STANDARD CELL ALL-DIGITAL PHASE LOCKED LOOP DESIGN, ANALYSIS AND HIGH-LEVEL SYNTHESIS

by

Yalçın Balcıoğlu

M.S., Electrical and Computer Engineering, Ohio State University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2016

ACKNOWLEDGEMENTS

Although this is an individual study, it could not have been completed without the support of some people to whom I want to thank for helping and supporting me during this period.

Firstly, I am gratefully indebted to my advisor, Prof. Günhan Dündar. Mr. Dündar have never hesitated to share his technical knowledge and more importantly his time. He has allocated time for my study despite the busyness in his life. I would also like to acknowledge Mr. Emre Pusane for his support on any theoretical questions that I had.

Of course the biggest gratitude should go to my lovely family who have and will always have a special role in my life. I don't know how to thank my parents Taner and Ayten and my wife Bahar. They have always cared and loved me. They have provided the best possible life for me with their all limited resources. They have always trusted me and they haven't expected anything for this. Their love, support and tolerance have always encouraged me to study tirelessly.

Finally, there are also other people who have helped and supported me and I am sorry that I could not type their names here, but I present great gratitude to all of them.

ABSTRACT

STANDARD CELL ALL-DIGITAL PHASE LOCKED LOOP DESIGN, ANALYSIS AND HIGH-LEVEL SYNTHESIS

This thesis presents a new quantization noise suppression method for a time-to-digital Converter (TDC) and proposes an all-digital phase locked loop (ADPLL) architecture using only standard cell logic gates. Using new multiple input multiple output (MIMO) quantization noise suppression method provides an order of $\sqrt{2}$ improvement in TDC resolution with N parallel TDC channels. Suppressed noise in TDC allows the ADPLL achieve superior jitter performance in both theoretical predictions and simulation results. In order to allow fast portability between process nodes, ease of modification, and provide flexibility, ADPLL architecture is designed completely in register transfer level (RTL) intensive Verilog code and the implementation is synthesized in order to obtain final microelectronic design schematics. In comparison to similar work in literature, designed ADPLL achieves superior long term jitter with comparable area and power consumption.

Furthermore, we present a new tool called CellPLL that provides a complete design, analysis, and high-level synthesis (HLS) flow for all-digital phase locked loops (ADPLL). CellPLL uses a methodology for direct design of transfer functions given a set of specifications by the user. In order to analyze the estimated phase noise of each design, a new phase domain model of ADPLL is incorporated. For automatic design implementation, a new HLS engine with a library parser and ADPLL realization template is used. The flow is applied for four different cases and the results match circuit level simulation results. CellPLL successfully generates ADPLL designs and provides ability to move between production processes.

ÖZET

STANDART KAPILARLA TAMAMEN DİJİTAL FAZ KİLİTLİ DÖNGÜ TASARIM, ANALİZ VE ÜST SEVİYE SENTEZLEME

Bu tezde tamamen standart mantık kapılarıyla tasarlanmış bir tamamen dijital faz kilitli döngü (ADPLL) tasarımı ve örnekleme gürültüsünü bastırmak için yeni bir yöntem kullanan zaman dijitalleştiricisi (TDC) sunulmaktadır. Yeni çok giriş ve çok çıkışlı (MIMO) örnekleme gürültüsü bastırma yöntemi önceki yöntemlere göre N paralel TDC kanalı için $\sqrt{2}$ iyileştirme sağlamaktadır. Örnekleme gürültüsünün bastırılmasının hem teorik olarak hem de simulasyon sonuçlarında faz gürültüsünü azalttığı görülmüştür. Üretim teknolojileri arasında hızlı geçiş yapılabilmesi, dizaynın kolayca değiştirilebilmesi ve esneklik sağlamak için tasarım tamamen Verilog programlama dilinde yapılmış ve HDL sentezleyicisi kullanarak transistör seviyesindeki şemalar elde edilmiştir. Literatürdeki önceki yayınlara kıyasla tasarlanan ADPLL benzer silikon alanı ve güç harcayarak daha iyi faz gürültüsü sağlamıştır. Ek olarak bu tasarım için istenen özellikleri sağlayacak ADPLL konfigürasyonunu yapmak için gerekli tasarım, analiz ve üst-seviye sentezleme metodu (HLS) geliştirildi ve sunuldu. Geliştirilen tasarım yardım programı CellPLL, kullanıcı tarafından verilen parametreleri kullanarak transfer fonksiyonlarını direk olarak oluşturmaktadır. Otomatik oluşturulan döngülerin faz gürültüsünü incelemek için ADPLL'in faz modeli yapıldı. Hesaplanmış döngülerin gerçeklenmesi için HDL sentezleme kütüphanelerini inceleyen bir yazılım geliştirilmiş ve tasarlanan esnek ADPLL yapısı kullanılarak istenen özelliği sağlayan bir faz kilitli döngünün otomatik dizayn gerçeklemesi yapılmıştır. CellPLL dört farklı tasarımın gerçeklenmesi için koşulmuş ve CellPLL'in tahmin ettiği faz gürültüsü ile simulasyon sonuçlarının birbirini doğruladığı gösterilmiştir. Tasarlanan dijital faz kilitli döngünün ve geliştirilen yazılımın döngüyü hesaplayıp, performans analizini doğru yaptığı ve tasarım kodlarını doğru gerçeklediği görülmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_1	Digital loop filter pole
b_1	Digital loop filter zero
f_o	Carrier center frequency
f_{off}	Offset frequency with respect to the carrier
$f_{Ref_{max}}$	Maximum reference frequency
$f_{Ref_{min}}$	Minimum reference frequency
k	Boltzmann constant
K	Open loop gain
K_1	Digital loop filter integral gain
K_2	Digital loop filter proportional gain
K_v	Digitally-controlled oscillator gain
N	Number of channels in TDC
n_i	Quantization noise
N_i	Time conversion result for channel i
N_{max}	Maximum feedback divider value
N_{min}	Minimum feedback divider value
r_i	Received signal
s_i	Input signal
T	Temperature (K)
T_i	TDC resolution for channel i
T_s	Sampling frequency
V_{DD}	Supply voltage
V_T	Threshold voltage
w_p	Pole frequency
w_z	Zero frequency
α	Digital loop filter IIR coefficient
γ_N	N-type mobility coefficient

γ_P	P-type mobility coefficient
σ	Standard deviation
σ^2	Variance
$\Sigma\Delta$	Sigma-delta modulator
$A(s)$	Open loop transfer function
$Err[k]$	Digital phase error
$G(s)$	Closed loop transfer function
$L(f)_w$	Single sided power spectral density
$N.F$	Effective feedback divider value

LIST OF ACRONYMS/ABBREVIATIONS

ADPLL	All-digital phase locked loop
APR	Auto place and route
BW	Bandwidth
COV	Covariance
COV	Noise transfer function
DPLL	Digital phase locked loop
DPC	Digital to phase converter
DTC	Digital to time converter
DNL	Differential non-linearity
FOM	Figure of merit
GRO	Gated ring oscillator
HDL	Hardware description language
HLS	High-level synthesis
INL	Integral non-linearity
MIMO	Multiple input multiple output
MISO	Multiple input single output
PDC	Phase to digital converter
PLL	Phase locked loop
PN	Phase noise
PSD	Power spectral density
PVT	Process voltage temperature
RNG	Random number generator
SIMO	Single input multiple output
SISO	Single input single output
SNR	Signal to noise ratio
TDA	Time difference amplifier
VCO	Voltage controlled oscillator

1. INTRODUCTION

Phase locked loops (PLLs) are being extensively used in today's wireline and wireless communication products as part of data recovery circuits, clock multipliers, and frequency synthesizers. With the increasingly tougher specifications of modern communication circuits, there is a constant push to develop small and low power PLLs while satisfying strict frequency spectrum specifications [1].

During PLL design, another driving factor is the design cycle time which does not allow the re-design and complete analysis of PLLs in many cases. Hence, it is crucial to develop a framework in which designs are modeled, analyzed for phase noise, and implemented within a guided flow. This allows minor modifications to be implemented rapidly with reduced risk, enables transfer of designs between technology nodes, and allows fast new design space exploration to identify what is possible within a technology.

In automotive industry, the demand for electronics has been increasing rapidly in the last decade. These dice are specified to be very high quality microelectronic chips that conform to rigorous automotive standards. All of the systems in cars are either being replaced by or being digitally assisted by microelectronic circuits. One of these areas is safe driving which includes various applications from assisted braking to surround view camera systems. Today some cars have several cameras whose outputs are combined in the processing units of the automobile. Imaging sensors can be placed at various locations around the car and these camera modules need power, communication facilities and a video link to the central processing unit of the vehicle.

Parallel video output from imaging sensors has to be transmitted using a wireline interface which generally requires a clock multiplying PLL. As the features such as block level calibration, auto white balance, and color correction in imaging sensors get more and more complex with each new sensor, the need to go to smaller process nodes emerge. This requires the redesign of analog components in the new process node. In

order to decrease the analog circuit redesign effort, ADPLLs are preferred.

In general, a PLL adjusts its oscillator in phase and frequency to track the reference clock input by comparing it to the feedback divider output. When locked, this allows the closed loop system to generate an output clock that is related in frequency and phase to the reference clock with a multiplication factor. The steady state phase error and input tracking capability of a PLL is determined by the order and bandwidth of the loop transfer function. On the other hand, the output frequency resolution is determined depending on the feedback method employed.

PLLs employ two types of feedback methods called integer division and fractional division. In integer-N method, a high frequency output clock is divided by an integer and the result is used as the feedback signal. On the other hand, the fractional-N method uses multiple division values that are switched between various integer values in order to create an effective fractional division value. An extra design effort is needed to overcome the inherent drawbacks of the fractional-N PLLs due to the noise created by the modulated feedback divider. However, due to their finer frequency resolution, fractional-N PLLs are increasingly utilized in order to satisfy the highly demanding modern design specifications. In order to have a clear understanding of all the noise sources in the design loop, it is crucial to extend the design flow and phase modeling to cover the divide value variations [2].

As a subset of PLLs, the fractional-N All Digital PLLs (ADPLLs) in Figure 1.1 are specifically suitable as a rapid design framework because of their standard cell architecture. Standard cell architectures allow reuse, provide flexibility, and ease scaling with technology migration. It is seen that ADPLLs lack a general phase model with various phase noise contributors such as:

(i) Inherent digitally-controlled oscillator noise
(ii) Digitally controlled oscillator (DCO) quantization noise
(iii) Quantization noise from digital phase detection
(iv) Quantization noise from the $\Sigma\Delta$ modulator

Additionally, a methodology for generating the loop transfer functions for given specifications and support of High-Level Synthesis (HLS) are required for ADPLL design flow.

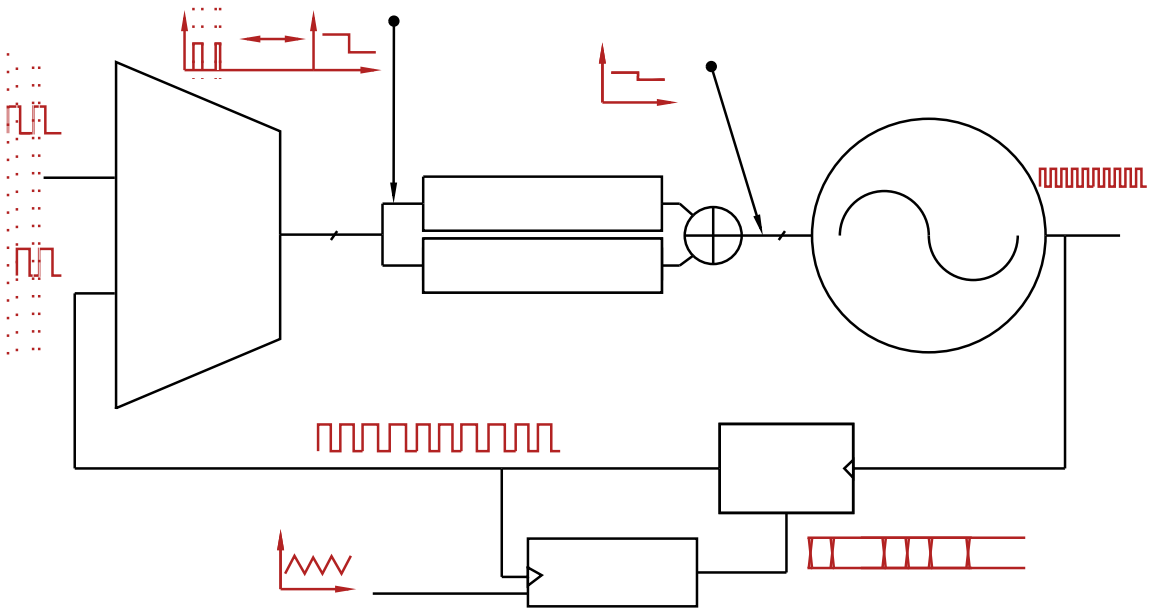


Figure 1.1. Fractional-N all-digital PLL

A fractional-N phase locked loop (PLL) allows generation of any desired clock frequency from a given constant reference clock source. In some high data rate baseband systems such as SAS, SATA, PCIExpress and DisplayPort, the serial communication link bit-rate over the cable is kept constant. In such systems, data or video throughput is variable and consumes only a portion of the available bandwidth. This requires the regeneration of the video clock on the receiver side from the recovered link clock using a fractional-N PLL. In other applications where serial link rate is variable, these types of PLLs are still useful when they act as fractional clock multipliers or spread spectrum generators. Fractional clock multiplication helps extract a certain fraction of the incoming throughput and route it to the output of the receiver.

Video sinks without buffer such as display panels or video timing controllers require low long term jitter from fractional-N PLLs because the video flow is continuous and the tolerance of throughput variation is low due to the huge storage requirement

that would otherwise be necessary with an uncompressed high definition video flow. As most ADPLLs have relatively poor jitter performance due to the discrete steps in their oscillators, such low long term jitter requirements have been traditionally met with analog or digitally assisted PLLs. Similar to many other circuit types, PLLs have traditionally been implemented using analog circuits. However, area hungry circuit components such as the capacitors in the loop filter and poor g_{ds} of transistors pose potential problems against larger scale integration in finer process nodes [1]. This creates the need for digitization of PLLs with cheaper digital resources. Previous work [3–5] on the digitization of PLLs has contributed on the following:

- Phase detection is implemented digitally by a TDC
- Voltage-controlled oscillator is replaced by a DCO
- Loop filter is in the digital domain instead of the analog

As there is an increasing need for packing more digital processing functions into such video processing transmitters and receivers, the need to move to finer process nodes emerges. Additionally, various video interface standards such as HDMI, CSI, DisplayPort, DSI, LVDS and OLDI need various configurations for PLLs. These two motivations push for the need to create ADPLLs that are easily configurable and RTL intensive. However for fractional-N PLLs, the digitization has been limited to some of the components in the design and approaches utilizing only standard cells have just recently been published [6, 7].

In [2], there is a general phase model for PLLs, however, it is targeted for analog PLL architectures. This thesis models phase noise of the sub-components of a general fractional-N ADPLL architecture. This extended phase noise model for the ADPLLs is created in an effort to reduce the need for time costly transient simulations. Similarly, in [8], an automatic loop design method has been presented, but this method concentrates on analog PLLs while an improved loop generation method for ADPLLs is illustrated in this work.

There are tools for HLS of digital circuit types other than ADPLLs [9,10]. However, this internally developed tool (CellPLL) adds a HLS engine for the first time to the design flow for ADPLLs. HLS accesses the loop design automatically and generates Verilog code together with synthesis scripts for the register transfer level (RTL) synthesizer. During HLS, CellPLL uses the internally embedded ADPLL template [11]; hence, this limits the set of specification space within template's boundaries. However, as the loop design generator and the phase model are independent of the template, any standard cell ADPLL implementation can be embedded in CellPLL in order to support a different subset of possible ADPLL specifications.

This work develops a general phase model and loop generation algorithm for proposed ADPLL design together with HLS engine. Chapter 2 provides details on the status of the literature on the chosen topic. Chapter 3 and Chapter 4 explain architecture, phase noise model, and HLS support for digitally-controlled oscillator (DCO) and Time-to-digital Converter (TDC) respectively. Chapter 5 details closed and open loop transfer function generation, loop design, phase noise model, and high-level synthesis support for the top-level of the ADPLL. In Chapter 6, various use cases with different specifications and process nodes are generated. Additionally, comparison results with circuit level simulations and other tools used in the industry are reported. Finally, we draw conclusions in Chapter 7.

2. BACKGROUND

This chapter presents a comprehensive literature survey which was done by reading through 101 articles about the latest PLL architectures, techniques and methodologies [1,3–5,12–86]. Strong emphasis is on the digitally assisted and all digital phase locked loop architectures. In [41], the reasons why ADPLLs have gained increased attention has been summarized very clearly as follows. "Phase locked loops have been one of the last remaining resorts that have not been conquered by digital approaches until the past several years have brought all digital techniques to the RF domain. Digital gates' switching activity has created excuses for RF engineers not to go to digital domain by pointing to the high sensitivity and high dynamic range requirements of these circuits. Digital logic with its cheap and powerful existence could not be avoided with the smaller and smaller process nodes and pressure for integration. Therefore, digital logic started to penetrate every possible domain of the RF world either by transforming RF functions directly into digital or by assisting analog circuits for better performance". In the rest of the section, a deeper analysis in to these motivations will be presented. This section gives an introduction of the digitization of conventional PLL and points to new and worthy topologies, methods and techniques in latest ADPLL technology.

