Exploration of Real Value Modelling for Complex Mixed Signal Verification

Student Name: Pallavi Das

RollNumber: MT13131
June, 2015

Indraprastha Institute of Information Technology
New Delhi

Under the Supervision of
Dr. Sujay Deb (IIITD)
Mr. Hardik Parekh (ST Microelectronics)

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Electronics and Communication Engineering,
with specialization in VLSI and Embedded Systems

This research has been done with the collaboration of ST Microelectronics Pvt. Ltd., Greater Noida.

Keywords: Modelling, Verification, Phase locked loop, Equivalence checking, Spice vs Behavioural, design and implementation and Real value modelling

Certificate

This is to certify that the thesis titled **"Exploration of Real Value Modelling for Complex Mixed Signal Verification"** submitted by **Pallavi Das** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *VLSI and Embedded System* is a record of the bonafide work carried out by her, under our guidance and supervision at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Sujay Dab
Indraprastha Institute of Information Technology, New Delhi

Mr. Hardik Parekh
ST Microelectronics, Greater Noida

Abstract

With the demand to have more functionality in today's systems, the high performance SOC will have to further accommodate Analog and Mixed Signal (AMS) designs. Also, due to increasing unpredictability and complexity of such system, circuit SPICE and Fast SPICE simulation cannot deliver a verification arrangement on time. This leads to growing necessity of methodology for accurate and fast verification of AMS designs.

In this dissertation, we have presented a novel approach for AMS verification which uses well known Real Value Modelling (RVM) concepts. RVM processes floating-point real numbers like analog world, based on discrete events. The developed verification technique in this work makes it possible to behaviourally model analog effects such as supply ramp behavior, PVT variations, using event driven simulators and compatible with existing digital verification techniques. This significantly reduces the verification time for Full Chip Simulations (FCS). Also, the advantages of this approach are illustrated by taking Phase locked loop as an examples.

Acknowledgments

This work would not have been possible without the guidance and support of several individuals who boosted me for completion of this work.
I would like to first thank my advisors Dr. Sujay Deb and Mr. Hardik Parekh, for providing excellent guidance, encouragement throughout the project work. Without his indispensable guidance, this work would never have been a successful one.
Thanks to IIIT-D and STMicroelectronics for providing excellent environment and infrastructure. I would also like to thank all my classmates for their valuable suggestions and fruitful discussions. Last but not the least, I would like to thank my family for supporting me spiritually and emotionally throughout my life.

Pallavi Das

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background & Motivation

Today, the Major part of SoC verification, measured in computer cycles and engineering time, is dedicated to simulation and verification. As complexity increases in modern SoCs, the number and length of simulation runs and the amount of time required to achieve good coverage is enormous, which will leads the verification bottleneck. Furthermore, System-on-Chip (SoC) devices which contained mostly digital circuitry now contain significant analog and mixed signal content commonly referred to as an Analog Mixed Signal (AMS). This escalating complexity poses severe challenges for mixed-signal SoC verification, such as incomplete SoC-level and system-level verification and uncertainties in verification coverage. Earlier AMS verification is done using mixed mode simulation with digital and analog IP in the single design, including their interconnections. Mixed signal simulation provides SPICE-level accuracy for the analog IP, but its simulation performance is not as fast as a pure event driven (digital) simulator. As the complexity of Mixed Signal System on Chip (MSSoCs) increase, verification of MSSoCs requires verifying the analog as well as the digital functional specification of the complete design, in a single verification environment with a stipulated amount of time. Another limitation of Mixed Signal simulators is capacity. The current trend of integrating multiple functions on MSSoCs and the shrinking size of transistors will increase the number of transistors and hence which leads to the difficulty for handling large SPICE (analog) content in mixed simulator. When simulating AMS circuits, the majority of simulation time is used for the analog portion of the design because it takes a long time to solve the complex equations required for this level of

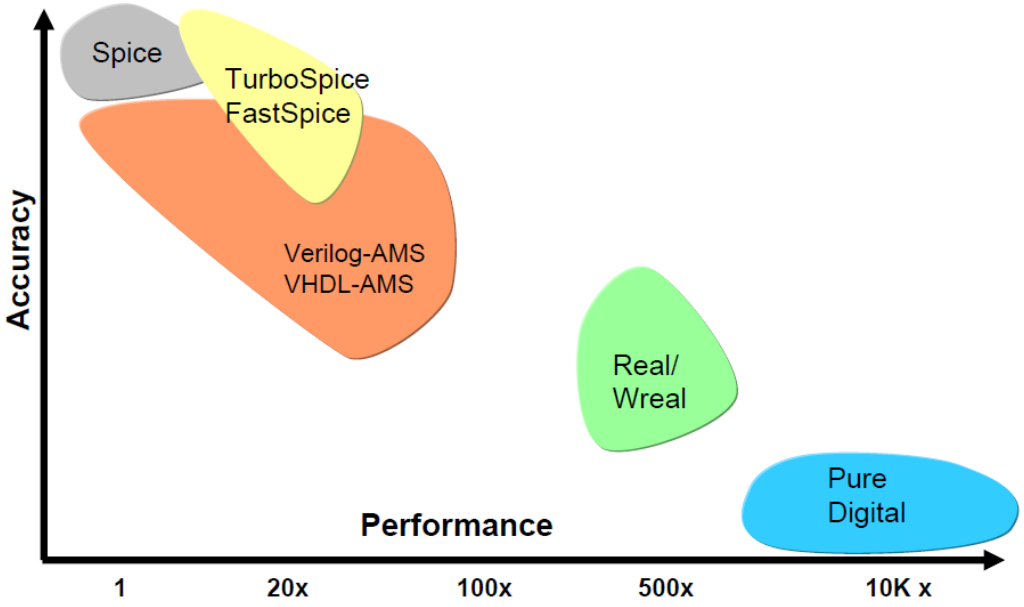

Figure 1.1: Model accuracy versus performance gain for mixed-signal simulation [6]

accuracy, hence simulation speeds are slow. The consequence of this huge simulation time is either less validation during a given validation period or the extension of the validation period itself. The pressure to produce ever more complex chips in ever shorter periods of time compels design teams to find ways to reduce the simulation time of analog circuits.

According to industry estimates, over 60% of SoC design re-spins at 45nm and below are due to mixed-signal errors. To mitigate this, the typical SoC Verification flow involves top-level simulation of components at various levels of abstraction. Functional complexity in terms of modes of operation, extensive digital calibration, and architectural algorithms is now overwhelming the traditional verification methodologies. Simulation at this top-level is extremely costly (both in terms of time and licenses cost) since a significant amount of the SoC is simulated inside the analog engine. By replacing the analog portions of the SoC with functionally equivalent digital models, which do not require the analog engine, we achieve a significant speed-up in simulation performance and reduction in the license cost. Meanwhile, typical analog simulation problems such as convergence issues are totally eliminated.

Figure 1 shows the trade off between simulation accuracy and performance among SPICE, Fast SPICE, analog behavioral models (Verilog-A/AMS and VHDL-AMS), RVM and pure digital simulation. Pure digital simulation can only represent an analog signal as a single logic value,

but it is only sufficient for connectivity checks in mixed-signal SoCs. Spice level simulations are used as golden reference simulation. Fast and Turbo Spice engines reach the same accuracy but can also trade-off some accuracy versus speed (mainly fast spice). Analog behavioral modeling provides a large range of capabilities, reaching from high accuracy to high performance. Especially convergence issues caused by AMS models might slow down the simulation significantly. We know Simulation at this top-level is extremely costly (both in terms of time and effort) since, a significant amount of the SoC is simulated inside the analog engine. Finding a way to reduce the time and expense to verify this SoC, while trading off some accuracy that is not needed at this high level of integration, is extremely valuable. This is the target application of Real Valued Modeling. With the replacement of AMS portions of the SoC with functionally equivalent real value digital models, which do not require the analog engine, can achieve a significant speed-up in simulation performance and reduction in the effort and handshaking problem. Meanwhile, typical analog simulation problems such as convergence issues are totally eliminated. The gain in simulation performance and the reduction in accuracy are highly dependent on the application. This thesis is an exploration of the time domain modeling and simulation of an AMS IP in general, with an initial emphasis on the system level architecture and the simulation environment. The thesis also focuses also on the non-ideality analysis of the IP in various simulation environments, such as phase noise and timestamps errors.

1.2 Problem Statement

Summarizing the previous discussion, the work presented in this thesis can be described as a two fold problem:

1. To design an accurate and efficient model the analog and mixed signal IPs (Phase locked loop) using Real value modeling for improving the performance of verification.
2. To introduce actual noise, jitter and supply ramp up and observe the behaviour with respect to spice in equivalence checking environment.

Hence, by implementing a Real value approach, we balance the performance and accuracy and reduces the overall verification time. The impact of this reduction achieves a significant speed

up in system performance without loss essential features.

1.3 Organization of the Thesis

Chapter 2 discusses about the related work and illustrates the proposed model. Chapter 3 provides the detailed description of Real value Modeling approach, its scope, functionality and design limitation. Chapter 4 explains verification methodologies used to verify PLL in real value approach in subsystem level. Chapter 5 illustrates the simulation results proving the functionality correctness of the proposed real value model with spice netlist for different test cases and analyzes its performance. Finally, Chapter 6 concludes this thesis by outlining my contribution and research, along with the possible scope of our work that can be explored.

Chapter 2

Literature Survey & Our Proposal

2.1 Literature Survey of Related Work

The increasing design complexity of analog and mixed signal system on chip together with the lesser time to market have necessitated the demand for enhancement of formal verification techniques for analog and mixed signal circuits. Mixed Signal and Analog IPs verification has been addressed at several fronts by researchers and industry. In this chapter few recent techniques intended for verification of analog and mixed signal circuits are reviewed. A more complete review of previously proposed techniques and methodologies is given in [23].

Several methods have been proposed for formal verification of analog circuits [23] [7]. These methods can be broadly divided in two methods, equivalence checking and model checking. Equivalence checking is a technique which gives comparison of the output of two different models for a same set of input conditions [23]. For analog circuits, the exact same magnitudes of current and voltage may not be attained, hence, an error bound or threshold is defined so that if the models are said to be equivalent if the error lies within this bound. The authors provide a good summary of the equivalence checking methods proposed till date [7]. Model checking represents the design to be verified in form of a transition system [7]. 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 [4] have achieved limited success for formal verification of analog circuits. Most of the methods often uses the conversion

of a high-dimensional continuous state space to a large digital equivalent so as to apply Boolean-like verification [2] [8]. Recently an interesting Boolean-satisfiability based approach has been proposed [17]. The methodology cleverly give a vision for recent advances in SAT (Satisfiability) engine for analog verification [19]. However, it also suffers from scalability issues, as it is difficult to solve the satisfiability problem for large complex systems like phase locked loops. Again, the run time increases exponentially as the granularity of the discretized device I-V tables used to formulate the satisfiability problem decreases.

