows the frequency of the output signal, \( V_{out} \), as obtained from the reference behavioral model and when calculations. Figure 15 shows the frequency of the output signal, \( V_{out} \), as obtained from the reference behavioral model and when calculations.

VI. ACKNOWLEDGEMENTS

This work was supported in part by the Semiconductor Research Corporation and Texas Analog Center of Excellence.

VII. CONCLUSION

In this paper we presented an optimization based, hierarchical behavioral model equivalence checking methodology for large Analog/Mixed Signal designs such as PLLs, ADCs and I/O’s. The verification methodology is not necessarily formal, but yet systematic and practical. It recognizes that the inputs to any specific circuit block in a system are constrained by the structure of the entire design, and hence, verification is done only on a selected set of inputs rather than the universe of all possible arbitrary inputs. The proposed methodology is applied for verification of a Phase Locked Loop as a test case.