In the recent years, increased attention to the phase locked loops by the academic researchers and the industry has been observed. The ongoing work related to conventional PLLs has continued for optimizing the readily available topologies. Moreover, as the digital circuits got faster with ultra-scaling of CMOS devices, digitally assisted PLLs (DPLL) and all-digital PLLs have started to emerge. A literature survey of the past several years compiled from prospective publications, indicate that the inclusion of digital into PLL domain has grown drastically. In Figure 2.1, one can deduce that the digitally assisted or all-digital PLLs have captured almost 68% of the research interest between 2009-2016.

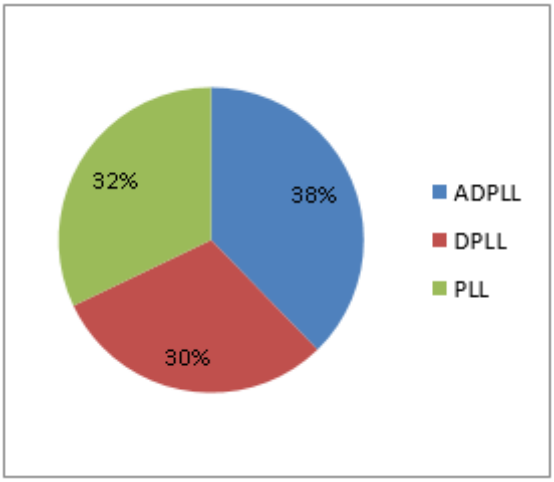


Figure 2.1. Research focus by PLL type

Considering the fact that research in this area is mainly pushed by the industry, one can conclude that the published articles concentrate on industry needs. Even though fractional-N PLLs have emerged later in research history, from clock recovery, tracking, and generation perspectives, it is seen that there is almost a balance between the need for integer-N PLLs and fractional-N PLLs in Figure 2.2a. This is a result emerging from the fact that fractional-N PLLs may not always be the right choice for the tightly specified requirements of applications in terms of area, power, spur generation and phase noise.

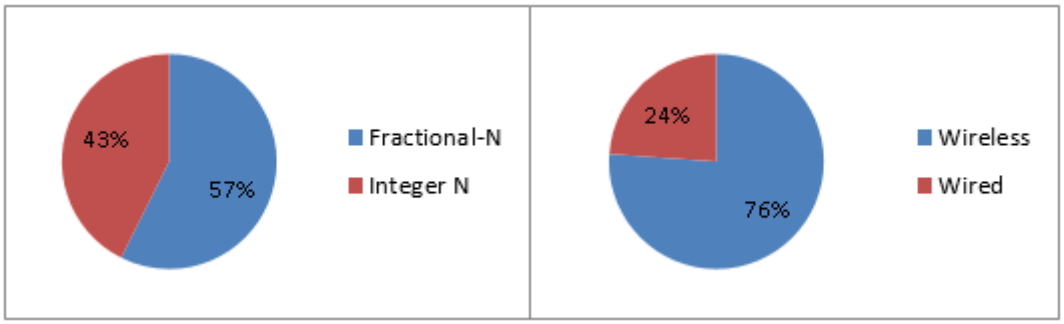


Figure 2.2. a) Research interest in F_{out}/F_{ref} ratio. b) Research interest in application area

Another result from the literature survey done in this thesis is that the push by industry directs the research towards topologies mainly targeting wireless applications

rather than the solutions targeted for wireline applications. From Figure 2.2b, it can be seen that most of the effort is spent on wireless applications. Evidently, the application of choice pushes the focus towards the more important design parameters of this application. As it can be seen from Figure 2.3, a dominating percentage (73%) of the contributors in the literature are aiming to make improvements over state of the art designs in terms of lower phase noise in higher loop bandwidths. Moreover, low phase noise targets are always seen to be accompanied with heavy spur reduction effort. When the work on low area is considered, one can see that the low area occupation efforts usually do not completely neglect the need for a relatively good noise/spur performance. On the other hand, it can be concluded that the papers focusing on low power consumption frequently deprioritize either the area consumption or the noise/spur performance of the design.

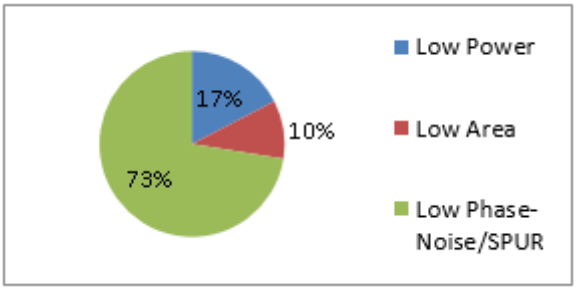


Figure 2.3. First aim of the designers while searching for good performance

State of the art conventional PLLs offer good results in terms of the stringent specifications of wireless applications. However, some digitally assisted PLLs and most all-digital PLLs are still struggling to catch up to their analog counterparts in terms of the performance results. This is considered as the main reason for increased focus on reducing phase noise and the spur levels in recent years. Even though time is still needed to replace analog PLLs completely with their ADPLL counterparts in a widespread fashion in the industry, one can see that some commercialization of ADPLLs is already happening in isolated product lines such as mobile communications market where the push for deep submicron scaling and integration is highest along with very tight area and power consumption requirements.

2.1. Main research topics in PLLs

In Figure 2.4, main research topics in PLL research are given with a break down in terms of PLL type. It can be seen that most of the attention is paid to performance optimization. In Section 2.2, we present the summary of recent research striving for better performance. Additionally in Section 2.3, we provide a summary of the auxiliary research topics on PLLs.

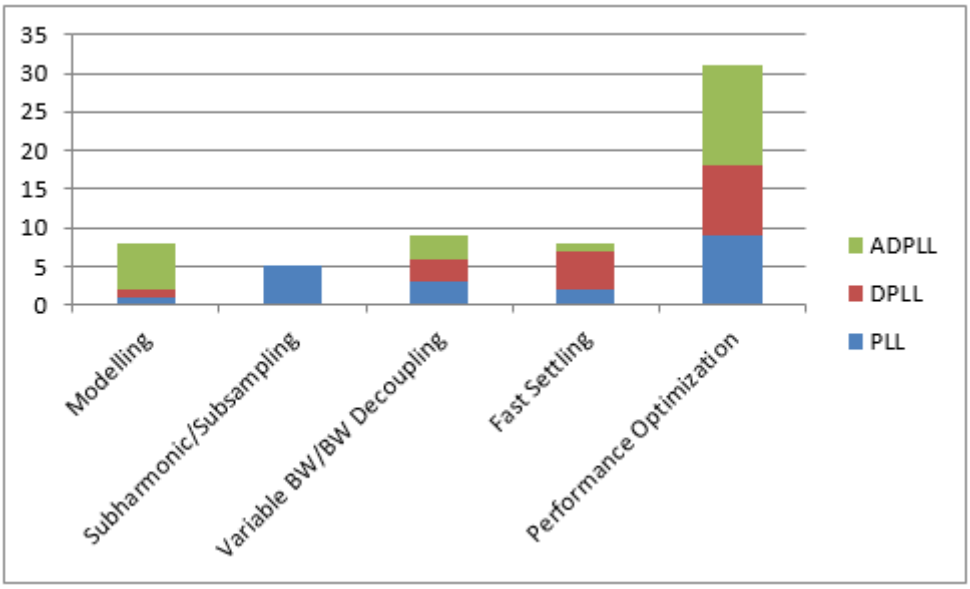


Figure 2.4. Main research topics

2.2. Performance optimization techniques and methods

There are several techniques that have been employed for increasing the performance of all types of PLLs. These techniques usually aim to find solutions for PVT mismatches, fast settling time, lower phase noise, and lower spur levels. Better noise performance is generally pursued with increased linearity, lower phase/spur levels by decreasing the effects of quantization noise, or by pushing the noise energy out of the loop bandwidth. As seen in Figure 2.5, as the digital content of the PLL increases, the need for such measures also increases. A brief summary of each mentioned method is given in the following sections.

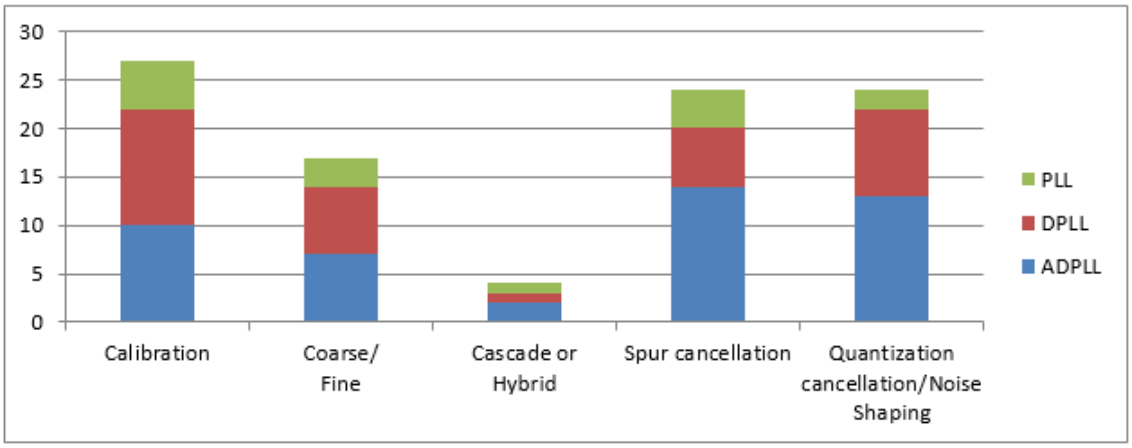


Figure 2.5. Performance enhancement techniques employed with respect to PLL type

2.2.1. Calibration

The need for calibration in order to decrease the effects of PVT mismatches has increased mainly with the introduction of DCOs and TDCs with ring oscillators. The basic inverter gate delay varies greatly on chip due to PVT variation. This introduces pronounced non-linearity in the mentioned sub-blocks as well as low DNL/INL values during digital frequency control or time-to-digital conversion. We can classify these types of circuits as online or offline. Online algorithms continuously check for PVT mismatches and therefore are valid for process, voltage, and temperature. However, offline algorithms can run at start-up only but at the expense of being able to compensate against process variation only. Several calibration techniques use redundancy at the cost of power and area while another technique does gate shuffling in order to spread out the non-linear quantization effects (spurs) of the PVT mismatches as the expense of larger phase noise floor.

2.2.2. Coarse and fine capacitor bank methods

Recent survey shows that the use of coarse varactor banks along with fine tuning varactor banks are preferred instead of supply starvation methods. Varactors can also be implemented using input capacitances of CMOS gates or switched actual capacitors.

This is mainly used for achieving fast settling time by searching through the coarse bank first and adjust fine varactor bank as the second step. Additionally, it has been seen that process mismatches in varactor banks are compensated to some extent by using dithering in the lower bits of the bank corresponding to the fine tuning bank and better resolution has been achieved through fractional digital word control of the bank.

2.2.3. Spur cancellation and quantization noise reduction

In ADPLLs there are several sources of quantization noise and non-linearity arising from sampling with PVT mismatches. While digital calibration methods are used in order to decrease these side effects, they are not enough. Around fractional division values near integer-N values, these effects are especially magnified. Therefore, several methods for decreasing spur levels have been reported. While some of these methods try to spread spur signal power to a broad band of frequencies and convert it to phase noise, some try to find out the exact location of the spur dynamically through digital algorithms in order to cancel them out. In terms of quantization noise, most of the methods try to increase effective resolution to improve noise floor and some reported structures try to shape the noise signal out of the band of interest by introducing history to the DCOs and TDCs such as the one done in gated ring oscillator.

2.2.4. Power supply rejection improvement

In the recent years, the decrease in the supply voltages has been particularly important. In FINFET technologies, supply voltages are extremely low and these type of processes are generally used for digitally heavy SOC architectures with poor switching noise isolation. Therefore, new supply rejection methods that track and compensate the supply noise have been implemented similar to the one in [28].

2.3. Supplementary research topics in PLLs

2.3.1. Modeling

Analytical ways to model characteristics of different PLL topologies in terms of noise transfer characteristics, theoretical limits on various building block families and procedural design tutorials are presented.

2.3.2. Sub-harmonically locked, sub-sampling and injection locked designs

PLLs that use this architecture usually target ultra-low phase noise and spur level performance, and are valid almost solely for analog PLLs. This type of PLL works with unconventional loop update rates. Recently, injection locked multiplying delay locked loops (MDLL) have gained attention especially in multi-core CPU architectures which introduced the start of digitization in this sub-category.

2.3.3. Variable BW and BW decoupling designs

Similar amount of research activity for all PLL types is observed in this category. Variable BW designs mainly focus on on-the-fly adjustment of loop BW for utilization in achieving fast settling time or noise suppression. While rarely employed, BW decoupling is the decoupling of loop bandwidth from the input frequency modulation BW. These type of PLLs allow tracking a wide-band input signal while keeping the loop phase noise levels low.

2.3.4. Hybrid and cascaded loop designs

It has been seen that combining the pros of different loop types is desirable. While digital loops offer lower area and faster lock time, they are still struggling to achieve noise performance levels as good as their analog counterparts. Therefore, hybrid loops where coarse locking is achieved by a digital PLL and the tracking phase is continued by a low BW, low noise analog PLL is reported. When we look at the

main drivers for the ADPLLs, we can see that this approach solves only fast lock problem. In terms of portability, area, and power issues still exist. Moreover, some cascaded loops have been reported in pursuit of achieving lower area and power by decreasing the required dynamic range of building blocks such as DCOs and TDCs with the help of the cascaded structures.

2.3.5. Fast settling design

Fast settling PLLs generally try to reduce the lock time of the PLLs using variable BW methods. However, can also be done by ingenious search algorithms and cycle slip (edge miss) compensators.

2.3.6. New architectures for sub blocks

New ideas are presented to implement some parts of the desired transfer functions with new circuit types. For example, a new oscillator based integral loop generation is presented in [31] and a new fractional divider is presented in [14].

2.4. Background on oscillators

State of the art PLLs increasingly employ digitally-controlled oscillators. Regardless of the PLL type there is a strong inclination towards digital frequency selection by enabling varactors especially for frequency synthesis applications. One can observe that digitally assisted PLL designs try to employ preferably ring oscillators while conventional PLLs still favor LC tank based oscillators.

Another recent trend for ADPLLs is the incorporation of the TDC in the DCO. This has emerged from the similar structures employed in both TDCs and ring DCOs. Designers try to share common hardware such as ring oscillator and delay line for these blocks in order to get better area and power performance while trying to minimize PVT mismatches and leakage. Moreover, several digital calibration techniques are used in DCOs for decreasing the effects of PVT variations and non-linearity in order

to suppress the generation of frequency spurs and elevated phase noise floor. In Figure 2.6, a breakdown of preferred oscillator type is given.

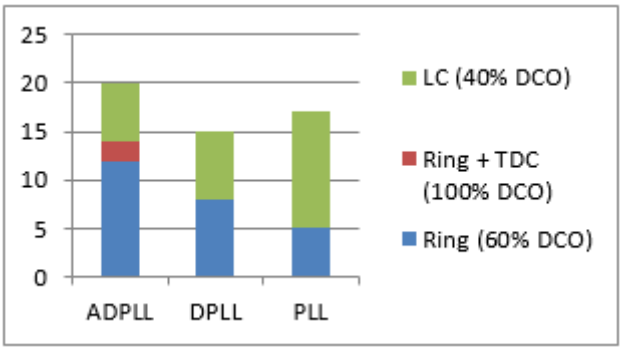


Figure 2.6. VCO/DCO choice by PLL type

2.5. Background on phase detection

In Figure 2.7, a list of TDC types are tabulated according to their resolution. Dynamic range performance is not listed as several works that were surveyed did not report number of TDC output bits. It can be seen that there are various techniques employed while trying to get PVT insensitive sub-gate delay TDCs. State of the art TDCs are trying to go below 5 ps of resolution with acceptable INL, DNL performance.

There are several reasons for trying to build a better TDC. Increasing the resolution of the TDC is important for improved phase noise performance. Better linearity is measured by INL and DNL, and is important for generation of monotonic phase noise profile without frequency spurs. Larger dynamic range is important as it allows faster settling of the loop.

Journey of TDC design started with delay lines which have been followed by Vernier delay lines. Additionally, ring type delay lines followed by noise shaping gated ring type delay lines (GRO) have been used. Recently, increasing number of techniques for PVT calibration that use scrambling, stochastic approaches, correlation, and adaptive filters have been introduced. Also time amplification (TDA) techniques using meta-stability of the gates are trying to zoom in to increase time resolution. Intro-

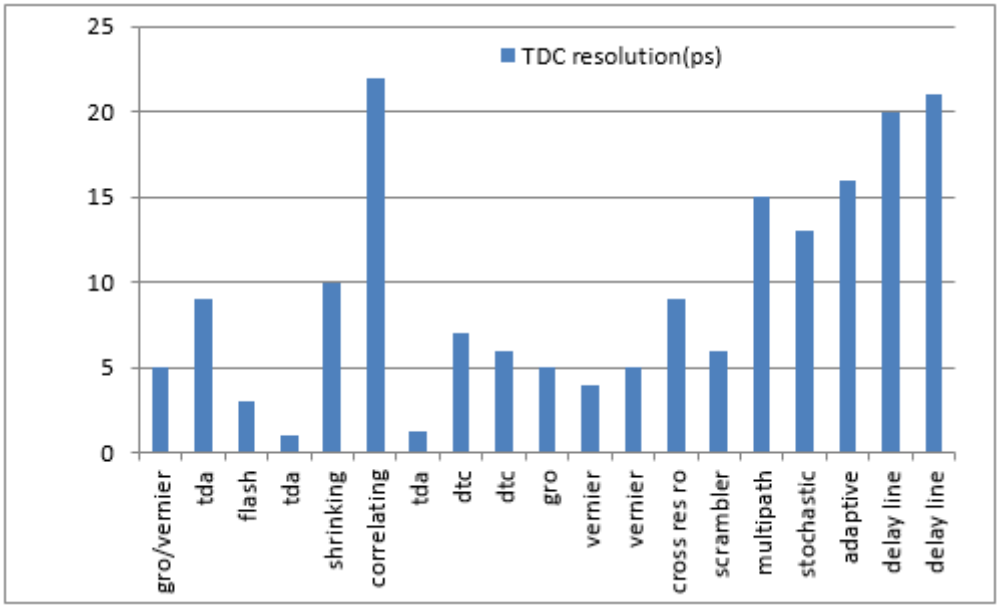


Figure 2.7. TDC Types and state-of-the-art performance

duced TDA methods have been shown to have best resolution performance. Another recently introduced TDC types called digital-to-time converter (DTC) and digital-to-phase converter (DPC) convert digital codes to "time or phase" for comparison rather than time difference to digital. This type of PFD operation has been reported to achieve good results. In this thesis we concentrate on multi-path TDCs and propose an SNR improvement on such designs by exploiting ideas employed in beam-forming antenna design. This can be achieved by exploiting correlated phase error measurements at multiple paths of parallel TDC paths.