Synopsis presented a white paper about modelling language which explains Verilog-AMS model provides a good trade-off between simulation accuracy and speed. But these models are parsed by a mixed-signal simulator, where design code is split internally into a digital portion to be handled by a digital event-driven simulator and an analog portion to be handled by an analog circuit simulator and results issues like convergence, poor performance, and timing violations. Recent studies show that different behavioral model can be used for a single IPs to verify functionality for different application like UPF (unified power format) simulations. Ayman Mounir developed an automatic calibration system is to accelerate the deployment of behavioral models in the bottom-up verification of AMS chips, without knowing enough details about the model?s internal equations and modelling techniques, based on parameters-search and fitting techniques, or optimization techniques between transistor level and corresponding behavioral model [17]. Various methods and strategy have been explored to estimate the methodology for AMS verification with particular attention to computation performance and simulation trade-offs. In [23], the authors classified run-time verification in two different groups: offline and online monitoring. In offline monitoring, the property is only verified after the whole simulation trace is provided, while in online monitoring, the monitoring is interleaved with the simulation process. Due to which, ability to detect violation with very less amount of time. An online monitoring approach was proposed in [18], where template monitors described by linear hybrid automata are used to model the time domain features of oscillatory behavior. In [2], the authors used interval arithmetics to describe the reachable states of the design behavior. The methodology we present in this paper can also be related to the work done in [8], where the authors proposed an online monitoring verification technique of hybrid systems using timed and linear hybrid automate. An IP-based SOC designs, bottom up extraction approaches presented in [2] which gives a picture of building the behavioral models from post-layout simulation results may be the

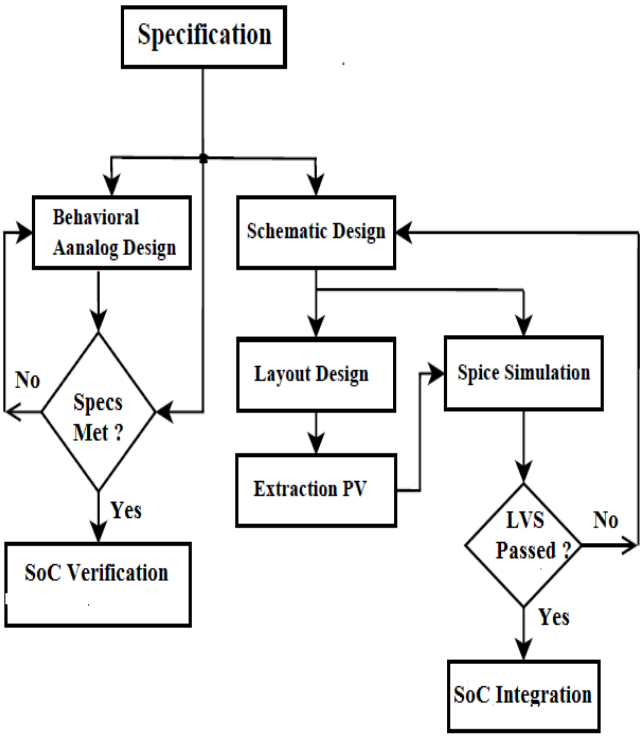

Figure 2.1: Overview of traditional AMS design and verification flow

better way to obtain more accurate models for existing designs.

2.2 Our Proposal

This thesis work proposes anew modeling strategy which overcome the challenges arising with the traditional methods, hence enhancing more reliable verification of analog and Mixed signal IPs. Fig 2.1 shows the verification as well as design flow of AMS designs In this flow analog design and analog behavioral model are made in parallel and a specification check with behavioral model. Again due to use of pure digital simulation, analog effects and other intermediate wire are not come into picture. Using this model we can not predict the realistic behavior of AMS design at silicon, hence, failure occurs. Like that there are various challenges in Mixed signal SoC verification which should be overcome. In our approach we have proposed a modeling strategy for enhanced and accurate full chip verification using AMS designs. New approach is nothing combination of mathematical formulation, signal processing algorithms and real value concept. To bridges the gap between actual model and the cad view model, RVM is needed for AMS IPs. Fig. 2.2 shows the flow to model AMS design in better way which enhances

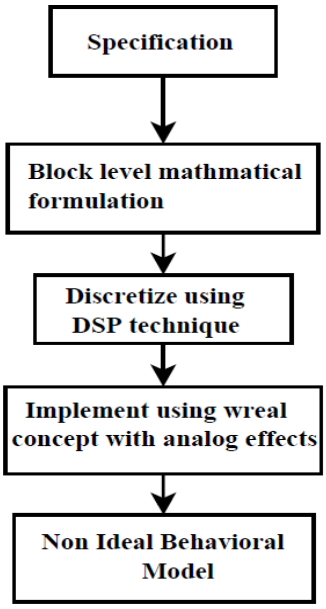

Figure 2.2: Proposed flow of Real Value Modeling

the verification in broader prospective. In our approach any AMS circuit can be model with help of mathematical formulation. Any AMS block can be represented in the form of linear equation and transfer function. But the equation are in time domain which could be discretized by applying signal processing technique. This digitalizes form can be arranged as difference equation which are implemented using standard RTL coding technique which facilitates the real floating point variable.

This type of modeling will change the overall flow of AMS verification which is shown in fig. 2.3. Now with the help of this methodology real equivalence checking can be done between cad view and post layout netlist which reduces the gap. This change in flow is only to ensure about effective verification for AMS design which is better than previous methodology. This model capture the realistic behavior of any AMS design, hence verification can be done at pre design stage at a faster speed.

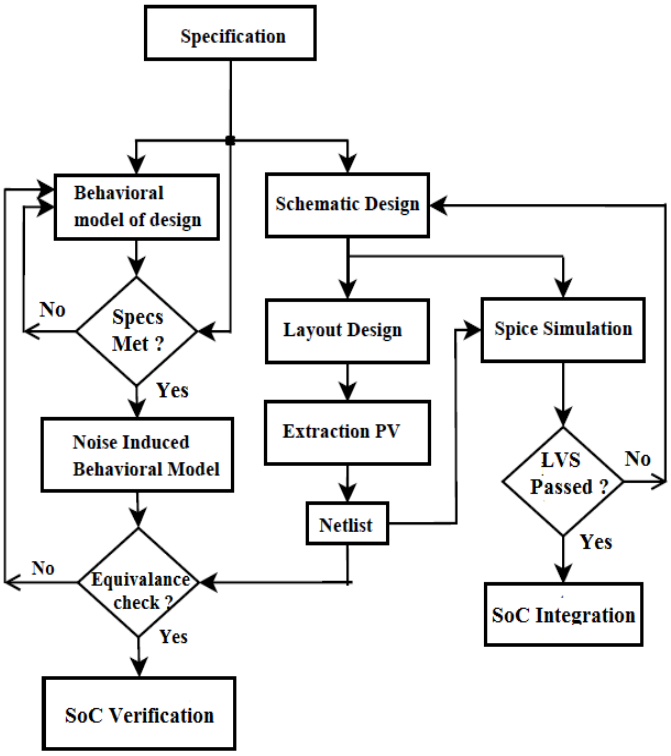

Figure 2.3: AMS design and verification flow due to RVM

Chapter 3

Real Value Modeling

3.1 Introduction

Now a days, with the increasing functionality at system level, the complexity as well as the number of Intellectual Properties(IPs) involved in a SOC is increasing. This leads to unavoidable interaction between Analog and digital IPs, creating an Analog and Mixed Signal(AMS) environment. Verification of such AMS environment involves a lot of simulations. Simulation for digital parts takes less time since they do not require high efficiency thus a large number of test patterns can be tested on it. While simulating analog part is time consuming process since it involves solving very complex differential equations for convergence. This simulation time adds up to total verification time at the SOC level making verification a hindrance. Hence, we need some methodology where we can maintain accuracy of analog simulation while achieve speed of digital one. We want to replace analog portions of SOC with their functionally equivalent digital models so that we do not require time taking analog engines for simulation. This indeed is achieved through modelling of analog blocks involved in AMS flow. The purpose of modelling is to provide simulation speedup where accuracy and speed trade-off must be taken into account. The classical approach to meet verification requirements was to use Verilog or have spice mixed simulation. Real value modelling is an add-on to these approaches. Real value modelling uses floating point numbers to represent data, like analog simulators while time is discrete like digital simulators.

3.2 Real value simulation with Accuracy and Effort

There are different abstraction levels generally used in AMS flow. The choice of abstraction level is generally dependent upon the specific application. These levels are:

1. Extracted Post Layout Spice (PLS)
2. SPICE level simulation
3. Fast SPICE simulations
4. Analog Behavioural model
5. Real Value modelling
6. Pure Digital modelling

To have faster time to market there was a need to have faster analog simulator. This led to have what today is commonly known as, fast SPICE. However the speed up ratio achieved with this was up to a factor of 1-2 which is not sufficient for verification. The simulation speed-up factor needed for AMS flow was in order of 4-5 which is achieved through Real value modelling and pure digital modelling.

Fig 1.1 shows the graph for different abstraction levels between their accuracy and performance. SPICE level simulations are golden reference enabling highest level of accuracy. While pure digital modelling is the fastest one but it has lowest accuracy. If we see the RVM, it provides simulation speed up but at the same time it suffers with lesser accuracy compared to AMS languages, which can be ignored from verification point. From literature servey and Fig 1.1 it can be derived 3.1 that, to generate model using Verilog or vhdl ams the effort requirement is very high and for spice is very low [[6]]. But the Real value approach gives midway from effort requirement as well as accuracy point. RVM is a mixed concept borrowed from both analog and digital domains. The values are continuous and floating-point (real) numbers like analog world while time is discrete i.e the real signals change values based on discrete events. In this approach, we apply the signal flow concept, so that the digital engine is able to solve the RVM system without

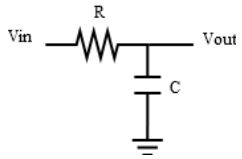

Figure 3.1: RC Filter

support of the analog solver. This guarantees a high simulation performance that is in the range of a normal digital simulation and orders of magnitudes higher than the analog simulation speed.

One thing about abstraction is that it is nothing but suppressing information which is redundant form the verification point of view. In such cases which details are hidden by the abstraction by modelling must be properly communicated till the verification engineer, otherwise that may lead to inefficient verification cycle. Also, an important aspect of modelling is that the behavioural models are, in general, differentiated based on modelling goal. A performance oriented models needs to precisely capture the critical behaviours which are needed to explore the design space. Functional model only capture the behaviours which are needed for top level verification and omits rest of the details. This type of modelling is very to investigate system characteristics before transistor level implementation, and to verify the connectivity of the entire system, thus, enabling a top-down design methodology. Real valued models that are simulated entirely within the event-driven (digital) simulator offer tremendous speed-up, thus, enables designers to do intensive verification in a short period of time. In addition, use of digital simulators enables more advanced methodologies to be put to use, such as dynamic random test benches, assertions, and functional coverage.

3.2.1 Example of real models in Verilog

Real variables i.e wreal models the continuous analog currents and voltages in discrete time stamp with accuracy comparable to spice simulators. Both wreal and spice evaluate at discrete time steps only difference is precision of time step. The largest differences description about the designs in different simulators. Discrete-time real models use difference equations, while analog representations use differential equations that require slow iterative solving. Discrete-time model for a voltage source driving an RC filter shown in Fig. 3.1 and the difference equation

and differential equation for the same RC filter is shown in Eqn. 3.2 and 3.4.

Difference Equation is

$$I_r = \frac{V_in - V_{out}}{R} \tag{3.1}$$

$$\Delta V_{out} = \frac{\Delta t I_r}{C} \tag{3.2}$$

Differential equation is

$$I_r = \frac{V_in - V_{out}}{R} \tag{3.3}$$

$$dV_{out} = \frac{dt I_r}{C} \tag{3.4}$$

Since Δt is a small fraction of the signal frequency, discrete-time difference equations could be same as periodically solved differential equations. Since the difference equations doesn't require iterative convergence and evaluation is done by the change of inputs, which allow execution speed to be faster than differential equations. It can be seen that Verilog is intermediate language to express discrete-time difference equations. Here dV/dt becomes $\Delta Vout/\Delta t$. Like that $\int Idt$ would become $I\Delta t$. Therefore, inductors and even parasitic coupling can be model with the help of difference equations and which gives the way for larger circuits to converge in finite time.

3.3 SoC Regression using RVM

Real Value Modeling (RVM) uses mixed signal flow of cadence and by using this, top-level verification can be done very efficiently with analog or mixed-signal designs using discretely simulated real number models. RNM gives the path to model digital equivalent of an analog block, hence enabling engineers to verify a full-chip SoC using only a digital simulator. This eliminates relatively slow analog simulation and convergence issues, allowing for nightly regression runs of the mixed signal SoC. RNM can also used with various advanced verification technologies, like assertion-based verification and metric-driven verification without interfacing with the analog engine. The MSV flow greatly enhances the top-level verification performance of the overall verification process.

3.4 Benefits and Limitations

3.4.1 Benefits

Improves Time To Market (TTM)

- Mixed-signal regression runs ensure the chip meets the specification with stipulated amount of time.
- Ensures product quality by accurate verification.

Reduces re-spins

- Finds and fixes errors much earlier in the design cycle by performing full-chip functional verification
- Performs SoC top-level verification

Boosts productivity

- Eliminates convergence issues with digital-speed performance.

3.4.2 Limitations

AMS modelling approaches are available since the starting of digital hardware description language, but the requirements were different from what we have today. Today, we have AMS simulators which can simulate analog parts of the design with desired accuracy. With respect to AMS languages, RVM modelling is used to significantly accelerate the simulation speed while trading off with accuracy. Models are described in different languages in different ways. In this modelling the design equations are solved for explicit solution. They actually formulate the behaviour of system in terms of differential algebraic equations (DAE). A conservation system is then created according to the data flow. Hence, RV modelling allows only data flow descriptions, where the data flow might be represented by a physical voltage or current. When the DAE is solved by an analog simulator, it contains time derivatives and to find out the initial condition, various numerical solution techniques are available. But for RV modelling only a single

approach which is forward Euler is available. This makes the job of model designers difficult since they now are only restricted for forward Eulers, and also maintain time steps by themselves.

Finally as concluding remarks, we have listed different areas where RV modelling can be successfully used.

- Data flow models that do not require a conservative system.
- Conservative models for which modelling of only current or voltage is sufficient.
- Analog level zero models where only propagation of data, power and control signals is needed without any implementation details.

Chapter 4

Proposed Real Value Approach for Phase Locked Loop

4.1 Background of PLL

This chapter explains about the fundamentals of the PLL, its operations, performances, building blocks, basic characteristics, and modelling strategy. At the beginning of the chapter, some fundamental on the PLL and its applications are introduced, and key characteristics of the PLL are described. Next, modelling of the main building blocks of the PLL are presented: VCO, loop filter, frequency divider, and especially Phase Frequency Detector (PFD). At last, noise sources and types with its modelling phenomenon is explained.

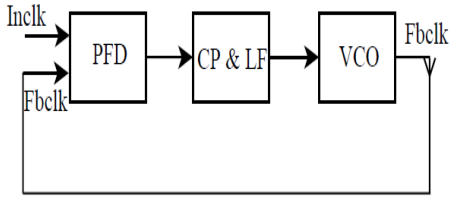

Figure 4.1: Basic block diagram of PLL

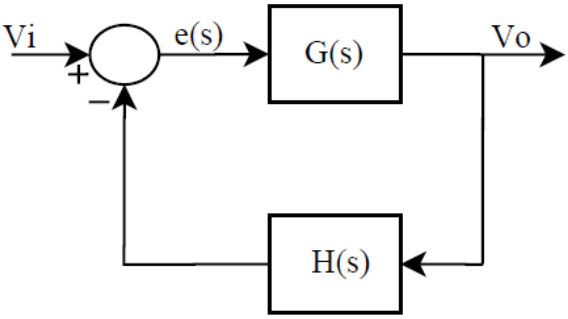

Figure 4.2: Standard negative fedback control system model

4.1.1 Fundamental of Phase locked loop

A phase-locked loop is a feedback system uses a voltage controlled oscillator (VCO) and a phase comparator which gives oscillations maintaining a constant phase angle relative to reference signal. The objective of PLL is to generate stable output high frequency signals from a fixed low-frequency signal. Another feature of PLLs is the filtering property, particularly with respect to the noise where its behavior recalls a very narrow low-pass arrangement that is not to be realized by other means and works as a feedback system shown in Figure 2.1. A PLL can be used in applications such as clock recovery and frequency synthesis [13]. Clock recovery is the process of extracting clock information from a given stream of data. Frequency synthesis is employed when particular clock frequencies need to be generated. The required frequencies are generally a multiple of some reliable reference frequency. Figure 4.1 shows the basic PLL model. The PLL can be evaluated as a negative feedback system using Laplace Transform theory with a forward gain as G(s), and a feedback magnitude of H(s), as shown in Figure 4.2.
The building blocks of the PLL are the Phase frequency detector (PFD) and a charge pump (CP), Loop Filter(LF), Voltage controlled Oscillator (VCO), and a Feedback Loop Divider. The effect of negative feedback the error output e(s), to reach zero where the feedback divider output and the reference frequency input are in phase and frequency lock i.e $F_0 = NF_{REF}$.

Referring to Figure 4.1, a system for using a PLL to generate higher frequencies than the input, the VCO oscillates at an frequency of F_O and a portion of this signal is fed back to the PFD, through a feed back path. The PFD compares the two signals and gives a constant output when two inputs signal are equal in phase and frequency, then the loop is said to be in a locked

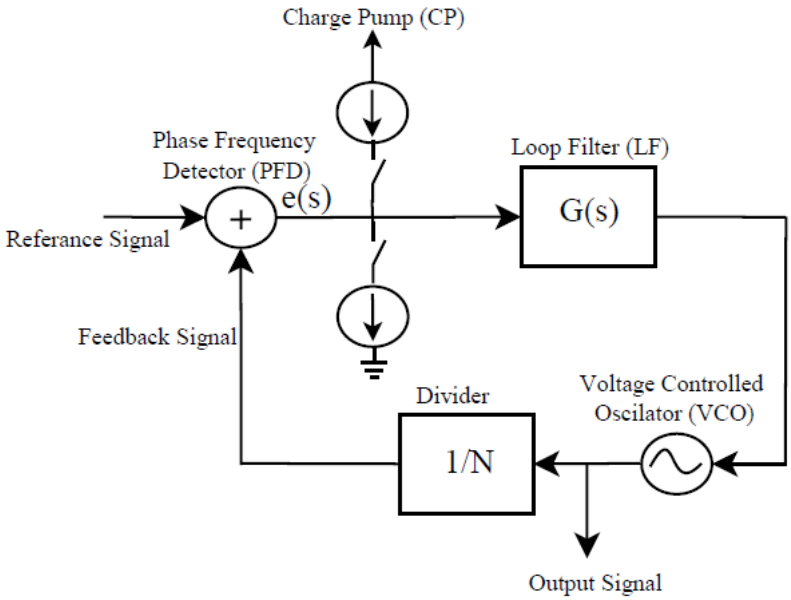

Figure 4.3: Block diagram of Integer PLL

condition .

4.1.2 Integer vs Fractional

An Integer-N PLL generates an output frequency at an integer multiple of its input frequency. A block diagram of an Integer-N PLL in Figure 4.3. The input to the PLL is the reference signal which is generally provided from a crystal oscillator. A Phase Frequency Detector (PFD) compares the rising edges of the Reference Signal and Feedback Signal to determine leading or lagging between the signals. Based on this comparison result the charge pump either sinks or sources current which is indirectly related to the phase difference of the reference and feedback signals. The Loop Filter which has two main responsibilities. Firstly it converts the charge pump current into a voltage used to control the Voltage Controlled Oscillator (VCO). Secondly the filter controls the loop dynamics of the PLL (bandwidth, settling time, etc.). The VCO outputs a frequency that is related to the input voltage of the VCO. The output of the PLL is then fed into a divider block which divides the frequency back down to the reference frequency. The feedback signal from the divider is then fed back into the PFD, completing the loop. This mechanism allows provides the output frequency which is an integer multiple of the reference [2]. One major problem with an Integer-N PLL is that the output frequency is limited to be an integer multiple of the reference frequency. Therefore in order to have a small resolution in the

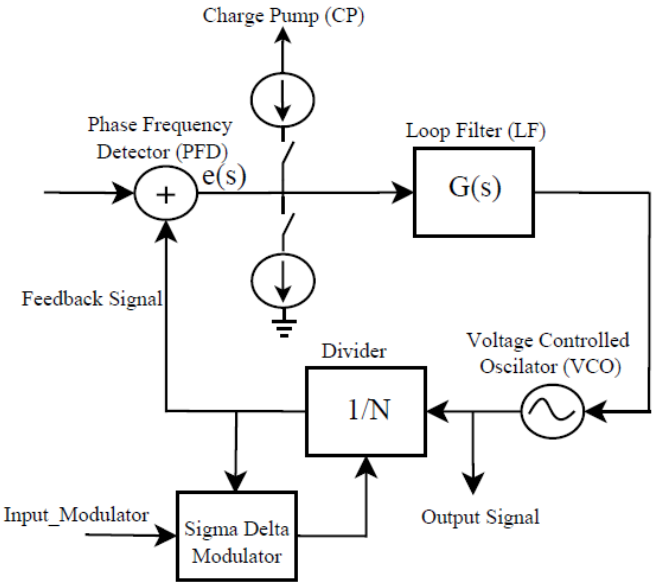

Figure 4.4: Block diagram of Fractional PLL

output frequency a small reference frequency is required. Small reference frequencies bring with it some problems such as decreased bandwidth, longer settling times, and increased noise.

A Frac-N PLL can be used to mitigate the shortcomings of an Integer-N PLL [5]. Figure 4.4 shows a Frac-N PLL block diagram. The Fractional PLL is having an additional feedback loop which includes the sigma-delta modulator (SDM). The SDM outputs a signal controls the divider bits of the loop divider. For example assume a PLL has a reference signal frequency of 25MHz. We wish our PLL output to have a frequency of 1.51GHz. This requires a divide ratio of 60 1/4. The way the PLL could achieve this divide ratio by dividing by 60 for three reference cycles, then 61 for one cycle. Over a certain time period we see that the average divide value is 60 1/4 as expected. The SDM is responsible to control the divider to divide by 60 or 61. The other way to simply divide by 60.25 and could be done with dithering between two integer divide ratios. The reason is the hardware implementation of the divider block allows only for integer division [24]. This means the output frequency of a Frac-N PLL must be an averaged result of many integer divide values.

One major benefit of a Frac-N PLL is the high resolution in output frequency. As discussed earlier, the output resolution of an integer-N PLL is limited to an integer multiple of the reference frequency.