TDC is not the only building block that can be used to replace PFD for building an ADPLL using digital loop filter. Accumulator or phase-to-digital converter (PDC) based topologies also exist with a limited number. From Figure 2.8, it can be seen that TDC based topology spans 70% of the digital PLL structures.

2.6. Power vs. phase noise analysis

As mentioned in the previous sections, phase noise is generally the first target of the designers. We can conclude that the designs try to satisfy spectral mask re-

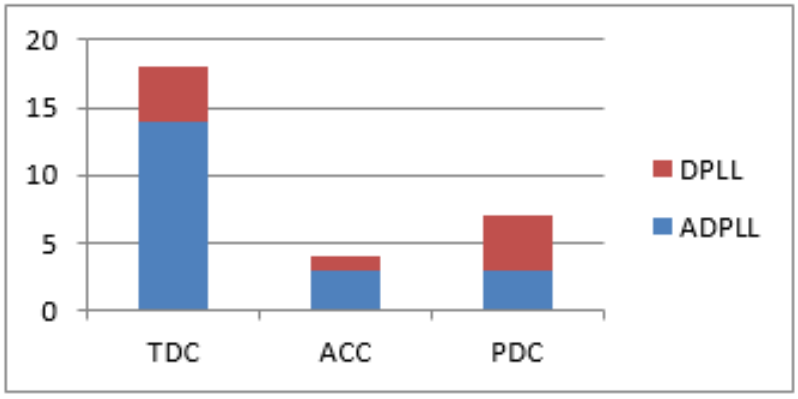


Figure 2.8. Digital PFD techniques in ADPLL and DPLL

quirements of wireless communication specifications as the ultimate goal as their low offset frequency phase noise level requirement is the harshest. While analog PLLs have long satisfied this goal, ADPLLs are still trying to close in to this requirement. Currently, most of the state of the art ADPLL designs report phase noise around -100 dBc/Hz@1MHz frequency offset by employing many performance enhancement techniques. Figure 2.9 gives the latest phase noise levels for a range of power consumption values for all PLL types. From this figure we can clearly see that a dominant portion of research papers report results at 1 MHz offset from the center frequency and phase noise results are mostly reported between $< -80, -120 >$ dBc/Hz with power consumption levels lower than 20 mW. In this set, the ones that use performance enhancement techniques in a combined fashion can achieve results below -100 dBc/Hz. Moreover, the designs that do not use redundant digital logic in TDC and DCO oscillator blocks or the ones that can shut down the power hungry blocks when not in use can go down to power consumption levels smaller than 5 mW. Therefore, to improve on state-of-the-art performance given in recent publications, one should be able to provide phase noise levels lower than -100 dBc/Hz@1MHz with 10 mW or less power consumption.

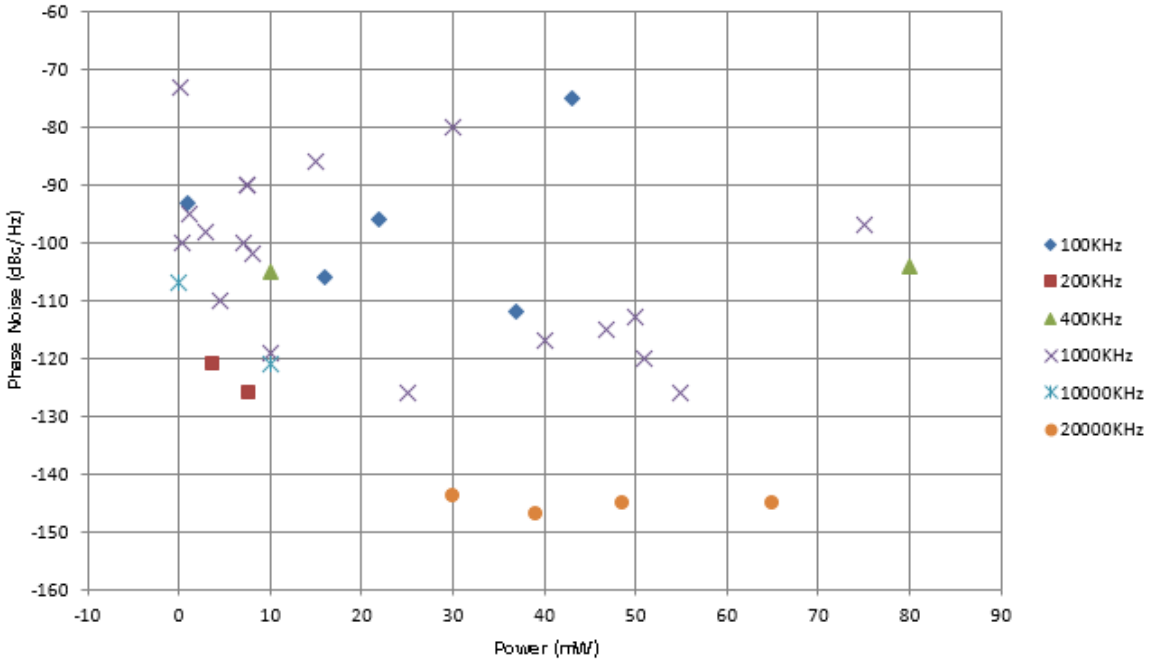


Figure 2.9. Power consumption vs. phase noise performance at various offset frequencies in recent literature for all types of PLLs

2.7. Area vs. phase noise analysis

Figure 2.10 shows the area occupation of state of the art PLLs vs. phase noise performance. Similar to power consumption analysis section, phase noise targets are the primary concern. Therefore, phase noise results mostly concentrate between $< -80, -120 >$ dBc/Hz@1MHz offset. It is seen that, in this phase noise performance range, most of the designs have less than 1 mm^2 area. Moreover, the better designs accumulate in a cluster that has an acceptable phase noise performance smaller than -100 dBc/Hz@1MHz with less than 0.1 mm^2 area consumption.

2.8. Figure of merit (FOM)

2.8.1. Analog PLLs

Figure of merit in an analog PLL is the normalized minimum achievable phase noise of the phase detector. It is measured in units of dBc/Hz. Assuming that N is

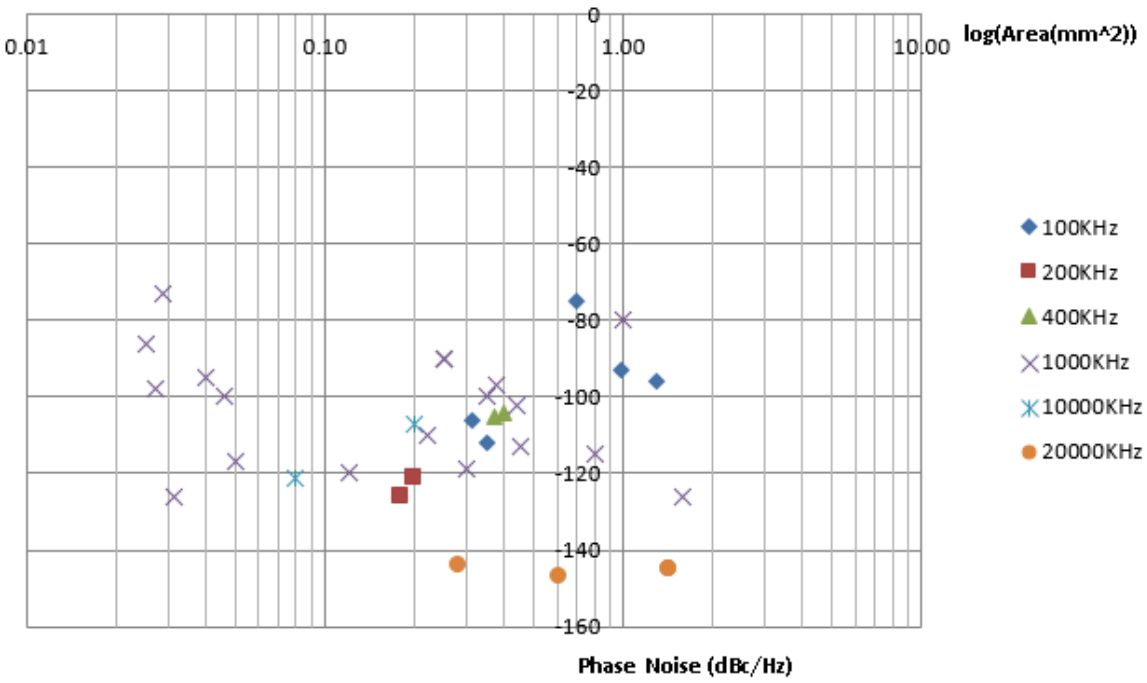


Figure 2.10. Phase noise vs. area consumption at various offset frequencies in recent literature for all types of PLLs

the feedback divider value and F_{pfd} is the phase detector frequency, the phase detector noise floor is approximated by subtracting $20logN$ and $10logF_{pfd}$ from in-band noise of the VCO output. In other words, after normalization $PN = PN_{tot} - 10logF_{pfd} - 20logN$. Overall effective close-in phase noise for a PLL (dB) can be estimated as follows:

$$PN_{total} = PN_{synth} + 20logN + 10logF_{pfd} \tag{2.1}$$

PN_{total} is the overall effective phase noise of the PLL. Due to the PLL frequency synthesizer itself, the phase noise is PN_{synth}. $20logN$ is the addition of phase noise because of the higher frequency multiplication related to the feedback ratio N. $10logF_{pfd}$ is the negative effect of increased incoming PFD frequency on noise.

Figure of Merit or FOM is often defined as the PN_{synth}. This rips off the noise contribution effects of PLL N value and PFD frequency from the synthesizer circuit and provides a normalized figure of merit. Therefore it would allow comparison be-

tween PLLs at different configurations. FOM for a PLL with VCO running at 3.932 GHz is given below.

$$\begin{aligned} FOM &= 220 \\ Feedback\ divider &= 32 \\ Phase\ detector\ rate &= 122.88\ MHz \\ PN_{total} &= -220 + 20log(32) + 10log(122.88\ MHz) \\ PN_{total} &= -220 + 30 + 81\ dBc/Hz \\ PN_{total} &= -109\ dBc/Hz \end{aligned}$$

This means that the user should see the tail noise of the carrier to be approximately -109 dBc/Hz at the 3.932 GHz output.

2.8.2. Digital PLLs

In this literature survey, a commonly used FOM adopted by many researchers has recently been encountered. Proposed methods seen in the literature used combinations of jitter, power and area in ambiguous ways. We believe that there is a need for creating a strong FOM that takes into account phase noise, area and power is required for ADPLLs in order to better analyze the trade-off mechanisms. Table 2.1 shows examples of reported values for the most commonly used FOM definition. It is seen that only jitter and power has been considered as in Eq. 2.2 [26].

$$FOM_1 = 10log(\frac{Jitter\ (\sigma^2_{rms})}{1s^2}\frac{Power}{1mW}) \tag{2.2}$$

$$FOM_2 = 10log(\frac{Jitter\ (\sigma^2_{rms})}{1s^2}\frac{Area\ (mm^2)}{Tech^2/0.18^2}\frac{Power}{1mW * (Output\ Frequency\ (MHz))}) \tag{2.3}$$

We believe area and operating frequency should also be included in this definition. The power has been scaled by frequency (mW*MHz) and the area has been scaled by the

technology. The proposed figure-of-merit (FOM) for comparing ADPLL architectures is as in Eq. 2.3.

Table 2.1. Recently reported FOM values in the literature

	[87]	[88]	[89]	[90]	[91]	[92]	[60]
FOM(dB)	-221.6	-224.8	-211.4	-206.1	-226.1	-234.1	-242.2

2.9. Focus of the thesis

As a result of the conducted literature survey, we believe that the industry needs portable mixed-signal circuits more than anything for sustainable integration of circuits in cutting edge processes. As high performance analog integrated circuits need increasing amount of digital logic, it becomes mandatory to port all the analog circuits in to smaller process nodes in order to incorporate sufficient digital logic. In this selected research topic, the main aim is to use standard cells and RTL digital design flow along with digital back-end design as much as possible.

The chosen topic not only includes various digital calibration, noise/spur cancellation techniques within, but also inherently contains the opportunity to develop EDA tools that would generate such RTL based ADPLL circuits according to given parametric input specifications.

This research topic is pursued in the light of the following priorities and choices. Area and power consumption is second priority compared to digitization of the PLL. Fractional-N architectures are favored above integer-N architectures. Concentration is on frequency tracking and multiplying PLLs rather than frequency synthesizers. The thesis concentrates on ADPLL architectures with an emphasis in RTL development and synthesis, and implements a new architecture under specific performance specifications. Additionally, we increase performance of previously known methods and propose new architectures with novel calibration and noise cancellation techniques. Furthermore, we concentrate on automatic ADPLL calculations and high-level synthesis according to user specifications including the phase noise with the help of flexible IP development.

3. DIGITALLY-CONTROLLED OSCILLATOR (DCO)

A DCO is an oscillator type where the frequency tuning control is done with a digital control word. While a DCO internally does not have to be digital for a digitally assisted PLL (DPLL), this work uses a standard cell architecture in order to be able to demonstrate HLS, modeling and generate a standard cell ADPLL.

An architecture suitable for automated generation in various process nodes for the same specifications is targeted. Given characterization of the standard cells in Liberty libraries, it is possible to generate similar performance DCOs in the selected process nodes. Compared to high frequency wireless mobility products that push for absolute performance, compromises in performance specs can be acceptable in wireline communication circuits. As the cable lengths in the vehicles are short and shielding is quite strong, requirements for phase noise and spur performance can be relaxed. Similarly, compromises in power and area due to all digital implementation are also possible due to abundance of energy and space in plugged devices.

A novel digitally-controlled oscillator (DCO) architecture which adjusts driving strength rather than capacitance for coarse and fine tuning in ring oscillator architectures has been proposed as part of an effort to fully digitize all-digital phase locked loops (ADPLL). There is previous work related to synthesizable DCOs in [6, 7], but [7] uses custom standard cells during synthesis and [6] uses supply starvation methods. The proposed DCO in this study is novel in the sense that it implements a new calibration scheme and is implemented in all-digital design flow compatible with synthesis, Auto Place and Route (APR) and usage of only standard library cells. Portable RTL code that is parametric in terms of PVT calibration and coarse tuning has been developed. Delay cells in rings use gate level HDL implementation for fine tuning. Except for a couple of recent papers, examples in the literature claim to be all-digital, but essentially they either contain custom gates [7] or are digital only at the block interface as in [3–5, 53] and operate in an analog fashion internally. The only truly digital DCOs in the literature are the ones that use tri-state buffers [91] similar to the DCO

presented in this study or the ones that utilize the Miller effect [17] and tune the frequency.

3.1. Architecture

The DCO in Figure 3.1 incorporates N ring oscillators. This architecture is selected in order to provide a standard cell based flexible design with a large frequency tuning range of $0.65 - 1.35$ GHz at the expense of larger power and area consumption due to the multi ring architecture. Every ring oscillator uses the same number of three, five or seven programmable delay cells rather than basic inverters for creating an oscillator loop. An offline calibration algorithm is deployed for use before the ADPLL starts using the DCO. During calibration, an externally provided clock source is used to measure the free running oscillation frequency for five delay cells in a ring while using the center frequency control word. If the oscillator frequency is slow due to process, voltage or temperature (PVT), the delay cell count in the rings are reduced to three. Similarly, if the oscillation frequency is initially too fast, the rings are programmed to use seven delay cells. Depending on the calibration result, unwanted delay cells are bypassed using multiplexers and the desired number of delay cells is connected to create a ring.