4.2 Noise Analysis using Phase domain model of PLL

When signals are interpreted as phase the small-signal noise performance can be analyzed by linearizing the components and evaluating the transfer functions [13]. The phase-domain model shows in Fig 4.5 defines the forward gain of the loop and then the transfer function from the output are

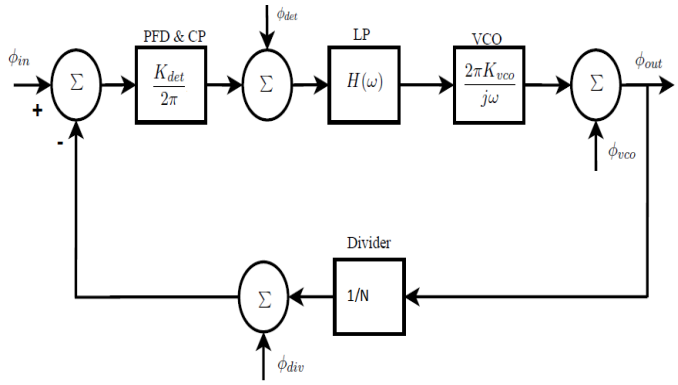

Figure 4.5: Phase-domain model

$$T_{fwd} = \frac{K_{det}}{2\pi} H(\omega) \frac{2\pi K_{vco}}{j\omega} = \frac{K_{det} K_{vco} H(\omega)}{j\omega} \tag{4.1}$$

where T_{fwd} is forward gain or open loop gain of the block diagram and T_{in} is closed loop transfer function.

$$T_{in} = \frac{\phi_{out}}{\phi_{in}} = \frac{T_{fwd}}{1 + T_{fwd}/N} = \frac{N T_{fwd}}{N + T_{fwd}} \tag{4.2}$$

$$T_{vco} = \frac{\phi_{out}}{\phi_{vco}} = \frac{N}{N + T_{fwd}} \tag{4.3}$$

And by inspection,

$$T_{div} = \frac{\phi_{out}}{\phi_{div}} = -T_{in} \tag{4.4}$$

and

$$T_{det} = \frac{\phi_{out}}{\phi_{det}} = \frac{2\pi T_{in}}{K_{det}} \tag{4.5}$$

On this last transfer function, we have simply referred ϕdet to the input by dividing through by the gain of the phase detector.

As $\omega \to \infty$, $T_{fwd} \to 0$ because of the VCO and low pass filter and so $T_{in}, T_{dev}, T_{det} \to 0$ and $T_{vco} \to 1$. At high frequencies, the noise is due to the VCO because the low pass LF blocks any feedback at high frequencies.

As $\omega \to 0$ $T_{fwd} \to \infty$ because of the $1/\text{j}\omega$ term from the VCO. So, at DC, $T_{in}, T_{dev} \to N$ and $T_{vco} \to 0$. At low frequencies, the noise of the PLL is given by the (Oscillator) OSC, PFD/CP, FD_M, FD_N and the noise from the VCO is diminished by the gain of the loop.

Consider the asymptotic behaviour of the loop and the VCO noise at low offset frequencies ($\omega \to 0$). Oscillator phase noise in the VCO results in the power spectral density S_{ϕ_vco} being proportional to $1/\omega^2$, or $S_{\phi_vco} \sim 1/\omega^2$ neglecting flicker noise. If the LF is chosen such that H(ω)$\sim$ 1, then $T_{fwd} \sim 1/\omega$, and noise contribution form the VCO to the output, $T_{vco}^2 S_{\phi_{vco}}$ is finite and nonzero. If the LF is chosen such that $H(\omega) \to 1/\omega$, as it typically is when a true charge pump is employed then $T_{fwd} \sim 1/\omega^2$ and the noise contribution to the output from the VCO goes to zero at low frequencies.

4.2.1 Jitter Analysis in PLL

Jitter can be defined as the randomness in the timing of events of a signal. To introduce jitter, the events of focus are the transitions at the output signal. Mathematically Jitter is the displacement in time with a stochastic process $j(t)$ for a noise-free signal v(t). The noisy signal becomes $v_n(t) = v(t) + j(t)$ [13]. To simplify , j can be assumed to be a zero-mean Gaussian process which might be non-stationary. Again we will be assumed to be T-periodic. There are two types of blocks in a PLL i.e driven blocks and autonomous blocks. Driven blocks, such as the PFD, CP, and Divider under goes phase modulation, or PM jitter. Autonomous blocks, such as the OSC and VCO, under goes frequency modulation, or FM jitter [12]. Table 4.1 previews the basic characteristics of PM and FM jitter

4.2.2 PM Jitter

PM jitter is a synchronous jitter from the driven systems of PLL like PFD, CP, and FDs. It cause a random fluctuation in the delay between the input and the output events. Therefore, PM jitter are generated by modulating the phase of the signal with a random process having zero mean and bounded variance. Thus, the frequency between output and input signal are exactly

Table 4.1: Jitter analysis

Jitter	Type	Circuits	J
PM	Synchronous	driven (PFD/CP, FD)	$\frac{\sqrt{var(n_v,t_c)}}{v(t_c)}$
FM	Accumulating	Autonomous (OSC, VCO)	$\sqrt{aT}$

same, only the phase of the output signal is fluctuating randomly with respect to that input. From above if we replace j by jPM, then $v_n(t) = v(t) + jPM(t)$ where $jPM = \eta T$ and ηT is T-cyclostationary [12]. If jPM is further restricted to be a white Gaussian T-cyclostationary process, then $v_n(t)$ exhibits simple PM jitter. The main characteristic of this jitter is that, each event is independent or uncorrelated. Driven circuits exhibit simple PM jitter if they are broadband and if the noise sources are Gaussian and small. The sources are considered small if the circuit responds linearly to the noise, though at the same time the circuit may be responding non-linearly to the periodic drive signal.

As will be shown in Section, if jPM is white and small, then from (29), the variation in vn is also white. jPM is small if $|Jpm| << T/k$ where T is the period of v and K is the highest significant harmonic of v [11].

4.2.3 FM Jitter

FM jitter is exhibited by autonomous systems which generates a stream of spontaneous output transitions, like oscillators, VCO in case of PLL. FM jitter is characterized by a randomness in the time since the last output transition, thus the uncertainty of when a transition occurs accumulates with every transition. Thus, compared with a jitter free signal, the frequency of a signal exhibiting FM jitter fluctuates randomly, and the phase drifts without bound in the form of a random walk.

$$j_{FM}(t) = \int_0^t \eta_\tau(t) d\tau \tag{4.6}$$

$$v_n(t) = v(t + j_{FM}(t)) \tag{4.7}$$

While ηT is cyclostationary and so has bounded variance, from (9) it is clear that the variance of jFM, and hence the phase difference between $v(t)$ and $v_n(t)$, can be unbounded [12]. If ηT is a white Gaussian T-cyclostationary process, then $v_n(t)$ exhibits simple FM jitter. In this case, the process $jFM(kT)$ that results from sampling jFM every T seconds is a discrete Wiener process and the phase difference between $v(kT)$ and $n(kT)$ is a random walk. The essential characteristic of simple FM jitter is that the incremental jitter that accumulates over each cycle is independent or uncorrelated. Autonomous circuits exhibit simple FM jitter if they are broadband and if the noise sources are Gaussian and small. The sources are considered small if the circuit responds linearly to the noise, though at the same time the circuit may be responding nonlinearly to the oscillation signal [11]. An autonomous circuit is considered broadband if there are no secondary resonant responses close in frequency to the primary resonance.

4.2.4 SOURCES OF JITTER

Thresholds

In systems where signals are continuous valued, an event is usually defined as a signal crossing a threshold in a particular direction. The threshold crossings of a noiseless periodic signal, $v(t)$, are precisely evenly spaced. However, when noise is added to the signal, $v_n(t) = v(t) + nv(t)$, each threshold crossing is displaced slightly. Thus, a threshold converts additive noise to PM jitter. The amount of displacement in time is determined by the amplitude of the noise signal,$nv(t)$, and the slew rate of the periodic signal, $dv(tc)/dt$, as the threshold is crossed. If the noise nv is stationary, then (31) where tc is the time of a threshold crossing. Generally nv is not stationary, but cyclostationary [12]. It is only important to know when the noisy periodic signal $v_n(t)$ crosses the threshold, so the statistics of nv are only significant at the time when $v_n(t)$ crosses the threshold,

$$var(j_{PM}(t_c)) = \frac{var(n_v, t_c)}{v(t_c)^2} \tag{4.8}$$

The jitter is computed

$$J = \sqrt{2var(j_{PM}(t_c))} \tag{4.9}$$

Oscillator Phase Noise

FM jitter is strongly related to oscillator phase noise. Both are different ways of describing the same underlying phenomenon. In other words, all free running oscillators exhibit a behavior that is generically referred to as oscillator phase noise. Any noise in an autonomous system will cause the phase to drift freely because there is no reference signal with which to lock. When the phase fluctuations are measured in terms of time deviation over a period, it is referred to as FM jitter. If it is measured in terms of noise signal amplitude as a function of frequency, it is referred to as oscillator phase noise [12]. Relating J and $S\phi$: In order to determine the period jitter J of $v_n(t)$ for a noisy oscillator, assume that it exhibits simple FM jitter so that η in (9) is a white Gaussian noise process (this excludes flicker noise) with a PSD of

$$S_n(f) = a \tag{4.10}$$

and an autocorrelation function of

$$R_n(t_1, t_2) = a\delta(t_1 - t_2) \tag{4.11}$$

where δ is a Dirac Delta function. Then

$$j_{FM}(t) = \int_0^t \eta_\tau(t) d\tau \tag{4.12}$$

is a Wiener process which has an autocorrelation function of

$$R_{jFM}(t_1, t_2) = a \min(t_1 - t_2) \tag{4.13}$$

The variance of jFM over one period T is

$$var(j_{FM}(t+T) - j_{FM}(t)) \tag{4.14}$$

$$= E[(j_{FM}(t+T) - j_{FM}(t))^2]$$
$$= E[j_{FM}(t+T)^2 - 2j_{FM}(t+T)j_{FM}(t) + j_{FM}(t)^2]$$
$$= E[j_{FM}(t+T)^2] - 2E[j_{FM}(t+T)j_{FM}(t)] + E[j_{FM}(t)^2]$$

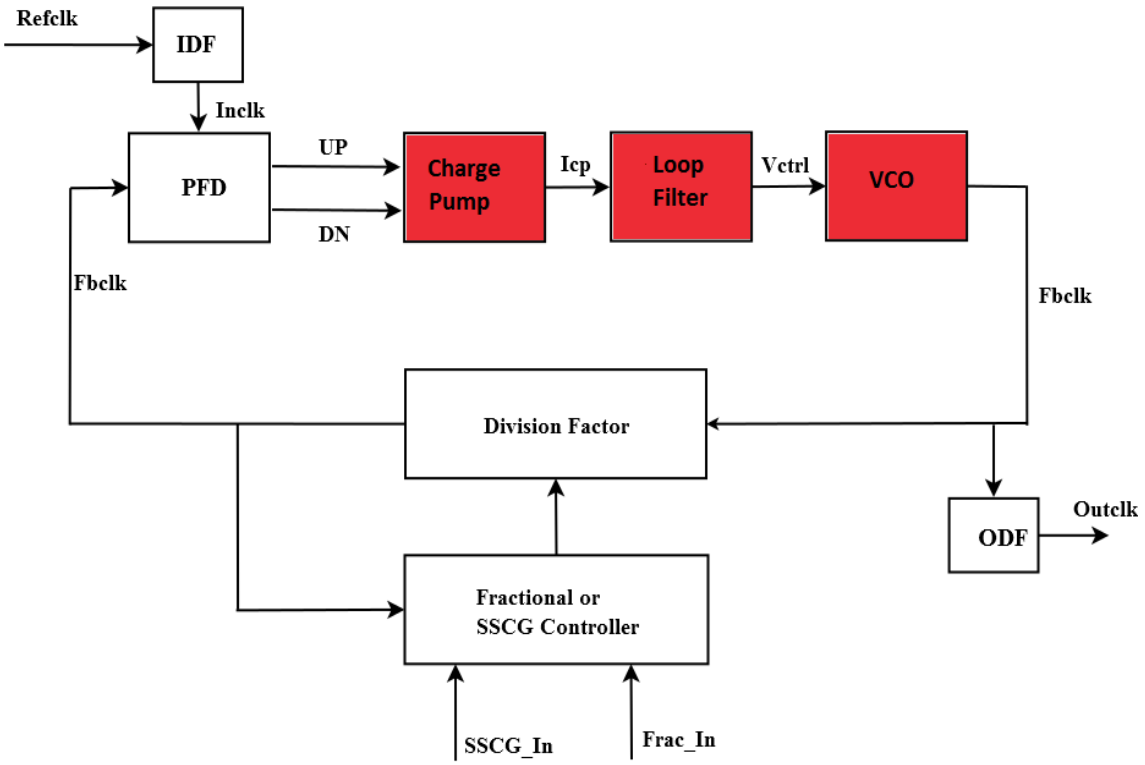

Figure 4.6: Block diagram of PLL

$$= R_{j_{FM}}(t+T, t+T) - 2R_{j_{FM}}(t+T, t) + R_{j_{FM}}(t, t)$$
$$= a(t+T) - 2at + at$$
$$= aT$$

Finally jitter is the standard deviation of the variation in the period and so,

$$J = \sqrt{aT} \tag{4.15}$$

4.3 Proposed Real Value Models

The block diagram of PLL frequency synthesizer with SSCG and fractional controller is shown in Fig. 4.6. The Red part shows the analog blocks and white part shows the digital blocks. Along with the conventional blocks, the SSCG controller controls the frequency spread at the output of PLL where as fractional division is controlled by fractional controller . There is selection pin which will select SSCG or Fractional or both. The design under modelling has

Table 4.2: Relation of UP and DN with charge pump

UP	DN	Charge Pump out
0	0	0
1	0	1
0	1	-1

Voltage Controlled Oscillator (VCO) which gives a low jitter output signal ranging from 2.5 GHz to 4.5 GHz and a programmable divider which produces wide range of division factor from 100 to 200. This allows VCO to operate at multiples of the reference clock, provided by a chip crystal oscillator running at 25MHz. The divider output signal is compared with input reference clock by a Phase Frequency Detector (PFD).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 [1].

4.3.1 Phase frequency Detector

PFD is the heart of PLL synthesizers, which processes discrete time phase difference between its two input signals and produces a proportional output signal. There are many types of phase detectors the design considerations include: noise, phase, capture range, and frequency capture range. The output signal of PFD is responsible for corrective measures in the output frequency. The phase detector compares the phase of the reference to the VCO phase. The schematic of a popular phase-frequency detector is shown in Figure 4.7(a) basically consisting of two D-type flip flops. One Q which is called UP triggers a positive current source and the other Q known as DN triggers the negative current source. Assuming taken are the D-type flip flop is positive-edge triggered, the possible states are shown in the logic table.

If the reference phase leads the VCO phase, then the phase detector produces an UP pulse. If the VCO phase is ahead of the reference, it produces a DOWN pulse. The digital output of a phase frequency detector is shown in Figure 4.7(b). The up and down pulses from the PFD go to charge pump, where they get converted into an analog signal. Consider if the system is out of lock, then the Inclk is much higher than the frequency of FBclk . The first rising edge on Inclk makes the output high ands is maintained till the first rising edge of FBclk to occur. In a practical system we can say that the UP is pulse trigger the VCO to increase its frequency with

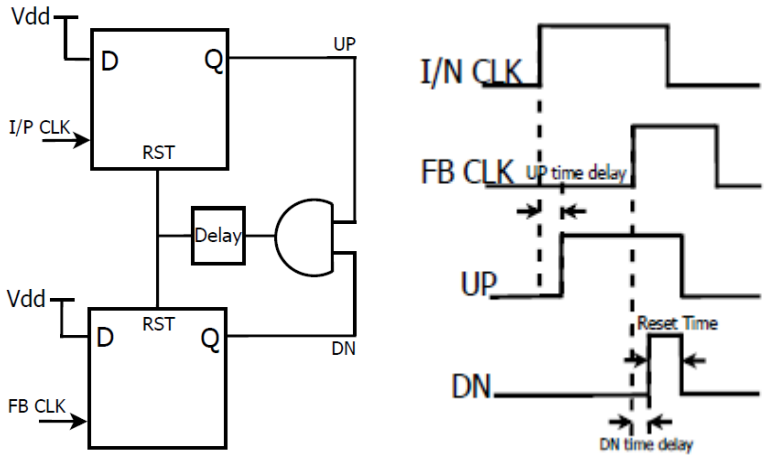

Figure 4.7: (a)Structure of PFD and (b) Input and Output responses of PFD

help up develop more control voltage at the loop filter output. This is exactly what is desired. If the frequency on INclk were much lower than on FBclk, the opposite effect would occur. This would have the effect of driving the VCO in the negative direction and again bring the frequency at FBclk much closer to that at INclk, to approach the locked condition [1].

The VCO would change until a significant phase error developed and initiated either positive or negative current pulses. Over a relatively long period of time, the cycling effect would charge and discharge the charge pump with respect to the sub-harmonic of the PFD. As this is a low frequency signal, it will not be attenuated by the loop filter and would result in very significant spurs in the VCO output spectrum, known as the "backlash" or "dead zone" effect [15]. Based on circuit level simulation, the PFD cannot detect small phase errors between input reference and feedback signal, thus loses its linearity. The SSCG modulator is very sensitive to such non-linearity and responds with increased phase noise and spurious emissions. Also from the first part of this chapter you know the PFD and Charge Pump will effected by the PM jitter. To reduce the influence of this small phase error and to successfully model the PM jitter, additional delay elements has been inserted between the AND gate and the reset signal of the flip flop as shown in fig 4.7(a). This leads to a non-ideality by reducing linear range of the PFD. So now, only phase errors between 0 and $2\pi - \Delta$, Where $\Delta = \frac{2\pi}{T_{ref}}$, can be corrected [15]. The charge pump shown in Fig. 4.8 will either source or sink current according to the up/down pulses provided by PFD. The mismatch of the up/down current from the charge pump is responsible

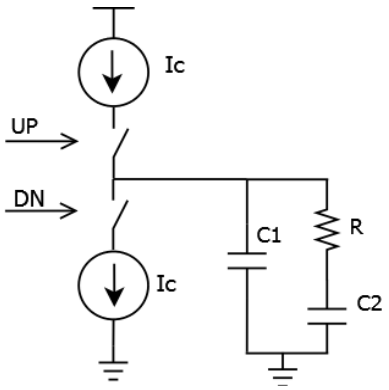

Figure 4.8: Loop filter with charge pump

for reference spurious at the output. Behaviours of such kind are modeled with new real value modelling and simulation approach. This helps to accomplish better modelling accuracy which in turn will enable us for more sophisticated verification.

4.3.2 Modeling of Loop Filter

For this real value event driven approach, instead of continuous time calculation, standard discrete time filtering method has been used. There 2 different type of approach for transformation of filter

1. Impulse Invariance Method
2. Bilinear Transformation

Impulse Invariance Method

Impulse invariance method for designing discrete-time infinite-impulse-response (IIR) filters from continuous-time filters. Due to sampling impulse response of the continuous-time system produces the impulse response of the discrete-time system. The frequency response of the discrete-time representation is the sum of shifted copies of the frequency response of the continuous-time system. There is condition for using this approach i.e continuous-time filter is approximately band-limited to a frequency less than the Nyquist frequency of the sampling, hence the frequency response of the discrete-time system will be approximately equal to it for frequencies

below the Nyquist frequency. In impulse invariance method the Laplace domain is converted in discontinuous time domain and again converted into discrete time domain using sampling [21].

Bilinear Transformation Method

The transformation is supposed to:

- approximate the frequency response of analog filter
- provides a stable digital filter by adding one zero to system.

Analog filter is stable when all the poles of transfer function are limited to the left half of s plane, whereas digital filter is stable when the poles are within the unit circle. This technique maps the left half of s plane with the area within the unit circle of z plane.
Here the loop filter is 2^{nd} order IIR filter as shown in Fig. 4.9 Specifically, we have used impulse invariance method to reconstruct the digital equivalent of continuous time loop filter.
The Laplace domain representation of this filter is

$$H(S) = \frac{V(S)}{I(S)} = \frac{RC_1S + 1}{RC_1C_2S^2 + (C_1 + C_2)S} \tag{4.16}$$

After inverse Laplace transformation, the impulse response in time domain is

$$h(t) = \frac{1}{(C_1 + C_2)}\{1 + \frac{C_1}{C_2}e^{-\{\frac{C_1+C_2}{RC_1C_2}t\}}\} \tag{4.17}$$

Impulse Response in discrete domain

$$h(n) = \frac{1}{(C_1 + C_2)}\{1 + \frac{C_1}{C_2}e^{-\{\frac{C_1+C_2}{RC_1C_2}nT\}}\} \tag{4.18}$$

Equation 4.18 is in continuous time domain which by Z transformation is converted for equivalent discrete time transfer function of the loop filter

$$H(Z) = \frac{V(Z)}{I(Z)} = \frac{1}{\beta}\{\frac{1 + \alpha - z^{-1}(e^{-\frac{\beta}{\alpha R}t} + \alpha)}{1 - Z^{-1}(1 + e^{-\frac{\beta}{\alpha R}t} + e^{-\frac{\beta}{\alpha R}t}Z^{-2})}\} \tag{4.19}$$

Equation (3) is converted to get impulse response in discrete time domain

$$v(n) + b_0 v(n-1) + b_1(n-2) = a_0 i(n) + a_1(n-1) \tag{4.20}$$

With the auxiliary variables a_0, a_1, b_0, b_1 an equivalent filter using a impulse invariance with difference equation method can be realized as

$$v(n) = a_0 i(n) + a_1(n-1) - b_0 v(n-1) - b_1(n-2) \tag{4.21}$$

Choice of sampling frequency is the most crucial element to model discretized equivalent of loop filter because smoothness of control voltage of VCO and overall simulation time for the model is governed by sampling frequency. Large sampling time will distort the VCO output while on the other hand small sampling time will detriment the benefits of event-driven approach [10]. As compared to the AMS modelling approach, by real value modelling the simulation time for discrete filter is decreased upto 30% [3].

module Loop Filter (I_n, $V_{control}$);

input I_n;

output $V_{control}$;

wreal I_n, $V_{control}$;

integer $seed$;

parameter $Thermal\_Noise$;

parameter $R, C_1, C_2, a_0, b_0, a_1, b_1$;

real $V(n), V(n-1), V(n-2), I(n), I(n-1)$;

initial begin

initializing the parameters..

end

assign $V_{control} = V(n)$;

always @($samplingclk$) begin

Th_Noise =$dist_normal($seed$,0,Noise_{voltage})$;

$I(n-1)$=$I(n)$;

$I(n)$=I_n;

$V(n-2)$=$V(n-1)$;

$V(n-1)$=$V(n)$ + Th_Noise;

$V(n)$=$a_0 i(n) + a_1 i(n-1) - b_0 v(n-1) - b_1 v(n-2)$;

end

List1: Part of loop filter source code

Higher order filters can also be implemented using this method or by cascading of 1st and 2nd order filters. A part of the sampled filter model is shown in List 1: The filter model uses dist_normal() function of Verilog to introduce thermal noise effect of impedance that also introduces equivalent timing jitter at the output. Comparative equivalence checking of the filter characteristic of sampled loop filter and its spice netlist is presented in result section. The modeled filter accurately tracks the frequency domain specification with respect to its spice counterpart which is sufficient for verification purpose. Also, the simulation time for 100μs transient analysis has been reduced by a factor of six.