Each ring has tri-state buffers at each delay element output and all of the rings are connected in parallel at the output of delay elements. Each ring has a unique and one bit drive enable signal that enables all of the delay elements. The nodes driven by multiple drivers create the main time constant for each delay stage as the capacitance from every active or inactive ring's driver and next stage input is summed. Frequency tuning is achieved by changing the effective resistance at each high time constant node by enabling more or less rings while the capacitance is the same. Tri-stated rings work as capacitive load; otherwise when their drivers are active they increase the driving strength, thereby increasing the output frequency of the loop by decreasing the time constant at the output node of every delay element. By adjusting how many of the rings are active, coarse frequency tuning is obtained. Additionally, each delay cell has a unique fine frequency control ($FCW[3:0]$) that allows the delay of each delay cell

to be adjusted in fine steps. There are seven FCW signals connected to each delay element in a ring and this signal is shared in all rings. Except "0000" all FCW values can be used to provide slightly tuned delay variations using the inherent propagation delay difference between the inputs of the gates. LSB bits of the linear F_{ctrl} binary vector are mapped to the non-linear FCW signals of delay cells for each delay cell in the ring separately. Combining coarse and fine frequency control mechanisms provide the tuning control for the DCO. [93, 94]

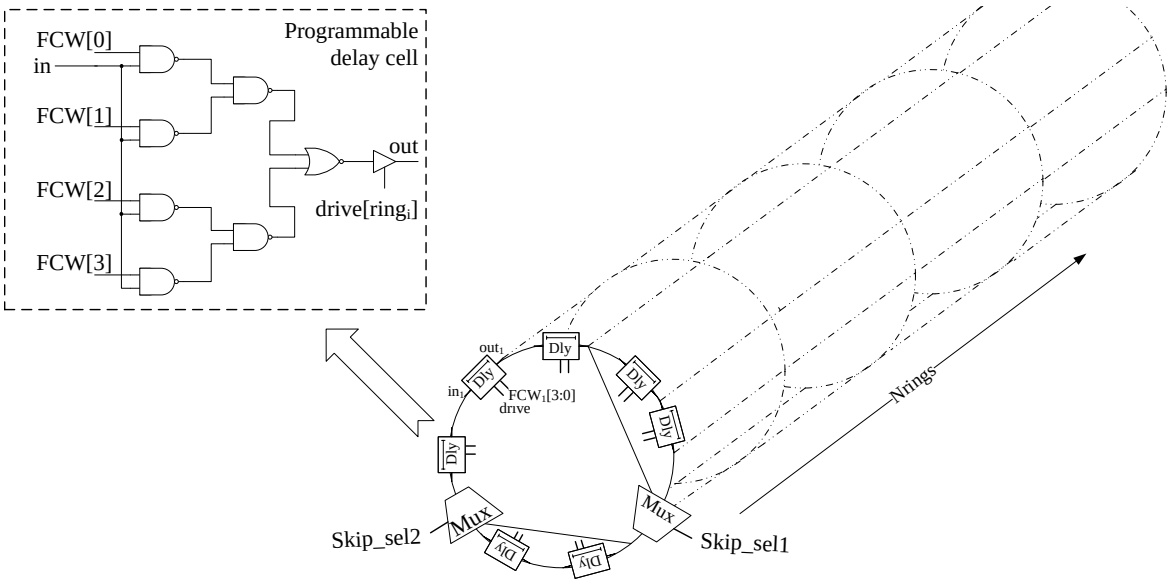


Figure 3.1. Digitally-controlled oscillator and delay element

3.1.1. Specification of input - output frequency range

High Definition imaging sensors have output pixel clock frequencies from 100 to 130 MHz; hence, the DCO range needs to be specified to cover 10 times this frequency range assuming that each pixel is 10 bits. The number of stages in each ring, the number of rings, and the fine/coarse tuning dynamic range of the oscillator need to be selected according to the desired output frequency range of $0.65 - 1.35$ GHz manually or using a HLS algorithm.

3.1.2. Determination of fine tuning frequency steps

In phase locked loops, traditionally analog voltage controlled oscillators (VCO) have been used. As the output frequency of VCOs is linearly related to the input voltage, there is no frequency step per LSB. As the DCOs try to mimic VCOs, small frequency steps per LSB are desired in order to diminish high frequency jitter that would otherwise occur in the output clock. Phase detector counterpart in ADPLLs is called a time-to-digital converter (TDC) and its resolution should be selected together with the resolution of the DCO in order not to over-design frequency steps that cannot be efficiently utilized by the ADPLL.

3.1.3. Amount of process, voltage, temperature (PVT) and mismatch variation in the process

The oscillation frequency difference for the ring oscillators between CMOS 65/55 nm PVT Corners is found to be as high as $\pm 50\%$ around the typical corner. Moreover, it is observed that mismatch is also effective and its effect can be observed as $\pm 15\%$ frequency variation around the PVT corners. As PVT and mismatch variation is very high in the process, several PVT calibration and coarse tuning mechanisms have to be adopted during the design of the all standard cell DCO.

3.1.4. Random phase noise characteristics

Phase noise is an important measure affecting the RMS jitter in the output eye diagram of the DCO. Open loop phase noise of -80 dBc/Hz@100kHz offset frequency is specified as the bottom line for this application in order to be in the ballpark of state of the art DCOs in terms of jitter. Phase noise characteristics of proposed ring oscillators using different strength standard cells have been simulated using Periodic Steady State (PSS) mode with Phase Noise (PNOISE) simulations. In order to satisfy the phase noise specification of the ring oscillator, cell strengths can be appropriately adjusted in the Verilog code manually or with the HLS algorithm given in Figure 3.4 to satisfy user specifications.

3.2. Implementation

The DCO in Figure 3.1 is composed of 256 rings with variable number of fine-tuned delay elements. Each ring has tri-state drivers at each delay element output and all of the activated rings drive the same node. Each delay element offers a fine tuning range by using the difference between gate inputs and simultaneous feed of transitions to multiple gate inputs. Cascading selectively $3, 5$, or 7 delay elements according to the estimated PVT point, a ring is composed. The nodes driven by multiple drivers create the main time constant for each delay stage as the capacitance from every active or inactive ring's driver and next stage input is connected to each other. Frequency tuning is achieved by changing the effective resistance at each high time constant node by switching in more or less drivers by keeping the capacitance the same. In order to overcome mismatch problems, all rings are connected at the output nodes of delay elements to each other. When disabled during coarse tuning, the tri-stated rings work as capacitive load; otherwise when active they increase the driving strength, thereby increasing the output frequency of the loop by decreasing the time constant at the output node of every delay element.

3.2.1. PVT calibration and coarse tuning method

In order to account for variation in oscillation frequency due to the PVT corners, a 13 MHz crystal oscillator generated ideal clock is used for creating an offline calibration algorithm. Selection of initial number of fine tunable delay stages in a ring is selected by counting the number of DCO output clock cycles during a crystal clock period while $N/2$ rings and 5 delay elements per ring are enabled. If PVT variations result in a slow oscillation before calibration, this counter slower than expected and suggests that fewer delay elements should be present in each ring in order to have a higher frequency range. However, selecting the size of the ring just by calibrating with the initial DCO output frequency turns out to require an unacceptable number of rings. Therefore, a second step by incorporating the desired output frequency is added and coarse frequency tuning is done together with the PVT calibration. During coarse frequency tuning, the crystal clock period is measured using the input refer-

ence clock signal as well. If the counter for input reference clock counts to indicate that a reference clock is in high side of the reference frequency range, the controller reduces the number of delay elements in a ring in order to allow for larger dynamic range during the coarse calibration. In summary, by comparing the counter values acquired from both measurements of the crystal clock, coarse frequency tuning and calibration are done concurrently by activating more or fewer rings and delay elements in the oscillator. This ensures that the best dynamic range is obtained for the coarse frequency tuning curve as shown in Figure 3.2.

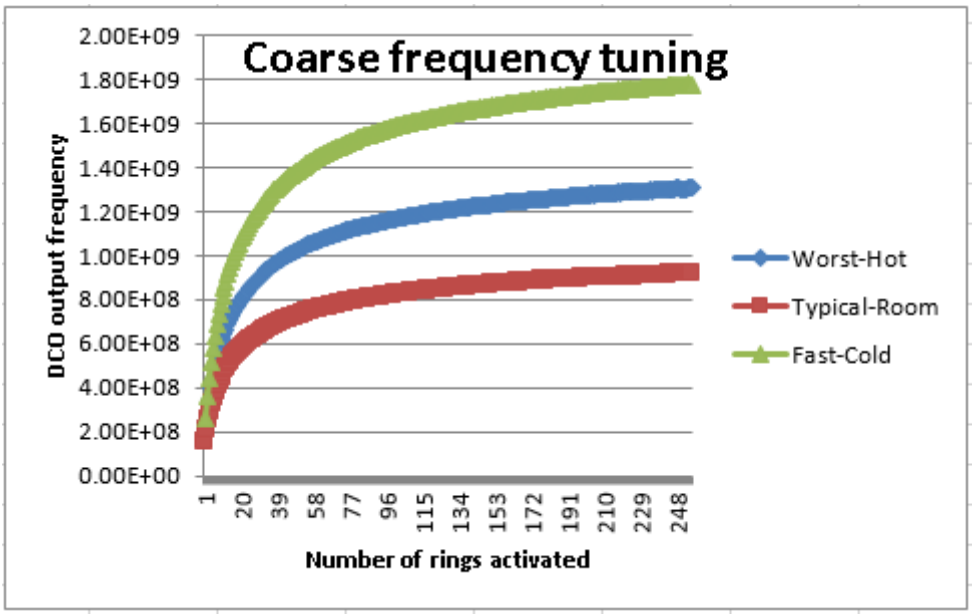


Figure 3.2. DCO coarse tuning curve example

In Figure 3.3, the number of delay elements activated in a ring is shown with respect to the initial untrimmed oscillation frequency and the desired output frequency. The adopted 2D calibration method allows better utilization of the number of rings activated during coarse frequency tuning and also compensates the non-linearity of coarse frequency tuning when very few rings (such as < 15) are activated. After 2D PVT calibration and coarse frequency tuning is completed, the number of delay stages activated in a ring and the number of rings activated (Figure 3.4) are fixed. Coarse frequency tuning ensures that the DCO is approximately around 30 MHz away from the desired output frequency. Figure 3.5 shows the result of PVT calibration and coarse tuning across the input frequency range and the initial PVT point.

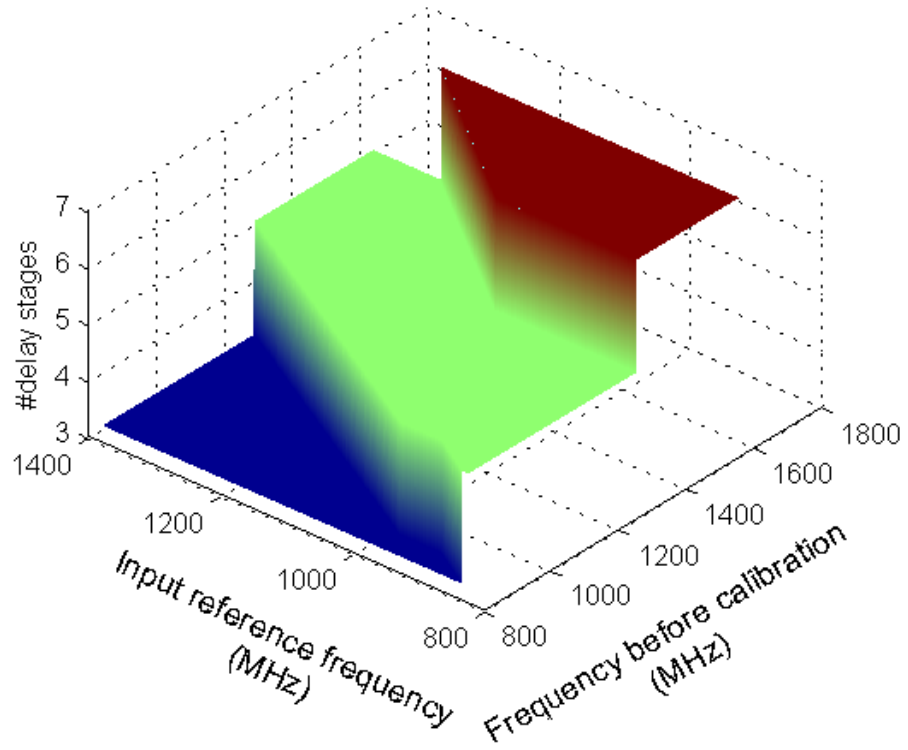


Figure 3.3. Number of delay stages activated in each ring for random initial oscillation point and 10 times input reference frequency

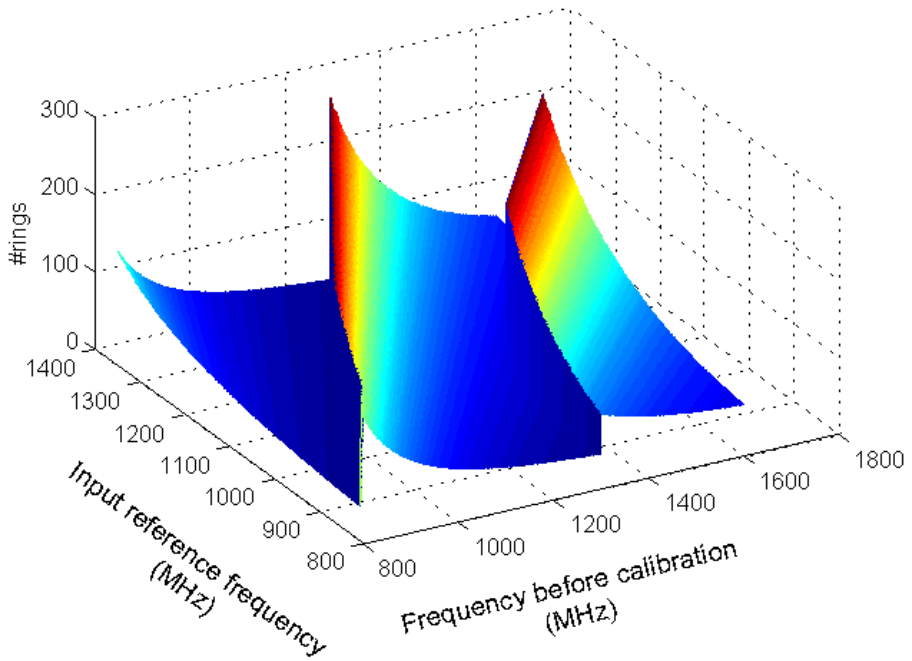


Figure 3.4. Number of rings activated in DCO

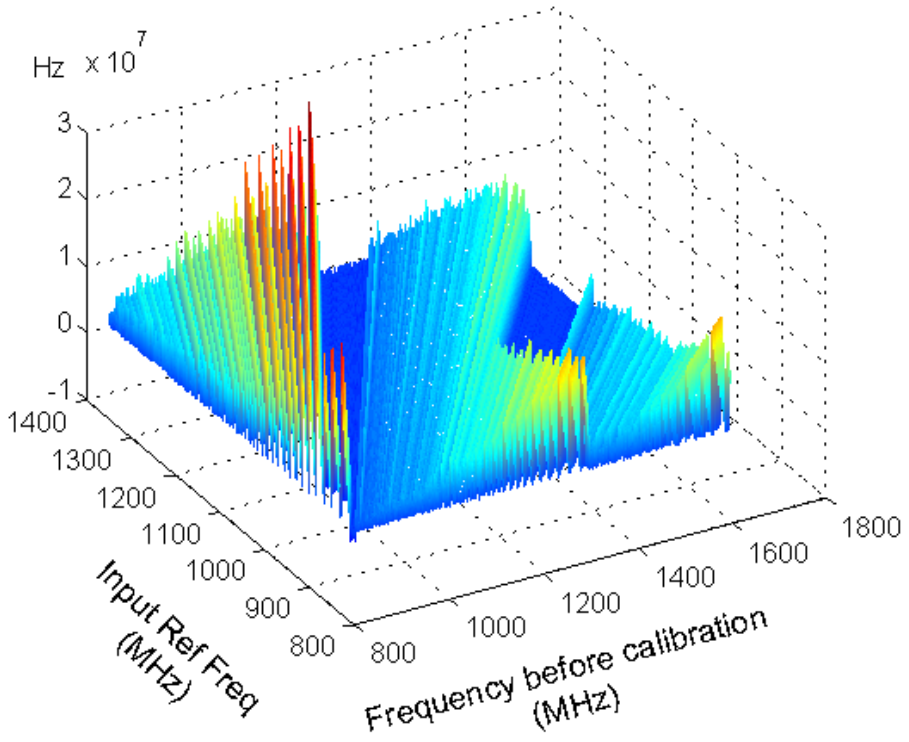


Figure 3.5. Frequency error after PVT calibration and coarse tuning

3.2.2. Fine frequency tuning method

The delay cells in Figure 3.1 utilize the single and multiple transitions at the inputs of the NAND gates and benefit from the intrinsic delay difference between input pin to output pin combinations. This results in 15 different but very close delay values generated from the input to the output of the cell depending on the selected code as shown in Figure 3.6. These small delay differences between code words translate into fine frequency tuning steps of approximately 1 MHz. Use of code 0000 is prohibited as it would block the oscillation. Every delay cell in a ring can be programmed by a different fine tuning code word, and by utilizing this method, a frequency code word to the fine tuning code words mapping has been implemented. For increasing DCO frequency code-words, fine tuning map starts from the slowest possible configuration for delay cells and decreases the chosen delay of the cell step by step before moving to the next delay cell. Loop dynamics ensure that this process goes on until the DCO

output period is in a TDC LSB step vicinity of the desired frequency. The frequency error signals required for the fine and coarse tuning algorithms are provided from the digital loop filter.

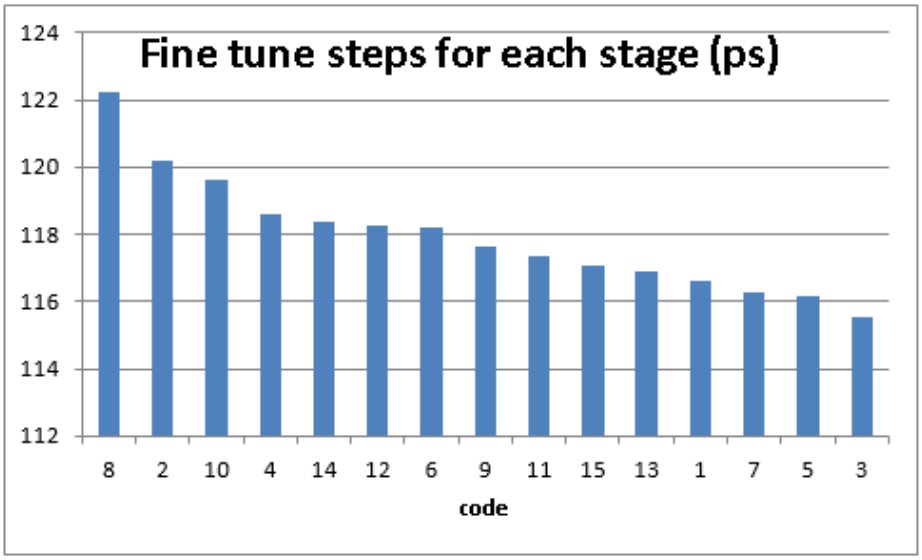


Figure 3.6. DCO fine frequency tuning example

Results of the phase and frequency locked DCO are depicted in Figure 3.7. As expected, it is observed that this DCO is able to approach the desired input frequency without dithering within a frequency error of approximately 1 MHz. This error translates to ±0.2% period jitter oscillation around the desired output without ΣΔ implementation. When a ΣΔ modulator is implemented to dither the frequency of the DCO, this error frequency can be pushed to higher frequencies.