4.3.3 Voltage Controlled Oscillator

Previously proposed VCO modelling approaches use phase variable method [**?**] and curve fitting method [16]. In phase variable approach the output of VCO is monotonically increasing. This makes detection of small phase deviations due to phase noise difficult. In the curve fitting approach extracted modelling coefficients must be large enough to maintain modelling accuracy. This increases the simulation time, depreciate the modelling purpose. The linear approach for VCO modelling is proposed in [14] which can predict more than 90% of real VCO characteristics. Therefore, in our approach, we have adopted linear VCO model to avoid limitations of previous approaches. In this work the critical concerns of VCO circuits are the gain (KVCO), the range of input control voltage and the output frequency. Frequency output instead of phase, simplifies the VCO model to linear relation present in [14] and provides a limited quantity at the VCO output which results in a more straightforward analysis of the results.

The Real value modelling of analog VCO, shown in List.2 involves accumulating jitter implementation in the wreal domain, respecting the influence of flicker (1/f) noise. It first converts the input voltage signal to the output frequency rather than the desired oscillation period. Then the timing jitter is computed and added to the period, which leads to the noisy VCO output signal. List 2 shows the part of the accumulating jitter implementation in the wreal VCO under noisy environment. Our target is to provide a VCO model with high simulation efficiency, which also maps the given specification from circuit level accurately enough in order to fulfil the aforementioned verification requirements.

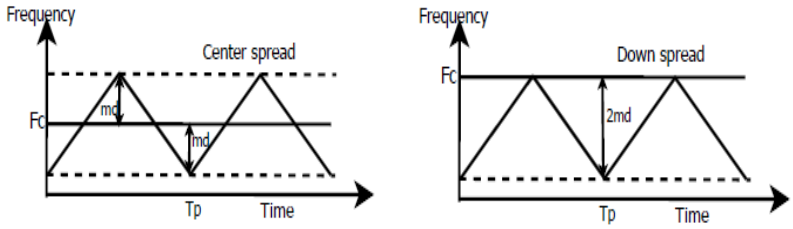

Figure 4.9: a. b. waveform from PFD

```
module vco (V_control, FB_clk);
Input V_control;
Output FB_clk;
Wreal V_control;
//declaration of parameters
Initial begin
//initializing parameters
end
always begin
J(t)=Jt * dist_normal(seed,0,1);
vco_freq = f_free+K_v*(V_control − V_t) +J(t);
vco_per = 0.5/Vco_freq;
??/// end
assign FB_CLK = vco_out;
end
```

List2: Part of VCO Source code

4.3.4 SSCG Controller

Spread spectrum clocking lowers clock-generated EMI from both the fundamental frequency and subsequent harmonics, thereby reducing the total system EMI. Due to its pure digital implementation on circuit level, the model of SSCG controller can be realized using event driven approach. There are two different types of spread spectrum clocking: down spread and center spread [22]. Down spread spectrum clocking modulates below the nominal clock frequency

while center spread spectrum clocking modulates evenly above and below the nominal clock frequency [22]. The type of spread spectrum clocking used depends on the specifications of the application. Some destination like chipsets, CPUs, etc., in which maximum clock frequency specification cannot be violated, uses down spread spectrum clocking [20].

4.3.5 Fractional Controller

The simulation of PLL with fractional controller is difficult because synthesizer are having time constants with different magnitude, which requires long simulation times with small time steps to capture all its effects [24]. The fundamental operation of these modulators relies on the fact that the spectrum of the quantization noise is shaped such that a small amount of noise power remains within the useful signal band, while the rest is pushed to higher frequencies [5]. The same principle can be exploited in fractional-N frequency synthesis applications by pushing the phase error towards higher frequencies so that the phase noise in the vicinity of the desired carrier frequency is small [9]. The high-frequency phase error is subsequently suppressed by the loop filter, which has a low-pass characteristic. This is a full digital implementation so, we have not used any modeling strategy like RVM concept. Due to its digital nature it is easy to integrate with RVM models and gives accurate behaviours. The main work of this block is to control divider ratio for a fractional output frequency.

4.4 MODELING DUE TO CORNER VARIATIONS

The motivation behind introducing cross corner analysis in behavioral model is to achieve full functional coverage with respect to spice. Using spice with extracted resistance and capacitance of PLL full functional SoC verification with all corners is not possible within stipulated time. Also, due to handshaking problems between analog and digital domains, we may not verify all test cases which can result in post silicon failure. To tackle these problems, non-idealities due to corner variations must be included in behavioural model, that helps in finding and fixing the issues generated due to the clk from PLL.

For the above mentioned purpose, we will need to introduce methodology to use the extraction

results from post-layout simulations which determine the parameter values of our behavioral model due to the change of corners [14]. In the proposed Real value models shown in Section III, important internal parameters like starting voltage of VCO, VCO gain etc. are extracted from voltage-domain measurement. Therefore, we make an assumption that the distribution of those internal parameters in our real value models will also change according to cross corner. To predict the corner variations effect on internal parameter accurately, we do need several post layout simulations. In our approach we have five simulation to extract those distribution of parameters for the real value behavioral model when the corner are slow-slow, fast-fast, typical-typical, slow-fast and fast-slow. These extracted values are again annotated to the single model using $ifdef$ statement, as all the real values are parameterize which can be change from the top level during simulation.

With this approach our models have shown accurate responses with the parameter variation, hence it meets the aim of closely track the SPICE behavior of PLL. In next section, we will demonstrate our idea by using a Gaussian distribution to produce various values of parameter changes with respect to the corner. A random variation generator is used to produce some unspecified parameter ranges of respective corner.

Chapter 5

Results

In order to demonstrate the accuracy of SSCG PLL which is modeled using RVM, we have used post layout SPICE of SSCG PLL designed in 40nm CMOS process. For this PLL, value of input reference frequency is 25MHz and the output frequency (F_{out}) is 4500MHz and divider ratio is taken to be 2.

5.1 Simulation Set-up

The accuracy of modelled PLL and its other parameters are calculated upon a simulation set-up. The abstract level of this simulation set-up is shown in the fig 5.1. Same stimuli is used to drive SPICE as well as behavioural model in Mentor QuestaADMS simulation tool. Thus obtained results are compared in EZwave Waveform viewer. All the results discussed in following sections are verified in the same set-up [18].

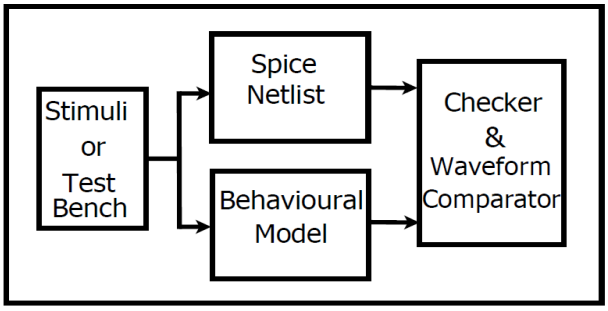

Figure 5.1: Comparison VCO control Voltage between Ideal Real Value Model and Spice Netlist

5.2 Verification Results

For the sake of comparing the accuracy of modeling approach, we can compare input control voltage of VCO i.e. V_{ctrl}. Control voltage is of interest because this only will define the output frequency of VCO, the overall system output.

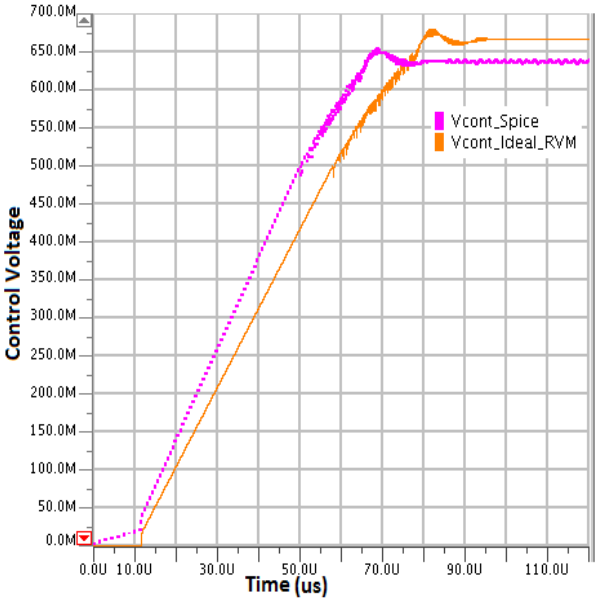

Figure 5.2: Comparison of VCO control voltage between Real Value Model and Spice Netlist

Fig 5.2 explains the difference of V_{ctrl} for SPICE as well as for RVM. As can be seen the difference between two is very high and not acceptable. This is the case because we have yet not modelled the noise behaviour of involved analog blocks. To mitigate this, next we have introduced actual analog effects which are present in actual SPICE. Previous chapters have discussed about these parameters like delay of flops, thermal noise, FM and PM jitter and its impact on output. With the help extracted value of mentioned parameters, and other factors like threshold variation due to corner analysis, from post layout netlist the loop filter will able to produce real noisy control voltage. This helps to have better accuracy which is evident from fig 5.3. Obtained realistic behavior is obvious since non-ideal effects have been successfully considered.

5.2.1 Result of Fractional

Fig 5.3 explains the difference of V_{ctrl} for SPICE as well as for RVM with all analog effects. This control voltage is input to the VCO and responsible for frequency generation at the output shown in Fig. 5.4. The output timing jitter defines the fluctuations of the zero crossings

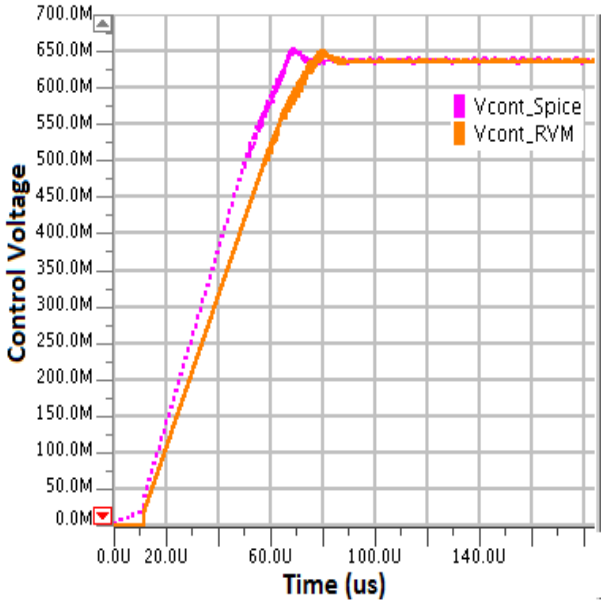

Figure 5.3: Comparison of VCO control voltage between Proposed Real Value Model and Spice Netlist

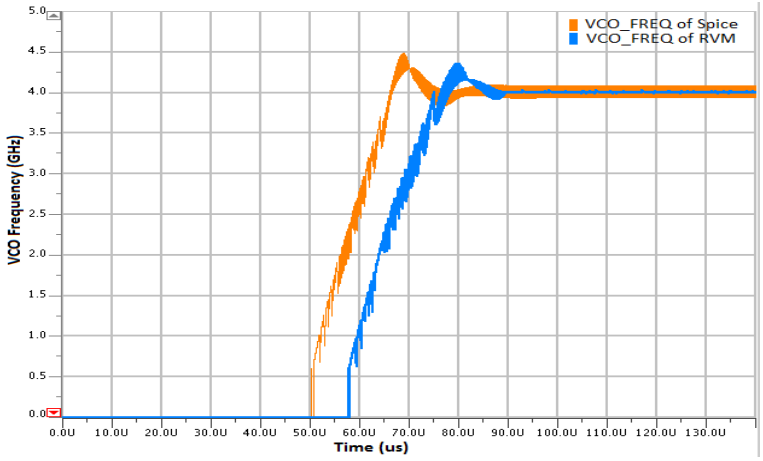

Figure 5.4: VCO Frequency vs Time Plot for fractional PLL

of the VCO signal over the ideal values, whereas phase noise is the corresponding frequency-domain equivalent. To demonstrate this, FFT of the output signal is taken after it has been reached to the steady state. The plotted results show frequency content available at the output