3.2.3. Implementation results

Performance results derived from SPICE/MATLAB mixed simulations for the proposed DCO are presented in Table 3.1. For all possible initial operating points between fast-cold, slow-hot corners, all the possible frequency code words are simulated to validate frequency range coverage and steps. The proposed architecture can successfully switch in and out drivers, hence change the effective resistance in the RC time constant of each ring node while keeping the capacitor load constant. Implement-

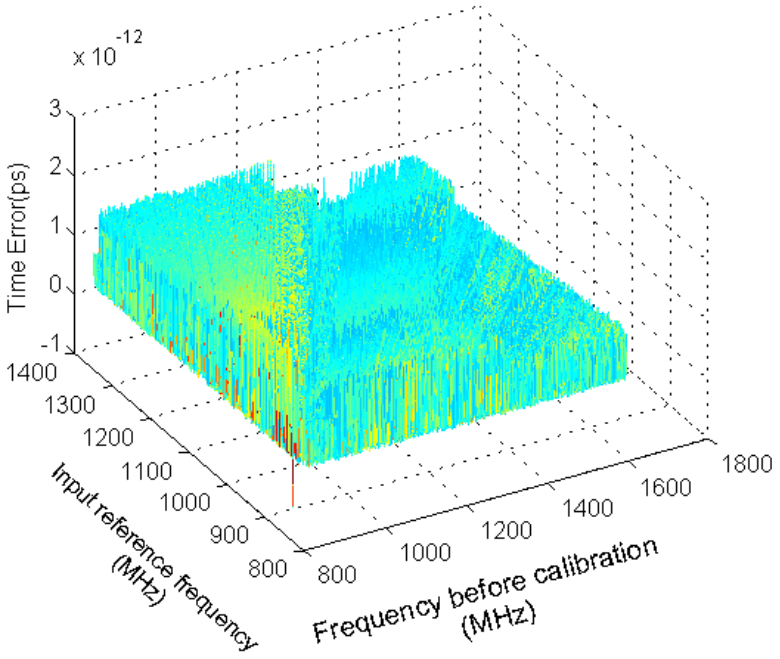


Figure 3.7. Time error after fine tuning

ing a novel calibration and tuning scheme and allowing synthesis only with standard cells, the DCO is implemented in all-digital flow and achieves the portability and flexibility goals while surpassing some recent work in this area either in resolution/range or area and power consumption while still staying in the ballpark for all performance parameters. While having acceptable phase noise, the period variation at the output of the foreseen ADPLL translates to an elevated phase noise floor. This open loop DCO phase noise floor is shaped by the NTF of the fractional ADPLL and the noise shaping characteristics of a $\Sigma\Delta$ modulator that is placed on the feedback path.

3.3. Phase noise modeling

The DCO has been modeled in the frequency domain as given in Figure 3.8a. Bit vector frequency control input F_{CTRL} goes through gain blocks with a gain of K_v [Hz/bits] and 2π. The resulting signal is the instantaneous frequency of oscillation in rad/sec which is subsequently integrated to get the continuous time phase of the output clock in radians. The DCO open loop noise is modeled using a random number

Table 3.1. DCO Performance compared to prior art

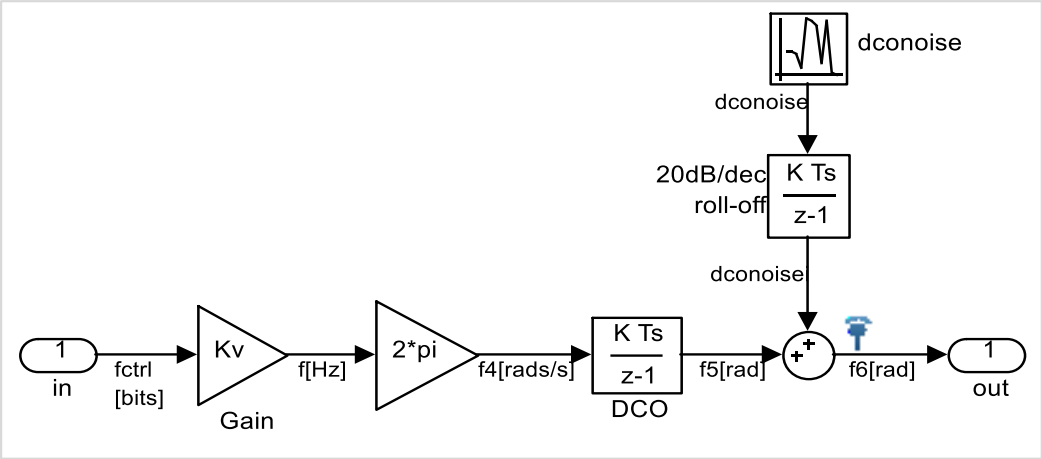
	This Work	[64]	[95]	[53]	[87]
Type	**Synthesized standard cells**	**Custom**	**Custom**	**Custom**	**Custom**
Voltage (V)	1	1.2	1.8	1.1	1.1
Power (mW)	20@0.85 GHz	33@2.6 GHz	7.2@446 Mhz	3.7@2 GHz	6.4@2.4 GHz
Process (nm)	65	65	180	65	40
Size (mm^2)	0.09	0.25	0.03	0.03	0.013
Resolution	1 MHz	1.8 MHz	1.6 MHz	0.25 MHz	0.004 MHz
Range (MHz)	810-1400	2600-4500	28-446	170-4270	2200-2400
Control Bit Width	9	10	8	14	14

generator (RNG) with a normal distribution to generate a white noise spectrum in the frequency domain. This instantaneous frequency noise is integrated as in Eq. 3.1 with the reference clock period T_s as the sampling time and the resulting phase noise seen in Figure 3.8b is added to the output phase of the DCO.

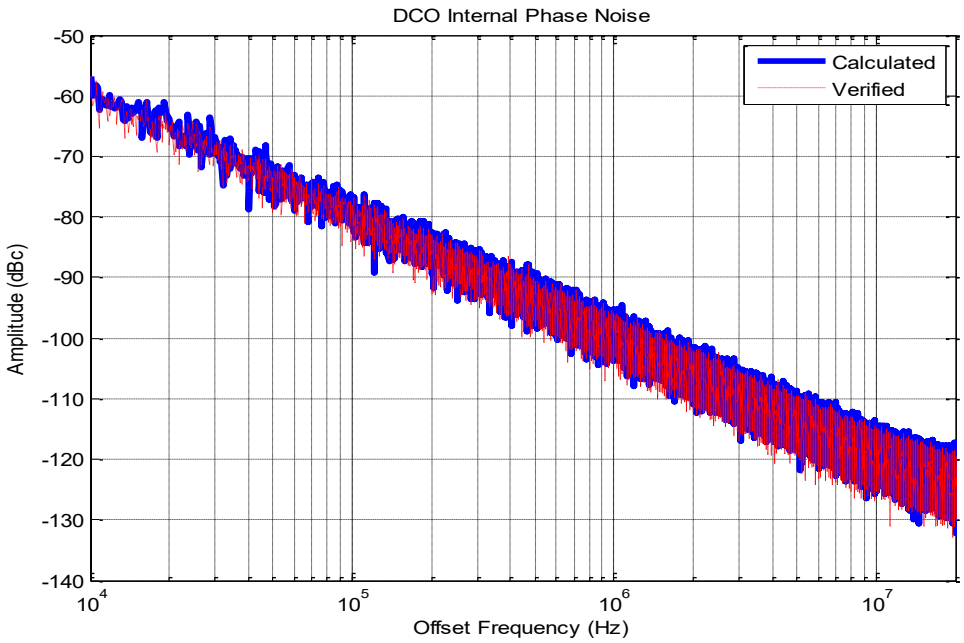
$$\Phi_{DCO_{out}} = DCO_{PN} + \sum 2\pi K_v T_s f_{ctrl_{in}}[k] \tag{3.1}$$

This noise power spectral density (PSD) rolls off with -20 dB/dec and its magnitude at zero offset frequency is determined by the variance of the RNG. With Eq. 3.2 provided in [96], the magnitude of the phase noise at the specified frequency by the user is calculated. F_{osc} is the center frequency where V_{DD} is the supply voltage, k is the Boltzmann constant, M is the number of delay stages, and C is the load capacitance.

$$\begin{aligned} f_o &= \frac{I}{CMV_{DD}} \\ L(f)_w &= \frac{2kT}{I}\left(\frac{\gamma_N + \gamma_P}{V_{DD} - V_t} + \frac{1}{V_{DD}}\right)\frac{f_o^2}{f^2} \end{aligned} \tag{3.2}$$



a.



b.

Figure 3.8. a. DCO phase model, b. DCO referred noise

In order to use this equation some parameters are extracted from the synthesis libraries and some are calculated. The DCO's internal phase noise parameters such as supply voltage V_{DD}, temperature T, threshold voltage V_t, and noise factor γ_N , γ_P are used from the library. However, the oscillation frequency f_{osc}, pull-up and pull-down current I, and the stage load capacitance value C which changes with the drive strength of the cells in the rings needs to be estimated. After HLS is run for the DCO, with the help of static timing analysis the strengths of the cells, load capacitance per

stage C, and the oscillation frequency are identified. The [96] assumes that the pull-up and pull-down currents are the same. With similar assumption, the pull current I is calculated by dividing the total charge $V_{DD}C$ by the average propagation delay t_p. Having all of the parameters of Eq. 3.2 allow the phase noise ($DCO_{PN}@f_{offset}$) at the desired offset frequency f_{offset} to be calculated. The desired maximum phase noise at the carrier frequency is projected using a linear interpolation with a slope of $-20dBc/dec$ starting from the result of Eq. 3.2 ($DCO_{PN}@f_{offset}$). Finally, the calculated phase noise at center frequency is used as the variance random noise source thereby completing the phase model for the DCO.

3.4. High-level synthesis

The standard cell Liberty library is parsed and the information extracted is stored in an internal database which is organized by cell type, strength, and pin-to-pin combinations. The extracted data includes propagation delay, power dissipation, capacitance, rise times, and fall times. Details on how the standard cell characterization library is parsed is given in Appendix B. As summarized in Figure 3.9, HLS starts by getting the feedback divider's maximum and minimum values together with the reference frequency range and calculates the desired DCO output frequency range $[N_{min} * f_{Ref_{min}}, N_{max} * f_{Ref_{max}}]$. The DCO template is deployed with minimal strength and an internal static timing analysis (STA) engine is used to calculate the oscillation range of the DCO. The ring oscillator in the DCO template uses gate instances in Verilog and the HLS algorithm adjusts the strengths of the cells in the rings in order to adjust the DCO tuning range and noise profile to satisfy the user specifications. The oscillation frequency is calculated for the worst PVT conditions, using the highest frequency tuning word, and optimal calibration to check if the highest desired frequency can be reached. Next, the opposite corner is verified at the fast process, high supply voltage, low temperature with the lowest tuning code word, and optimal calibration to guarantee operation at the minimum desired DCO output frequency. If the tuning range is not satisfied, the strengths of the cells are increased until the range is covered; otherwise, an exception is reported to the user denoting an impossible request .

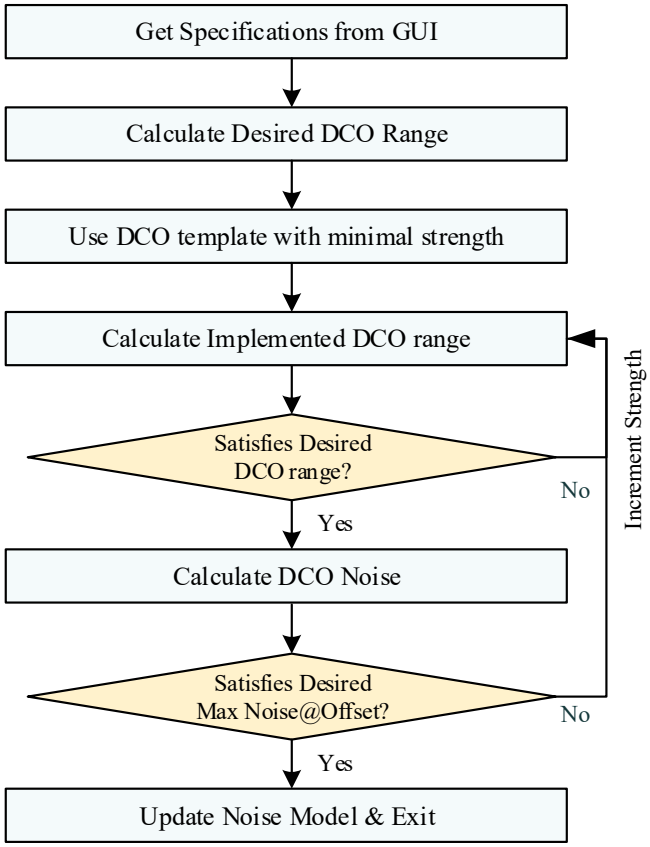


Figure 3.9. DCO high-level synthesis algorithm

STA is performed at each ring node using the 2D propagation delay and the transition time tables from the library. The propagation delay calculation in Eq. 3.3 uses 2D extrapolation between data points. The propagation delay and output transition time of every cell are calculated as a function of the input transition time and the total calculated capacitance at the output node of each cell. Then, the period of the oscillator core is calculated by accumulating the propagation delay through the cells. While running the STA, the loop is broken and an initial seed is given as the transition time for the first gate's input on the delay line. With this seed, all the nodes in the delay line are evaluated for their propagation and transition time. As the last node's transition time has to be equal to the input transition time of the first gate, an evolution loop is run by modifying the seed until the ring can be reconnected. The Verilog gate instances in the DCO's rings are marked as *"don't touch"* and the

controller section of the DCO is provided as RTL code for the RTL synthesizer.

$$T_{plh,phl} = T_{int} + F_{LUT}(C_{load}, T_{rf_{in}}) \tag{3.3}$$

With the calculated center of the DCO tuning range, the open loop phase noise for the DCO is calculated at the frequency offset that is set from the GUI as explained in Figure 3.3. Calculated noise at the specified frequency offset is compared to the maximum noise level set in the GUI. If the noise is not as low as desired, the DCO synthesis is restarted but with the remaining set of possible cell strengths in the new iteration. The loop repeats itself until the specifications are met or the synthesizer runs out of available strengths and returns an impossible realization request exception. The DCO HLS results are verified using the phase noise specification from CellPLL GUI as the upper bound. During the verification, an agreement between the calculated phase noise and the PSD generated from transient simulations is observed as shown in Figure 3.8b.

Chapter 3 presented the details of frequency generation block. Chapter 4 discusses the other fundamental block TDC that is used for phase detection in an ADPLL.

4. TIME-TO-DIGITAL CONVERTER (TDC)

Time-to-Digital Converter (TDC) is one of the main blocks in an ADPLL. It measures the time from the reference clock edge to the feedback clock edge and gives a digital output as shown in Figure 4.1. Phase detection is implemented by the TDC block which produces an error signal for minimization by the loop. The TDC works similarly to an analog-to-digital converter, but it converts a time duration instead of an amplitude to a digital representation.

The digital phase error is used by the ADPLL in order to adjust the frequency and the phase of the output clock such that it is $N+F$ times the frequency of the input reference clock where N is the integer and F is the fractional part of a fractional clock multiplication value. Compared to the high frequency wireless mobility products that push for absolute performance such as $<$ 1 ps/LSB TDC resolution, compromises in performance specifications can be acceptable in wireline communication applications. As the cable lengths in the vehicles are short and shielding is quite strong, requirements for phase noise and spur performance can be relaxed which allows the TDC resolution on the order of 1 ps to 10 ps/LSB. There is previous work related to synthesizable TDCs in [6,97] but the proposed TDC in this study is novel in the sense that it implements an improved digital signal processing scheme to decrease the effective TDC resolution and is implemented in all-digital design flow compatible with synthesis, Auto Place and Route (APR) and integrated circuit (IC) fabrication using only standard library cells. The proposed architecture uses Verilog RTL coding in general with a gate level Verilog section accounting for the ring oscillators. Previous work in the literature contain papers claiming all-digital operation. However, [36, 98] contain custom gates and use methods that introduce extra analog behavior in addition to ring oscillators within

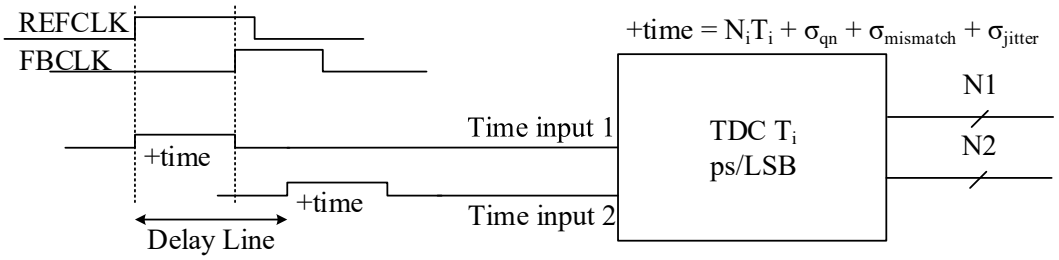


Figure 4.1. Multiple input single output (MISO) 2×1 TDC

their designs. [76, 99] are digital only at the interface and [77] is not synthesizable.