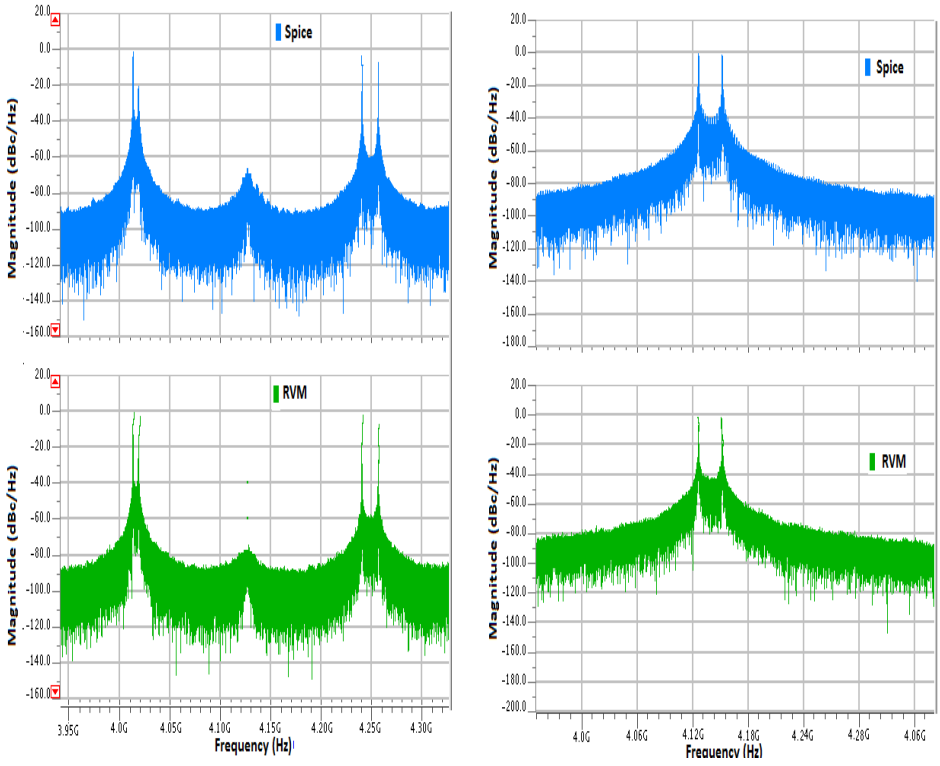

Figure 5.5: (a) Spectral purity of Fractional PLL and (b) spectral purity of average Frequency in Fractional PLL

which includes principal as well as harmonics. The principal spectrum has the highest level of magnitude while the harmonics are attenuated enough to be discarded. Fig. 5.4 and 5.5 show the simulated spectral purity of fractional. From the graphs it is evident that the principal output frequency is 4.15GHz. It is clearly shown from Fig 5.5 that peak occurs at 4.0 and 4.25GHz. The output wave consists of 4.0GHz instantaneous frequency and 4.25 instantaneous frequency. It can be observed from Fig 5.6, for a particular number of cycles, the frequency is all most average of the two instantaneous frequency. 4.0 and 4.25.

5.2.2 Result of SSCG PLL

Fig 5.3 explains the difference of V_{ctrl} for SPICE as well as for RVM with all analog effects for SSCG PLL and Fig shows the VCO frequency vs time comparison.

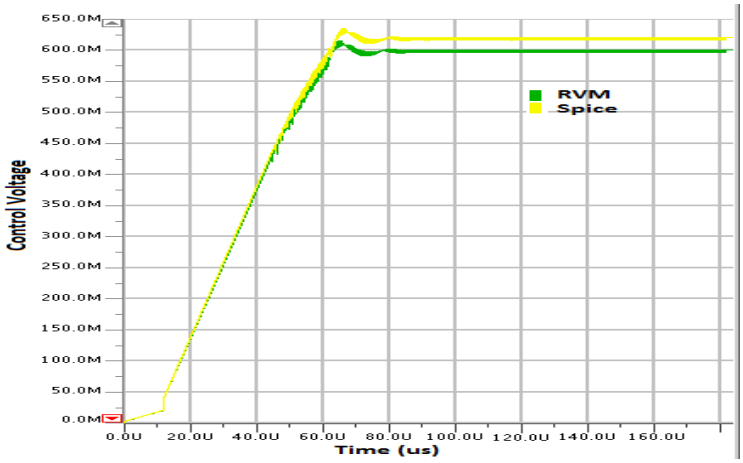

Figure 5.6: VCO control voltage vs Time Plot for SSCG PLL

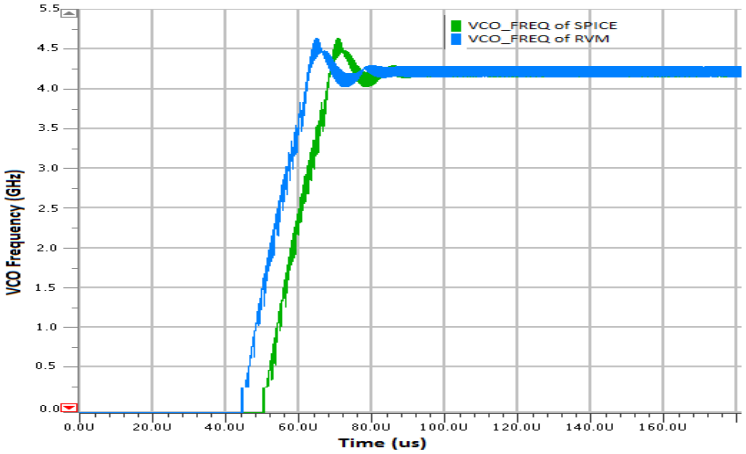

Figure 5.7: VCO Frequency vs Time Plot for SSCG PLL

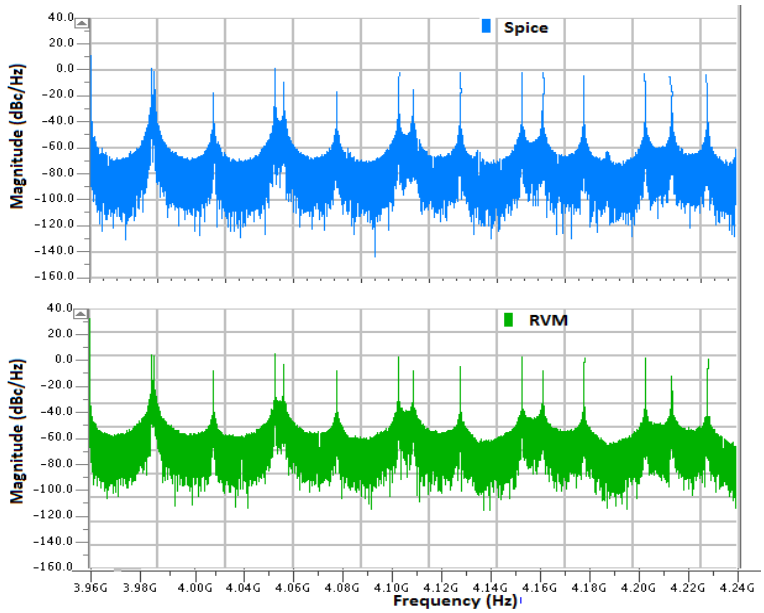

Figure 5.8: spctral purity of SSCG PLL

5.3 Cross Corner Analysis

In the SPICE, the behaviour of transistor is modeled for process, temperature and voltage variations. This is the realistic behaviour of silicon, specially in deep-submicron technologies. Hence this becomes a necessity for accurate modeling to consider these variations. Fig 5.9, 5.10, 5.11, 5.12 shows the cross corner variations of proposed modelling with SPICE for SSCG PLL. The results are taken on worst case corners where worst corners are defined as the ones which has maximum variations w.r.t. SPICE.

5.3.1 Fractional PLL

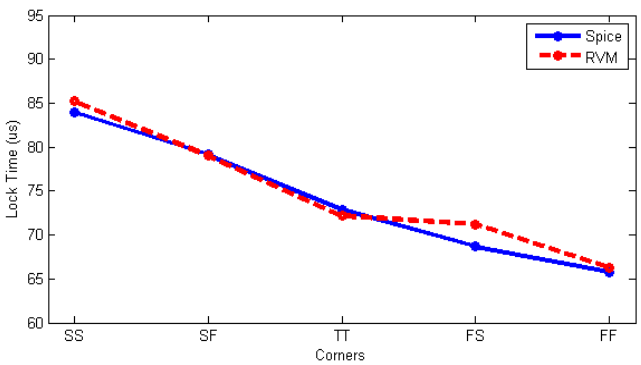

Figure 5.9: Cross corner analysis of Lock Time between Spice vs RVM for Fractional PLL

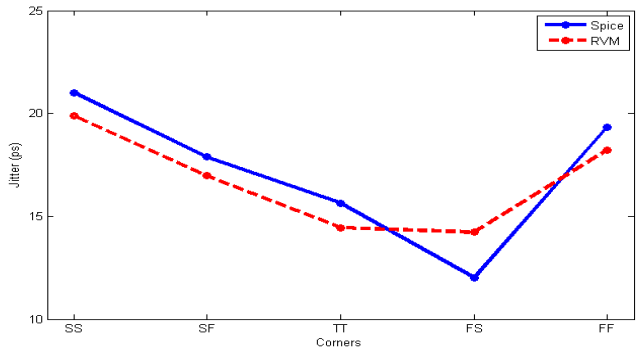

Figure 5.10: Cross corner analysis of Jitter between Spice and RVM For Fractional PLL

Table 5.1: Cross corner analysis between Spice vs RVM for different parameters for Fractional PLL

	SS		**SF**		**TT**		**FS**		**FF**	
	SPICE	RVM	SPICE	RVM	SPICE	RVM	SPICE	RVM	SPICE	RVM
Lock Time (us)	83.90	85.2	79.12	79.01	72.90	72.16	68.66	71.22	65.76	66.32
V_{lock} (v)	0.600	0.630	0.590	0.596	0.610	0.619	0.588	0.579	0.569	0.571
V_{max} (v)	0.620	0.648	0.600	0.610	0.629	0.630	0.611	0.598	0.586	0.589
Period Jitter (ps)	21.11	19.88	17.88	16.97	15.63	14.44	12.01	14.22	19.35	18.23

Fig 5.9 and 5.10 shows the variation of lock time and jitter through out the corner which will again give an efficient verification for PLL. The table shown at 5.1 gives the best case and worst case parameters due to corner variation and comparative analysis of different parameter like lock time, lock voltage, maximum overshoot of control voltage, period jitter.