4.1. Proposed method

In this section, a novel quantization noise suppression method is presented. First, the background information about prior method is given in Subsection 4.1.1 and the proposed MIMO quantization noise suppression method is analyzed in Subsection 4.1.2. As shown in Figure 4.2, reference clock, feedback clock, and the delayed clone of the time input are processed in multiple parallel TDCs and the results are combined in order to get superior TDC resolution with this new method by reducing the sampling jitter component of quantization noise.

4.1.1. Single input multiple output (SIMO) quantization noise suppression

In order to get better effective resolution in TDCs, a quantization noise suppression method has been initially presented in [97]. The technique requires digitization of time input by multiple independent observers. Parallel TDC paths with unique conversion resolutions are utilized in order to achieve effective measurement accuracy better than each individual observer. As each TDC has a unique resolution, different quantization noise profiles and independent observation results are obtained. Analogous to multiple receiver antennas in a single input multiple output (SIMO) phased array antenna grid, parallel TDCs can provide receiver diversity. This diversity is

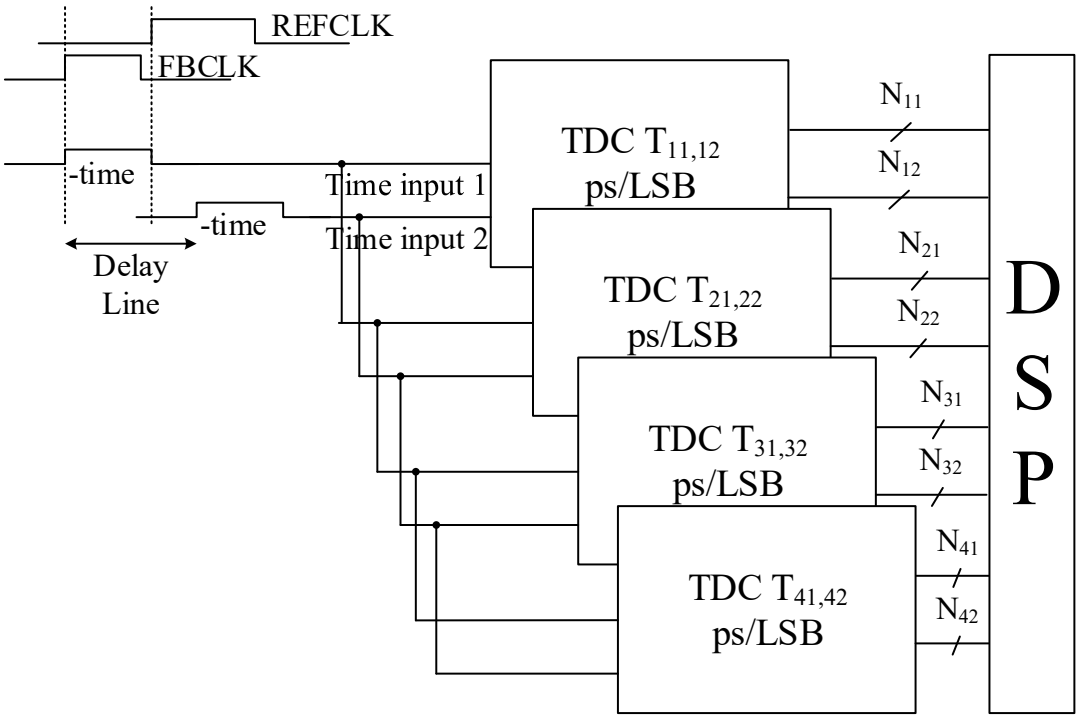


Figure 4.2. Multiple input multiple output (MIMO) 2×4 TDC

provided by the principle of superposition and the fact that one can benefit from the result of covariance (COV) of the correlated and uncorrelated signals. Effective TDC resolution is improved using the equal gain combining method; coherent absolute time measurements for summation are created by multiplying the output of TDCs back with their individual estimated resolutions to create quantized versions of the time input. Finally, these products are averaged to achieve superposition. Rest of the section explains how the RMS quantization noise standard deviation σ is suppressed by an order of $\sqrt{N}$ via digital post processing compared to the signal level.

SIMO case is observed when only the "time input 1" in Figure 4.3 is processed. The equivalent baseband model is as follows. The individual branch signals are

$$Err_{i1} = \frac{S_1}{T_{i1}} + n_{i1} \tag{4.1}$$

Where S_1 is the signal "time input 1", T_{i1} is the TDC channel gain, n_{i1} is the uniformly distributed quantization noise with σ_{i1}^2. The output of the combiner is

$$Err_{out} \;=\; \sum_{i=1}^{N} T_{i1} Err_{i1} \;=\; S_1 \sum_i \frac{T'_{i1}}{T_{i1}} + \sum_i T'_{i1} n_{i1} \tag{4.2}$$

Where T'_{i1} are the combining weights that try to estimate T_{i1} with details explained in Subsection 4.1.3. In Eq. 4.2 signal and noise components are given by 1^{st} and 2^{nd} term correspondingly. The signal and noise power at output are:

$$P_{s_1} = \overline{\left| S_1 \sum_i \frac{T'_{i1}}{T_{i1}} \right|^2} \;=\; \frac{1}{2}|S_1|^2 \left| \sum_i \frac{T'_{i1}}{T_{i1}} \right|^2 \tag{4.3}$$

$$P_{n_1} \;=\; \left| \sum_i T'_{i1} n_{i1} \right|^2 \;=\; \sum_i \left| T'_{i1} \right|^2 s_{i1}^2 \tag{4.4}$$

where $s_{i1}^2 \;=\; \overline{|n_{i1}|}^2 \;=\; \frac{{T_{i1}}^2}{12}$ is the branch noise power. Output SNR is:

$$SNR_{out} \;=\; \frac{P_{S_1}}{P_{n_1}} \;=\; \frac{1}{2}|S_1|^2 \frac{\left| \sum_i \frac{T'_{i1}}{T_{i1}} \right|^2}{\sum_i \left| T'_{i1} \right|^2 s_{i1}^2} \tag{4.5}$$

The range of index i is N and a comparison of SNR at $N = 1$ and $N = 4$ for unique but close $T_{i1} = \{T_{11}, T_{21}, T_{31}, T_{41}\}$ with an average of T_{avg} shows:

$$\frac{SNR_{out} \mid_{N=4,\;\; T_{i1} = \{T_{11}, T_{21}, T_{31}, T_{41}\}}}{SNR_{out} \mid_{N=1,\;\; T_{i1} \;=\; \{T_{11}\}}} \;\tilde{=}\; 4 \tilde{=} \frac{SNR_{out} \mid_{N=1,\;\; T_{i1} = \{T_{avg}/\sqrt{4}\}}}{SNR_{out} \mid_{N=1,\;\; T_{i1} \;=\; \{T_{avg}\}}} \tag{4.6}$$

which indicates that the employed weighted gain combining method provides quantization noise suppression and allow the 1 × N TDC act as if it was a TDC with single channel and resolution of $\frac{T_{avg}}{\sqrt{N}}$.

4.1.2. Proposed MIMO quantization noise suppression method

Using the same number of TDCs, MIMO quantization noise suppression method achieves improved resolution compared to the SIMO configuration. A transmitter diversity similar to antenna arrays with multiple transmitters is obtained by creating a delayed clone of the time input and refeeding it to the locked loop's TDCs for reconversion with another resolution setting. In order to have an independent second observation from the same channel, TDC resolution is changed after the first measurement hence the same time input's delayed clone is observed with a different quantization noise. Transmitter diversity for the same receiver is obtained as the system acts as if there is a second time input source feeding through a different sampling mechanism. To be able to use the same TDC for the time input and its delayed clone, these pulses need to be non-overlapping, which can only be achieved in the locked state of the PLL.

MIMO case is observed when "time input 1 and 2" in Figure 4.3 is processed. The equivalent baseband model is as follows. The individual branch signals are

$$Err_{ij} \;=\; \frac{S_j}{T_{ij}} \;+\; n_{ij} \tag{4.7}$$

Where S_j is the signal "time input j", T_{ij} is the TDC channel gain, n_{ij} is the uniformly distributed quantization noise with σ_{ij}^2. The output of the combiner is

$$Err_{out} \;=\; \sum_{i=1}^{N}\sum_{j=1}^{M} T_{ij}Err_{ij} \;=\; S_j \sum_i \sum_j \frac{T'_{ij}}{T_{ij}} + \sum_i \sum_j T'_{ij} n_{ij} \tag{4.8}$$

Where T'_{ij} are the combining weights that try to estimate T_{ij} with details explained in Subsection 4.1.3. In Eq. 4.8 signal and noise components are given by 1^{st} and 2^{nd} term correspondingly. As signals $S_1$and S_2 are ideally the same, the signal and noise power at output are:

$$P_s = \overline{\left| S \sum_i \sum_j \frac{T'_{ij}}{T_{ij}} \right|^2} \;=\; \frac{1}{2}|S|^2 \left| \sum_i \sum_j \frac{T'_{ij}}{T_{ij}} \right|^2 \tag{4.9}$$

$$P_n \;=\; \left|\sum_i \sum_j T'_{ij} n_{ij}\right|^2 \;=\; \sum_i \sum_j \left|T'_{ij}\right|^2 \sigma_{ij}^2 \tag{4.10}$$

where $s_{ij}^2 \;=\; \overline{|n_{ij}|}^2 \;=\; \frac{{T_{ij}}^2}{12}$ is the branch noise power. Output SNR is:

$$SNR_{out} \;=\; \frac{P_S}{P_n} \;=\; \frac{1}{2}|S|^2 \frac{\left|\sum_i \sum_j \frac{T'_{ij}}{T_{ij}}\right|^2}{\sum_i \sum_j \left|T'_{ij}\right|^2 s_{ij}^2} \tag{4.11}$$

The range of index (j, i) is (M, N) and a comparison of SNR at $(M, N) = (1, 1)$ and $(2, 4)$ for unique but close $T_{ij} = \{T_{11}, T_{21}, T_{31}, T_{41}, T_{12}, T_{22}, T_{32}, T_{42}\}$ with an average of T_{avg} shows:

$$\frac{SN_{out} \mid_{N=4,\;\; M=2,\;\; T_{ij}=\;\{T_{11},T_{21},T_{31},T_{41},T_{12},T_{22},T_{32},T_{42}\}}}{SN_{out} \mid_{N=1,\;M=1,\;T_{ij}\;=\;\{T_{11}\}}} \;\simeq\; \frac{SN_{out} \mid_{N=1,\;M=1,\;T_{ij}=\;\{T_{avg}/\sqrt{8}\}}}{SN_{out} \mid_{N=1,\;M=1,\;T_{ij}\;=\;\{T_{avg}\}}} \tag{4.12}$$

Eq. 4.12 shows that utilizing four parallel 2 × 1 TDCs to create a 2 × 4 TDC as proposed in Figure 4.3 suppresses quantization noise as much as a 1 × 8 SIMO TDC but with half of the number of TDC channels used in SIMO configuration. That is, the 2 × N MIMO configuration acts as if there is a single TDC with a resolution of $\frac{T_{avg}}{\sqrt{2N}}$, which is improvement of $\sqrt{2}$ over the SIMO case.

4.1.3. Online TDC resolution estimation

Completion of digital post processing requires the online estimation of the TDC resolutions. In the targeted 2×4 MIMO TDC application, resolutions within 1 ps of each other need to be distinguished. The method presented in [97] is used for the required online estimation. A priori known output sequence of the ΣΔ modulator creates a known phase error at the input and the output of the TDC, which allows the estimation of TDC resolutions. Starting from the typical resolution values and digitally filtering each estimation sample, stable resolution estimation is obtained.

Feedback divider divides DCO output clock with a desired fractional value of NF and generate feedback clock for comparison to reference clock. This comparison gives the digital phase error ($Err[k]$) which allows the loop to adjust the phase and the frequency of the DCO clock to achieve the desired clock frequency multiplication value NF. In other words, NF is a user specified value that determines the desired DCO output frequency. $Err[k]$ is filtered by the digital loop filter.

By exercising the TDC inputs with an a-priori known input sequence $S_i[k]$ and comparing the conversion result $Err[k]$, the resolution T_i can be estimated. This can be done in the fractional-N PLL setup used throughout this study. In $\Sigma\Delta$ fractional-N PLLs, the correlation between the TDC's input time error $s_i[k]$ and the immediate $\Sigma\Delta$ output $NF[k]$ is defined by Eq. 4.13.

$$s_i[k] = \frac{\sum_k N[k] - NF}{F_{ref} * NF} \tag{4.13}$$

with reference frequency F_{ref}, and fractional divider value $N + F$. The signal at TDC input in Eq. 4.13 can be calculated on the fly as all the parameters are known at every time step. This result can be used to compute the running TDC gains $T_i[k]$.

Use of this method results in no overhead as a divider and $\Sigma\Delta$ are already implemented in a fractional-N PLL. Using the defined TDC input time and TDC output conversion statistics, instantaneous T_i can be calculated as given in Eq. 4.14

$$T_i[k] = \frac{N[k] - NF}{F_{ref} * NF * (Err[k] - Err[k-1])} \tag{4.14}$$

4.2. Architecture of proposed TDC

In a basic TDC architecture, during the error time window between the rising edges of REF_{clk} and FB_{clk}, the ring oscillator inside the TDC is enabled using $en1$ and $en2$ signals. When the ring oscillator is running, the outputs of the delay cells in the ring are used as the clock signal for increment counters (ACC). At the end of the

error window, results of the counters from each node of the ring are summed to get the final result as given in Eq. 4.15.

$$Err_{ij}[k] = \sum_{m=1}^{M} ACC_m[k]$$
$$T_{err_{ij}}[k] = Err_{ij}[k] * T_{res_{ij}} \tag{4.15}$$

The delay element in the ring determines the conversion resolution T_{res} and sets the level of quantization noise generated during the conversion. In [97], an enhanced TDC architecture that can suppress quantization noise by $\sqrt{N}$ is presented. The enhancement is obtained by using N parallel TDC channels with unique conversion resolutions and averaging the results in order to benefit from the signal correlation between channels. CellPLL's ADPLL template uses the TDC in Figure 4.3 which suppresses the quantization noise further up to $\sqrt{2N}$ with its re-conversion technique [11] as given in Eq. 4.16 when $N = 4$.

$$T_{err}[k] = Err[k] * T_{res_{eff}} = \frac{1}{8} \sum_{i=1}^{4} \sum_{j=1}^{2} Err[k]_{ij} * T_{res_{ij}} \tag{4.16}$$

Proposed design is composed of a phase detector, delay line, two gear ring oscillator with counters and digital post processing. Delayed clone of the input up-down pulse is multiplexed to the phase detector during the silent phase after the falling edge of the time input. Phase detector works from rising to rising edge of reference clock (REF_{clk}) or feedback clock (FB_{clk}) signals and create a positive output if REF_{clk} is leading the FB_{clk}. Delay cells are used to delay the phase detector output for use in the 2^{nd} conversion. It should be noted that this delay does not need to be equal to a specific value and that the delays in each channel do not need to match each other.

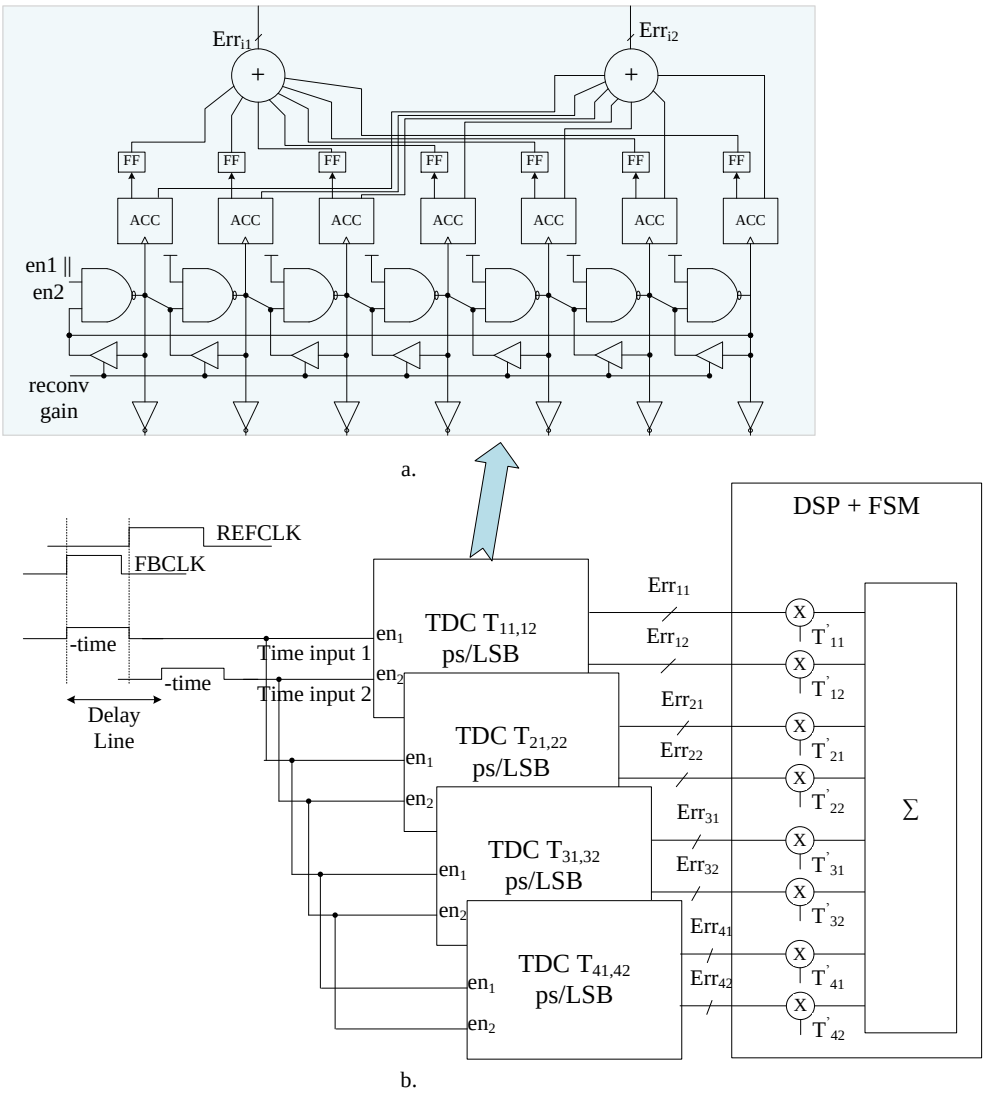


Figure 4.3. a. Architecture of a channel, b. TDC with parallel channels

Design just needs to make sure that the clone signal overlaps with the idle window of the phase detector. Silent window for reconversion is minimum when feedback divider is minimum and the $\Sigma\Delta$ modulator disposition is maximally negative. In order to ensure that a minimum length idle window is available in the phase detector reference clock period needs to be constrained. For a constrained reference clock frequency of maximum 100 MHz, this window corresponds to a minimum silent window starting from 4 ns to 8 ns after the rising edge of the up/down signal. While the exact delay for the time input clone is flexible, it has to be in this range in all PVT corners so that the 1st conversion is non-overlapping with the 2nd conversion. In the fast-cold

corner the delay should be greater than 4 ns while the slow-hot corner delay should be smaller than 8 ns.