5.3.2 SSCG PLL

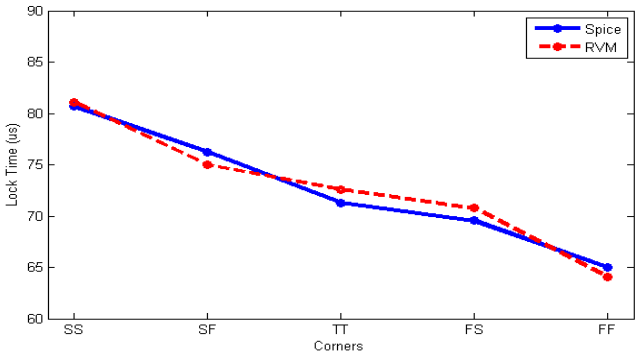

Figure 5.11: Cross corner analysis of Lock Time between Spice vs RVM for SSCG PLL

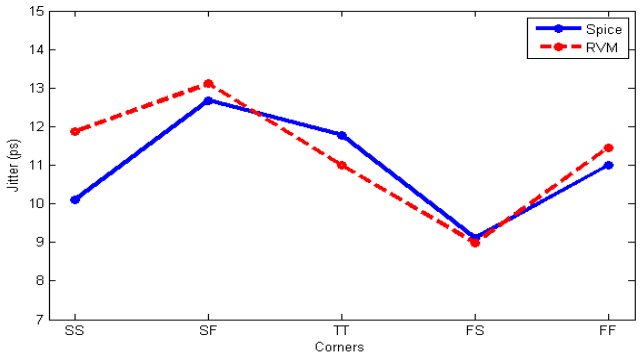

Figure 5.12: Cross corner analysis of Jitter between Spice and RVM for SSCG PLL

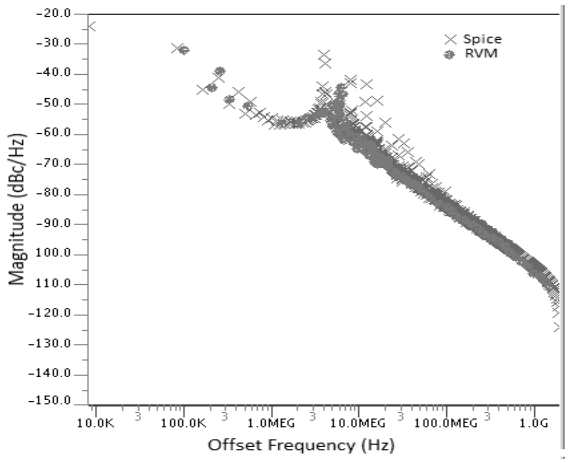

Figure 5.13: Phase noise comparison of PLL spice model and our proposed model for SSCG PLL

Finally, we have presented phase noise comparison of SPICE and proposed model for SSCG PLL in Fig 5.13. The summary of all the results is shown in table 5.2. Phase noise show the change of phase of output when multiplied to some offset frequency. Some what the phase noise of spice version is more than RVM but it follows the same pattern.

5.4 Subsystem and SoC Verification Result

The proposed approach of verification is used for verifying HDMI port and DSP processor. In both design clock is generated using Phase locked loop. For DSP processors except PLL all other digital blocks are written using HDL languages. Similarly for HDMI port PLL is the only AMS block. Table 5.3 shows the simulation time for verification of aforesaid blocks. With the results we can observe reduction in simulation time which 60-75% for tradition approach w.r.t.

Table 5.2: Cross corner analysis between Spice vs RVM for different parameters

	SS		**SF**		**TT**		**FS**		**FF**	
	SPICE	RVM	SPICE	RVM	SPICE	RVM	SPICE	RVM	SPICE	RVM
Lock Time (us)	80.68	81.1	76.24	74.99	71.27	72.59	69.55	70.79	64.99	64.11
V_{lock} (v)	0.720	0.700	0.570	0.590	0.638	0.635	0.598	0.600	0.550	0.571
V_{max} (v)	0.740	0.729	0.590	0.600	0.656	0.651	0.616	0.615	0.566	0.589
Period Jitter (ps)	10.11	11.88	12.68	13.11	11.78	10.99	9.11	8.99	10.99	11.44

Table 5.3: Comparison of Spice, Traditional and Proposed Approach

Product	Subsystem	SoC
Application	HDMI Port	DSP Processor
Specification	PLL, Controller, Transmeter, Receiver, Encoder	2 PLL, Communication Processor, LCD driver, Memories, Image Proceesor
Verification time Using Spice (weeks)	9-11	7-9
Verification time Using Traditional Approach (weeks)	3-5	2-4
Verification time Using RVM (weeks)	3.7-5.5	2.5-4.3

SPICE simulations. With the proposed approach we can see a slight i.e 8-12 % increase in the simulation time which is obvious since analog behaviours are included in the proposed models. The gain in the accuracy mitigates this negligible increse in simulation time. Also, table 5.4 shows the environment used for the simulation purpose.

Table 5.4: Simulation Environment

Operating Systems	Red Hat Enterprise Linux v5.9
processing cores	8
Processor Speed(GHz)	2.80
Simulation Tool	Questa ADMS

Chapter 6

Conclusion and Future Work

6.1 Conclusion

A Real Value Modelling based Phase Locked Loop (PLL) is has been developed as an example for Analog and Mixed signal verification to providing accurate functionality to on chip PLL and prevent failure due to PVT variations. The motivation for the RVM has been emerged to reduce the time and expense to verify this SoC and also to restrict after silicon failure due to lack of verification strategy. The RVM proto type have been designed successfully with comparison of its spice and other traditional behavioural model.

The model has been integrated into a digital SoC to show the reduction of verification time as well functionality check full SoC due to clk. The functionality of the IP has been verified by using different functionality check test cases. The proposed verification methodology particularly focuses on the intrinsic reflection in behavioral, which adds to the deviation of the behavioral model from the electrical model. Two model space mapping capacities are additionally characterized to guide the changes from behavioral to electrical area and vice versa. Moreover, the models provide observable speedup against the widely used ams models in the existing SoC designs. The implementation of the proposed approach into an available design flow can lead to a fast and reliable functional verification process.

6.2 Future Work

Exploration of real value modelling has been covered in this research while specific issue must be dealt according to application requirements. If required, features like voltage and temperature variation for all corner analysis can be examined, which will again predict the spice very accurately etc. Those are not used often for the verification point, but the addition of extra features will make the methodology more efficient and effective. Prototype for Phase locked loop using RVM has been successfully presented, which enable the idea to build RVM based model for Analog to Digital Converters (ADC), Voltage Sensor, Thermal Sensor, Voltage Regulators etc.

Bibliography

[1] Antao, B. A., El-Turky, F. M., and Leonowich, R. H. Behavioral modeling phase-locked loops for mixed-mode simulation. *Analog Integrated Circuits and Signal Processing 10*, 1-2 (1996), 45–65.

[2] Balivada, A., Hoskote, Y., Abraham, J., et al. Verification of transient response of linear analog circuits. In *VLSI Test Symposium, 1995. Proceedings., 13th IEEE* (1995), IEEE, pp. 42–47.

[3] Cottrell, R. Event-driven behavioural simulation of analogue transfer functions. In *Proceedings of the conference on European design automation* (1990), IEEE Computer Society Press, pp. 240–243.

[4] Dang, T., Donzé, A., and Maler, O. Verification of analog and mixed-signal circuits using hybrid system techniques. In *FMCAD* (2004), vol. 3312, Springer, pp. 21–36.

[5] Duarte, R., and Fernandes, J. R. A behavioural model for sigma delta fractional pll. In *Mixed Design of Integrated Circuits & Systems, 2009. MIXDES'09. MIXDES-16th International Conference* (2009), IEEE, pp. 392–395.

[6] Hartong, W., and Cranston, S. Real valued modeling for mixed signal simulation. *Cadence Design Systems, Jan* (2009).

[7] Hartong, W., Hedrich, L., and Barke, E. Model checking algorithms for analog verification. In *Proceedings of the 39th annual Design Automation Conference* (2002), ACM, pp. 542–547.

[8] Hedrich, L., and Barke, E. A formal approach to nonlinear analog circuit verification. In *Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design* (1995), IEEE Computer Society, pp. 123–127.

[9] Huang, S., Ma, H., and Wang, Z. Modeling and simulation to the design of $\sigma\delta$ fractional-n frequency synthesizer. In *Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE'07* (2007), IEEE, pp. 1–6.

[10] Joeres, S., Groh, H.-W., and Heinen, S. Event driven analog modeling of rf frontends. In *Behavioral Modeling and Simulation Workshop, 2007. BMAS 2007. IEEE International* (2007), IEEE, pp. 46–51.

[11] Kim, J. W., Lu, Y.-C., and Dutton, R. W. Modeling and simulation of jitter in phase-locked loops due to substrate noise. In *Behavioral Modeling and Simulation Workshop, 2005. BMAS 2005. Proceedings of the 2005 IEEE International* (2005), IEEE, pp. 25–30.

[12] Kundert, K. Predicting the phase noise and jitter of pll-based frequency synthesizers. *Available from www. designers-guide. com* (2003).

[13] Kundert, K. Modeling jitter in pll-based frequency synthesizers. *www. designers-guide. org* (2006).

[14] Kuo, C.-C., and Liu, C.-N. Accurate behavioral modeling approach for pll designs with supply noise effects. In *Behavioral Modeling and Simulation Workshop, 2005. BMAS 2005. Proceedings of the 2005 IEEE International* (2005), IEEE, pp. 48–53.

[15] Lee, K.-S., Park, B.-H., Lee, H.-i., and Yoh, M. J. Phase frequency detectors for fast frequency acquisition in zero-dead-zone cpplls for mobile communication systems. In *Solid-State Circuits Conference, 2003. ESSCIRC'03. Proceedings of the 29th European* (2003), IEEE, pp. 525–528.

[16] Mounir, A., Mostafa, A., and Fikry, M. Automatic behavioural model calibration for efficient pll system verification. In *Proceedings of the conference on Design, Automation and Test in Europe: Designers' Forum-Volume 2* (2003), IEEE Computer Society, p. 20280.

[17] Rutenbar, R. A., Gielen, G., and Antao, B. *Computer-aided design of analog integrated circuits and systems*. IEEE Press Piscataway, NJ, USA, 2002.

[18] Steinhorst, S., and Hedrich, L. Advanced methods for equivalence checking of analog circuits with strong nonlinearities. *Formal Methods in System Design 36*, 2 (2010), 131–147.

[19] Tiwary, S. K., Gupta, A., Phillips, J. R., Pinello, C., and Zlatanovici, R. First steps towards sat-based formal analog verification. In *Computer-Aided Design-Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on* (2009), IEEE, pp. 1–8.

[20] Wang, B., and Ngoya, E. Integer-n plls verification methodology: Large signal steady state and noise analysis. *Circuits and Systems I: Regular Papers, IEEE Transactions on 59*, 11 (2012), 2738–2748.

[21] Wang, Y., Van-Meersbergen, C., Groh, H.-W., and Heinen, S. Event driven analog modeling for the verification of pll frequency synthesizers. In *Behavioral Modeling and Simulation Workshop, 2009. BMAS 2009. IEEE* (2009), IEEE, pp. 25–30.

[22] Yang, C.-Y., Chang, C.-H., and Wong, W.-G. A pll-based spread-spectrum clock generator with a ditherless fractional topology. *Circuits and Systems I: Regular Papers, IEEE Transactions on 56*, 1 (2009), 51–59.

[23] Zaki, M. H., Tahar, S., and Bois, G. Formal verification of analog and mixed signal designs: A survey. *Microelectronics Journal 39*, 12 (2008), 1395–1404.

[24] Zhipeng, Y., Wenbin, C., and Kennedy, M. P. Modeling and simulation of $\delta\sigma$ fractional-n pll frequency synthesizer in verilog-ams. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 90*, 10 (2007), 2141–2147.