4.2.1. Two Gear Ring Oscillator with Counters

A seven stage NAND gate ring oscillator is implemented with an enable input in one of the stages. Number of stages in a ring is chosen to be seven in order to keep the counter widths at each node small while having enough dynamic range to cover the reference clock range with the minimum TDC resolution. Utilized quantization noise method requires all simultaneous conversions to be done with a unique resolution. In order to equip each TDC with a unique resolution, oscillation frequency is adjusted for each TDC by incorporating dangling inverters at each node of the ring. Sizes of the inverters are configured for TDC replicas in order to provide the desired frequency offset. In order to select a slightly different resolution during the 2nd conversion using the same TDC, standard cell tri-state buffers are connected between each NAND gate output and input. When enabled, these tri-state buffers decrease the period of oscillation and increase single channel TDC resolution. Oscillation is enabled only during the time pulses and their delayed clones. There are eight bit wide asynchronous counters at the output of each ring stage and these counters are summed in order to get the N_{i1} and N_{i2} outputs. First conversion output is latched into FF with the falling edge of the time input 1 and the same hardware is used for the second conversion. In order to have a dynamic range spanning specified reference clock range with the given TDC resolution, 1^{st} and 2^{nd} conversion outputs are provided in eleven bit two's complement format to the post processor. The tri-state buffer strengths and dangling inverter sizes are fine-tuned by the HLS algorithm to get typical TDC resolutions given in Table 4.2.

4.2.2. Digital Post Processing

Both outputs of 2×1 TDCs are multiplied with their corresponding five bit wide estimated resolution (T_{res_i}) and these products are averaged as shown Figure 4.3b. While both outputs of each parallel TDC are used during the MIMO operation, second

output is omitted for post processing in the SIMO mode. The result is a twenty bit wide output for use in the loop filter. Quantization noise has components due to mismatch, jitter and sampling error. The saw-tooth shaped sampling error of the quantization noise for a single TDC channel is simulated as shown in Figure 4.4. The combined quantization noise due to all off the TDC channels in time domain is simulated as shown in Figure 4.5. In order to present that the sampling error suppression is obtained, TDC has been simulated in transient simulation and the phase detection results have been compared to the actual input signal to generate histograms that converge to the resulting quantization noise profile. While the ADPLL is locking, SIMO operation provides an effective resolution of $T_{res_{eff}} * \sqrt{2}$ ps/LSB (Figure 4.6b), which is improved to $T_{res_{eff}}$ ps/LSB (Figure 4.6a) when MIMO mode is enabled after loop is locked. When Figure 4.6a and Figure 4.6c are compared, it is observed that the 2×4 MIMO configuration has the same amount of quantization noise suppression capability as the 1×8 SIMO configuration given in Figure 4.6b.

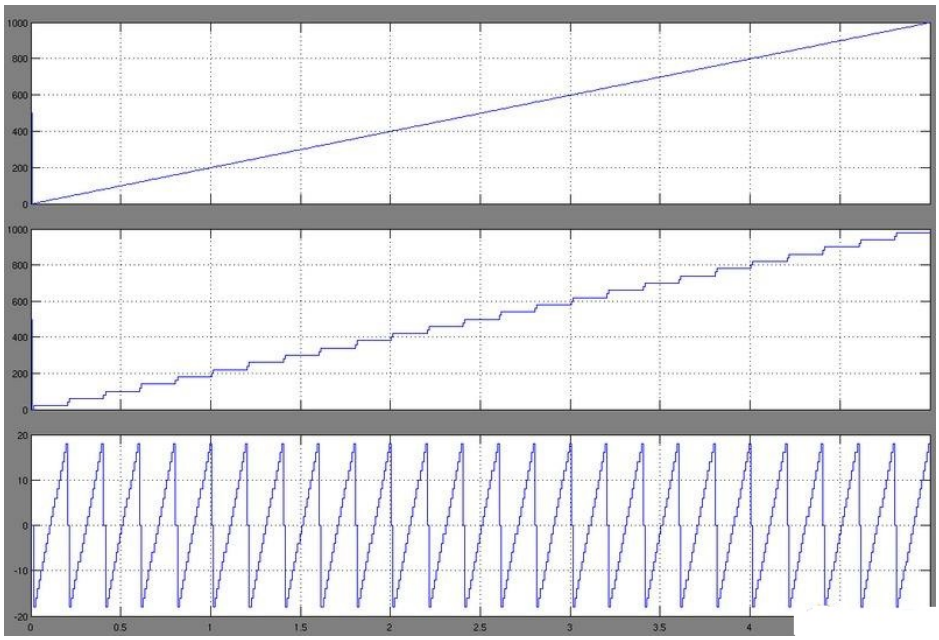


Figure 4.4. Sampled output time and sampling error for a 2×1 configuration

4.2.3. Implementation results

Performance results derived from SPICE/Verilog mixed simulations for the proposed TDC are presented in Table 4.1. The design achieves the same theoretical

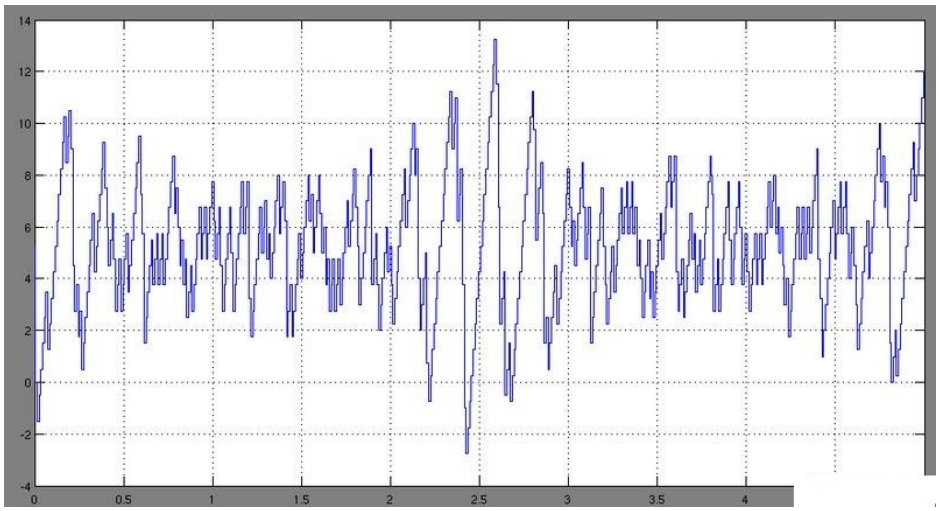


Figure 4.5. Combined sampling error of the 2×4 TDC configuration

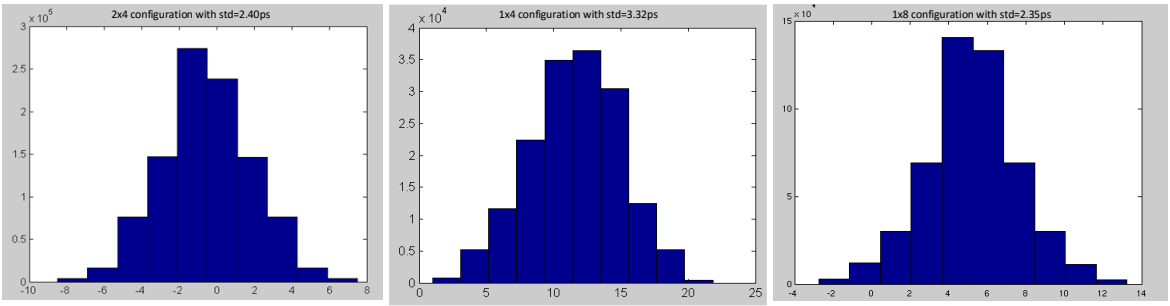


Figure 4.6. Comparison of a. 2×4, b. 1×4, c. 1×8 TDC quantization noise histograms when $T_{res_{eff}} = 7\ ps$

resolution and noise performance of [97, 100] with half the number of TDC instances used, hence half the number of gates compared to [97] but still none of the digital implementations can achieve the sub-gate delay performance of analog counterpart given in [101]. Comparison to [97] is done according to the simulation results given in [97] rather than the measured result as the results of this thesis are obtained from simulations. Area and power consumption of the proposed design is superior to all in comparison even after technology scaling is applied especially compared to the analog implementation given in [101].

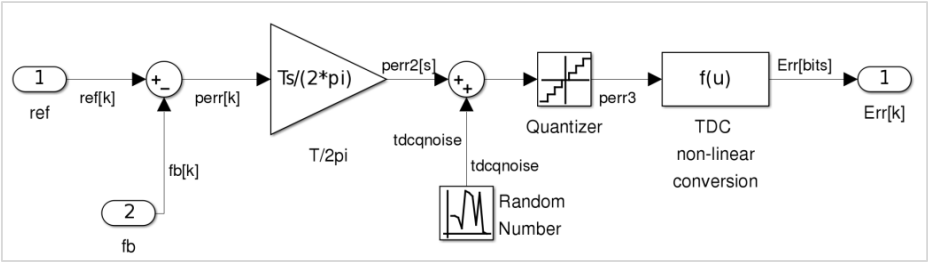
Table 4.1. TDC Performance compared to prior art

	This Thesis	[97]	[100]	[101]	[102]
Type	**Synthesized standard cells**	**Custom standard cells**	**Analog Mixed**	**Analog**	**Synthesized standard cells**
Voltage (V)	1	1.2	1	1.2	1.2
Power (mW)	3.9	9.12	10	70	0.36
Process (nm)	65	90	65	90	65
Size (mm^2)	0.02	0.26	0.4	2.2	0.02
Resolution (ps/LSB)	7	$14/8^1$	8	2.6	9

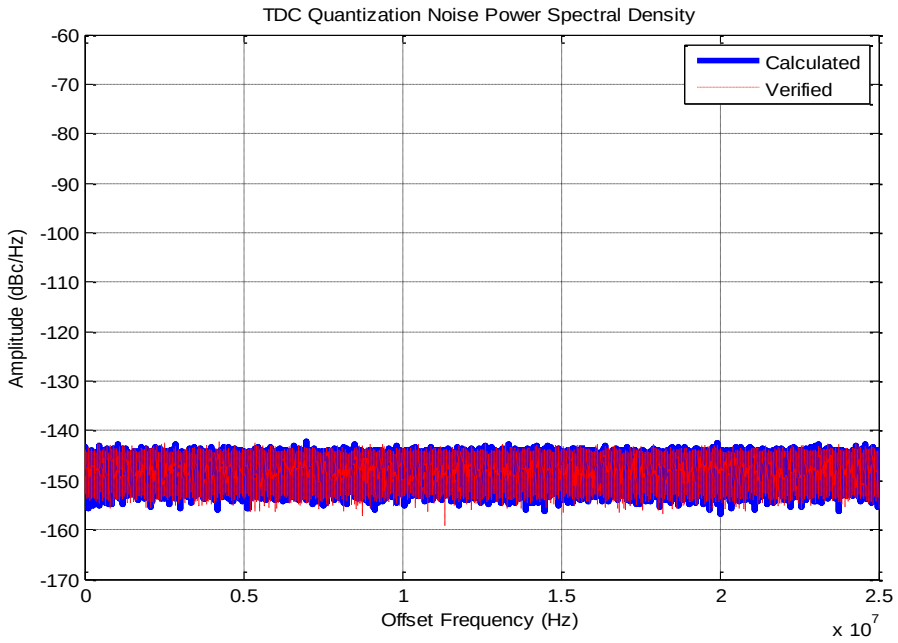
4.3. Phase noise modeling

The phase domain model for the TDC is given in Figure 4.7. The feedback phase input is subtracted from the reference phase input in the discrete domain. The resulting error signal goes through a gain block $T_s/(2\pi)$ to convert the discrete time error signal $perr[k]$ to a phase error $perr[s]$ in continuous time. During the time-to-digital conversion, a quantization error uniformly distributed between $[-T_{res_{eff}}/2, T_{res_{eff}}/2]$ is introduced. This error is known to have a bounded white noise PSD with a variance $\sigma^2 = {T_{res_{eff}}}^2/12$. In order to get the white noise PSD, a quantization error is modeled as an addition to $perr[s]$ using a normally distributed RNG with a variance $\sigma^2/T_{res_{eff}}$. Scaling with $1/T_{res_{eff}}$ is added in order to account for the PSD translation from discrete time input domain to continuous time output domain of complete ADPLL phase model [2]. Finally, the resulting phase signal goes through a non-linear quantizer with an ideal step size of $T_{res_{eff}}$ and the result of quantization generates the output digital error vector $Terr[k]$. When the HLS is ran, the model is updated by replacing the desired TDC resolution with what was actually implemented $T_{res_{eff}}$.

TDC non-linearity is another important design parameter that results in elevated close in phase noise and fractional spurs. In order to account for TDC non-linearity, the TDC model incorporates a programmable non-linearity parameter which allows the user to investigate the effects of this non-linearity on the system performance. Figure



a.



b.

Figure 4.7. a. TDC Phase domain model, b. Quantization noise profile ($T_{res_{eff}} = 20ps$)

4.8 illustrates how the non-linearity is modeled during the time to digital conversion and presents the implemented non-linearity equation. In Figure 5.6, the results with various non-linearity amounts are discussed.

4.4. High-level synthesis

ADPLL template in CellPLL has an embedded TDC that has ($N = 4$) ring oscillators implemented with gate instances in Verilog code and an RTL portion that contains the rest of the block design. An HLS algorithm for the TDC is given in Figure 4.9. In HLS, only the ring oscillator frequencies need to be synthesized as the rest of the design is synthesized by the RTL synthesizer.

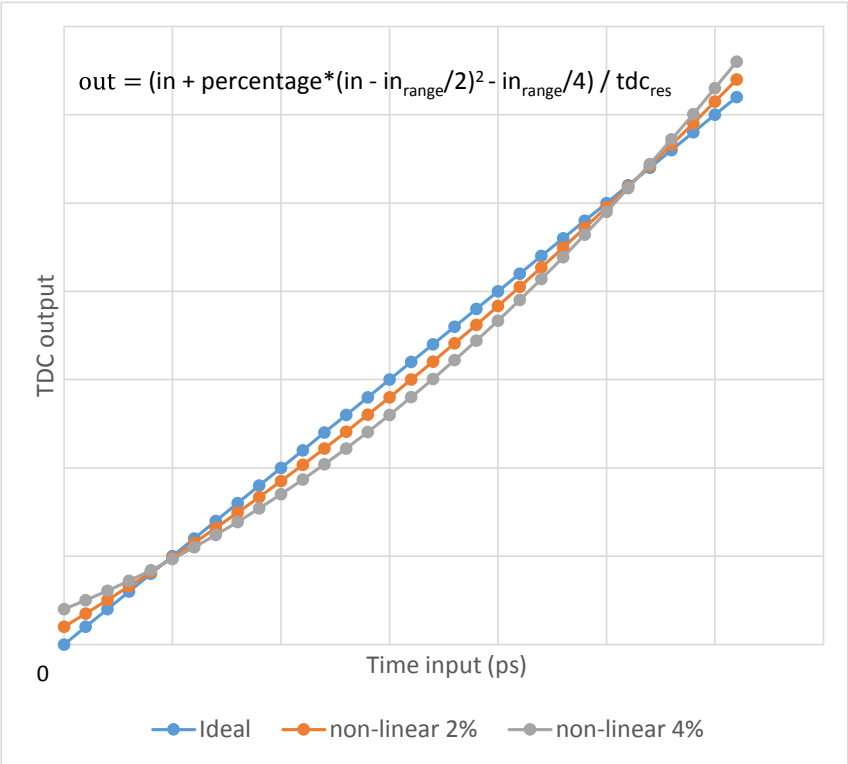


Figure 4.8. TDC non-linearity model

The ring oscillators in the design need to have unique T_{res_i} in order to enable the quantization noise suppression algorithm that is implemented by the DSP unit. These unique resolutions do not have to be set to exact values; they just need to be different from each other but must also be in the same ballpark. $T_{res_{eff}}$ is set from the GUI and $T_{res_{eff}} * \sqrt{8}$ is set as the upper bound of the range for the TDC channel resolutions. NAND gates in the rings are assigned target propagation delays as given in Table 4.2 for the first, second, third, and fourth rings. When the target values are met, this specific TDC implementation gives an effective TDC resolution of $T_{res_{eff}}$.

The HLS algorithm of the TDC is the similar to the one of the DCO for the highest desired ring oscillation frequency. It increases the strengths of the NAND gates of the ring until the desired stage delay that will satisfy the finest individual TDC channel resolution T_{res_i} is obtained. This step sets the strengths of all the ring oscillators in the design. Next, the engine starts adding dangling inverters to each node of the ring oscillators as capacitive load in order to increase the delay per stage

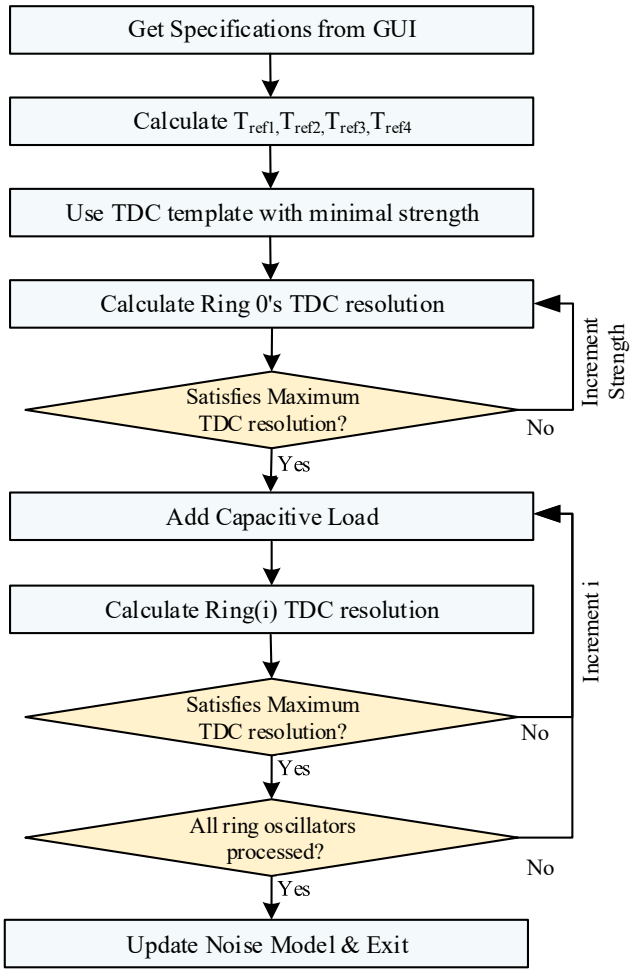


Figure 4.9. TDC high-level synthesis algorithm

Table 4.2. Implemented HLS targets for TDC channel resolutions

TDC channel resolution	**HLS target value [ps]**
$\mathbf{T_{res_1}}$	$T_{res_{eff}} * \sqrt{2N}$
$\mathbf{T_{res_2}}$	$T_{res_{eff}} * \sqrt{2N} - 2$
$\mathbf{T_{res_3}}$	$T_{res_{eff}} * \sqrt{2N} - 4$
$\mathbf{T_{res_4}}$	$T_{res_{eff}} * \sqrt{2N} - 6$

of the $2^{nd}, 3^{rd}$, and 4^{th} rings until T_{res_i} converges to the desired value. The rings are marked for *"don't touch"* in synthesis scripts and TDC phase model is updated with

the actual TDC resolution thereby completing the HLS for TDC. The HLS run for the TDC is verified by generating a TDC with $T_{res_{eff}} = 20\ ps$ and matching the modeled phase noise profile to the PSD produced from transient simulation as illustrated in Figure 4.7b.

As the HLS algorithm's main functionality is to calculate propagation delays through digital gates, the proposed HLS method can be generalized to work with standard cell based ring oscillator architectures. For any gate-level Verilog code that needs cell strength optimization based on propagation delays, this static timing analysis approach can be used to adjust the cell sizes before the RTL portion of the code is synthesized.

Chapter 4 presented the details of the last remaining fundamental block TDC. Using the two main sub-blocks DCO and TDC, Chapter 5 discusses the top level ADPLL design and automatic loop generation.

5. ALL-DIGITAL PHASE LOCKED LOOP (ADPLL)

CellPLL uses the previously defined DCO and TDC together with a digital loop filter, a ΣΔ modulator, and a multi-modulus feedback divider shown in Figure 1.1 to generate top-level of the ADPLL. To start with, the architecture of the remaining sub-blocks in the loop are presented and the phase model of the system is completed. Next, transfer functions and loop parameters are generated and analyzed according to user specifications. Finally, noise transfer functions (NTF), phase noise estimation, and high-level synthesis of the ADPLL top-level are explained.

5.1. Architecture

The ADPLL top-level and loop filter implementation are shown in Figure 5.1. The structure digitally imitates integral and proportional paths of a type-2 order-2 analog system response with the help of accumulation, digital scaling, and IIR filtering operations in the loop filter. The transfer function of this loop filter circuit is suitable for realizing the calculated digital filter response presented in Section 5.2. Real valued filter coefficients of the transfer function are approximated with scaling in order to allow synthesis with integers. The phase error $Err[k]$ is fed to the filter. The signal goes through integral and proportional paths which use IIR filtering, multiplication, and accumulation with parameters such as K_1, K_2, and α. The result from the loop filter is scaled down to generate the DCO frequency control word at its output.

In order to generate a fractional-N architecture, the ΣΔ modulator shown in Figure 5.2b is used. In order to suppress the frequency spurs generated in the fractional mode, the method in [103] is used. Additionally, this modulator is used to generate a known pattern while estimating the resolution of the TDC channels during operation as explained in [11].

By connecting first order digital ΣΔ blocks given in Figure 5.2b, a 2^{nd} order MASH11 digital ΣΔ topology similar to the one in [104] is implemented. These eight

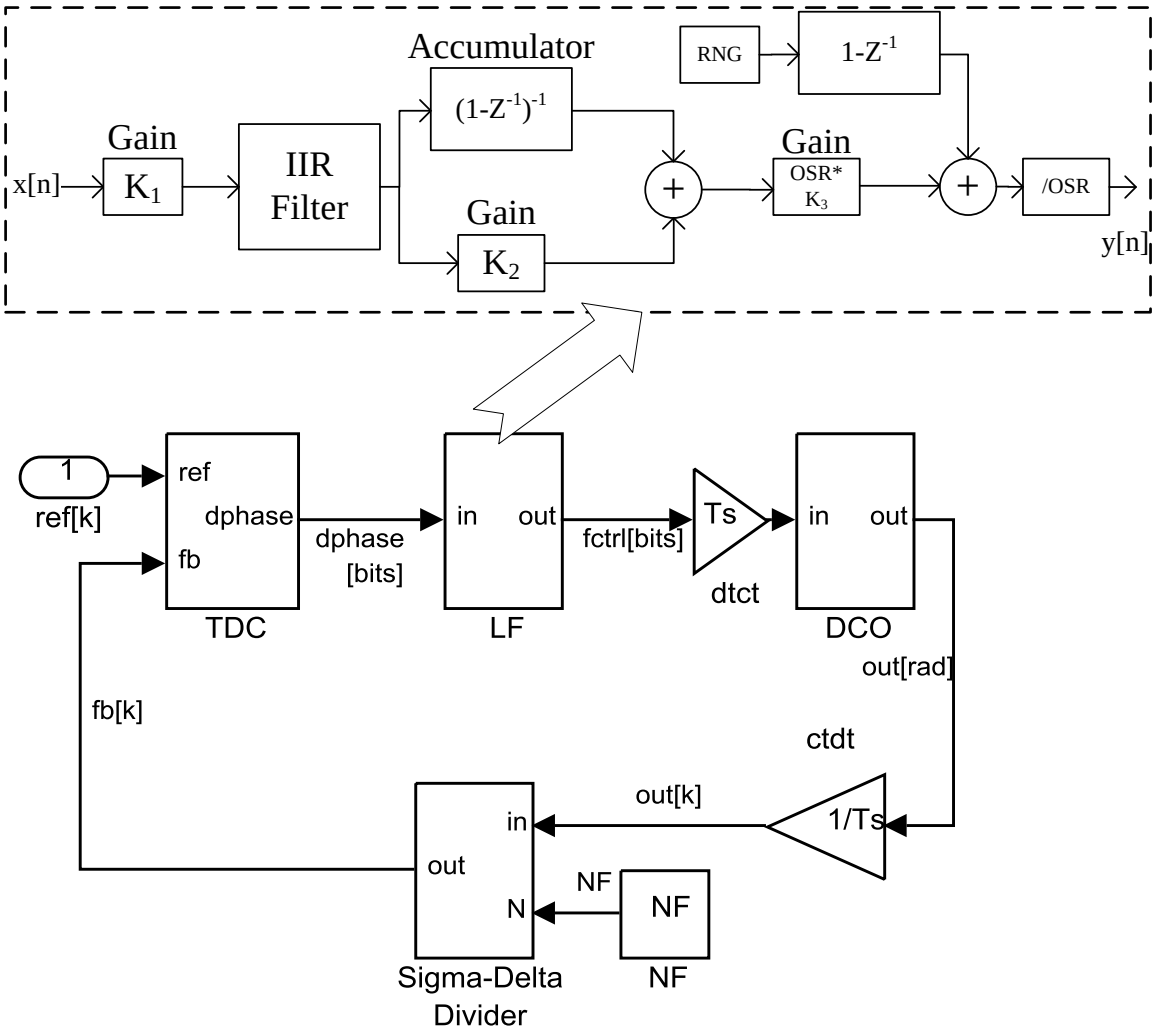
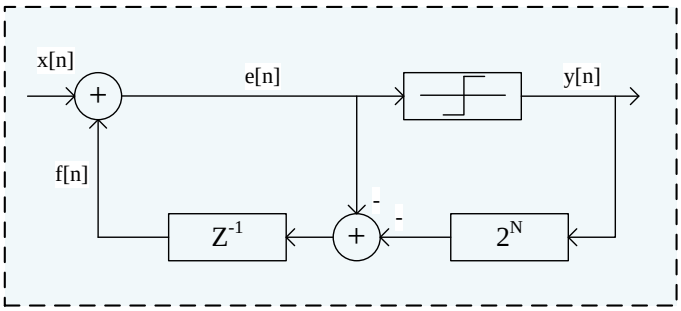


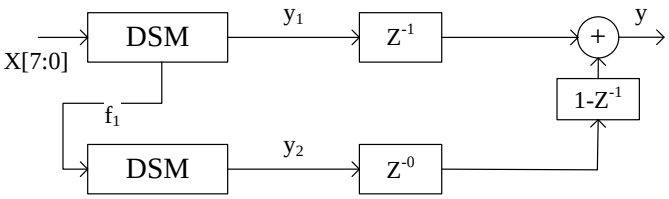
Figure 5.1. Top-level phase model and digital loop filter

bit input and one bit output 1^{st} order digital $\Sigma\Delta$ cores are implemented as shown in Figure 5.2a using delay, comparison to zero, and addition operations. The output of the modulator is one bit and the density of high and low duration of the output signal is controlled by the fractional part F of the clock multiplication value.

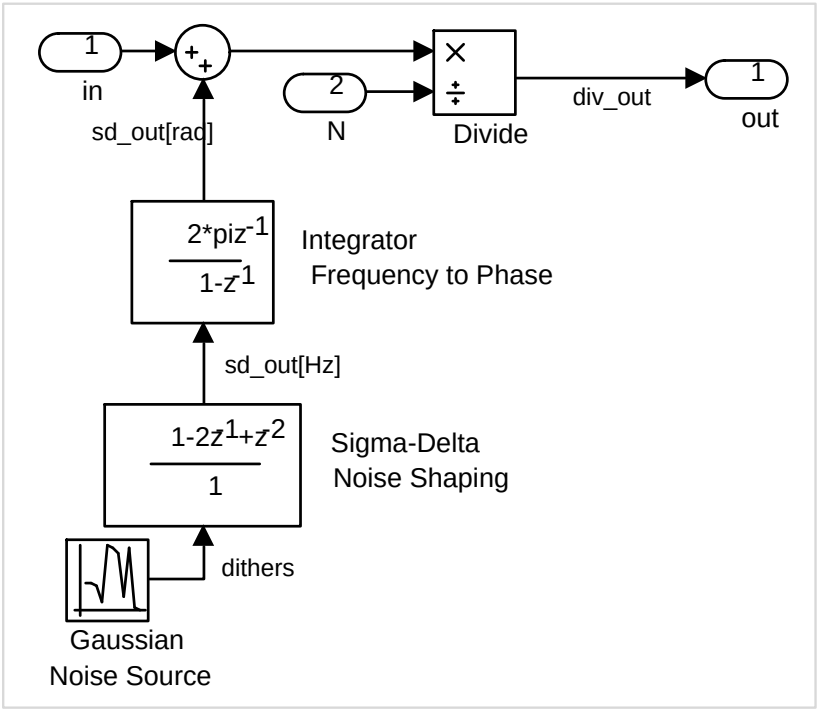
The ADPLL has a top-level controller which triggers the start-up calibrations. After the calibration of the DCO is complete, the locking procedure starts. Lock signal is asserted if the counted feedback clocks and reference clocks are within 0.1% of each other after every long observation window.



(a) Digital ΣΔ architecture

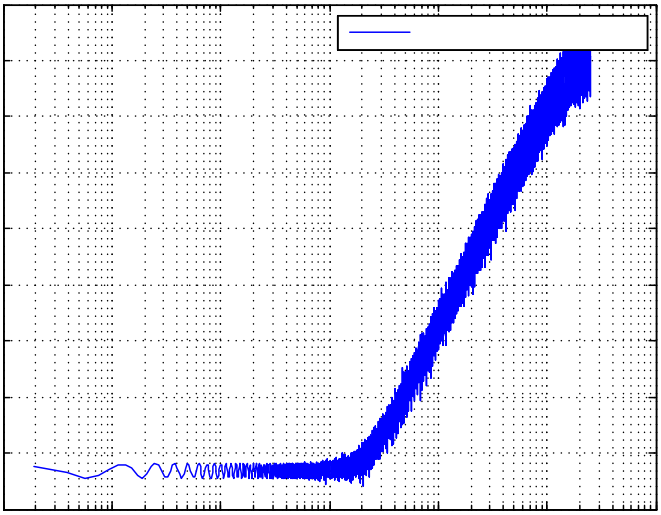


(b) MASH-11 topology



(c) Phase domain model

Figure 5.2. Digital ΣΔ module architecture and phase modeling



(d) Phase noise profile

Figure 5.2. Digital ΣΔ module architecture and phase modeling

5.2. Loop transfer function generation for user specifications

Maintaining a well-known closed loop transfer characteristic under various operating conditions is crucial in order to guarantee stability and keep phase noise bounded. A type-2 order-2 closed loop transfer function in the format of Eq. 5.1 is selected for its phase error minimization property.

$$G(s) = \frac{1}{s+a}\frac{s+d}{(s+b+cj)(s+b-cj)} \tag{5.1}$$

After the user enters the specifications to the GUI, the bandwidth of the loop is set, which means the poles and zeros of the system can be calculated. As a next step, the corresponding open loop and the loop filter transfer functions are calculated from the closed loop system. The poles and zeros for $G(s)$ are chosen from Butterworth

polynomials in order to obtain a maximally flat pass band with f_o as the cut-off frequency and roll-off with -40 dB/dec in the stop band.

The open loop transfer function for the proposed type-2 order-2 design is defined as in Eq. 5.2. K is the open loop gain, w_p is the pole, and w_z is the zero frequency. When the loop is closed, $G(s)$ can be expressed in terms of the parameters of $A(s)$ as expressed in Eq. 5.3.

$$A(s) = \frac{K}{s^2}\frac{s + w_z}{s + w_p} \tag{5.2}$$

$$G(s) = \frac{A(s)}{1 + A(s)} = \frac{K(s + w_z)}{s^3 + w_p s^2 + Ks + Kw_z} \tag{5.3}$$

Comparing $G(s)$ from Eq. 5.3 and Eq. 5.1 reveals that the open and closed loop zeros (i.e. d and w_z) are the same. Open loop transfer function terms are expressed in terms of $G(s)$ parameters as given in Eq. 5.4.

$$w_p = a + 2b, w_z = d$$

$$K = \frac{b^2 + c^2 + 2ab}{w_p/w_z}$$

$$a = \frac{w_z(b^2 + c^2)}{b^2 + c^2 - 2bw_z} \tag{5.4}$$

In order to realize a $G(s)$ with Butterworth polynomial, the complex poles identified by b and c are placed at magnitude and phase of $f_o\angle \pm 135^\circ$ as in Figure 5.3a. The pole magnitudes define the desired bandwidth, while the angle of the poles guarantees a fixed damping coefficient and hence the system's stability. The user determines the zero location by setting the f_o/f_z field in the GUI. Traditionally, the zero is set to $f_o/10$ in order to compensate the real pole a and maximize the flatness of the pass band in $G(s)$. The closed loop real pole a is automatically set when the parameters b,

c, and w_z are set as seen in Eq. 5.4. Additionally, using Eq. 5.4, the open loop gain K and open loop pole w_p are calculated respectively. Figure 5.3c shows the calculated open loop frequency response. With the integration in the oscillator and the integrator in loop filter, $A(s)$ starts with a -40 dB/dec roll-off from zero offset frequency. Zero placement before the 3^{rd} pole frequency provides compensation, and a phase margin of 60° is obtained. All the blocks in the system except the loop filter have a combined transfer function in the form of an integrator with a gain component. Therefore, the loop filter transfer function has to be of the form given in Eq. 5.5 to create $A(s)$.

$$H(s) = \frac{K_{alf}}{s}\frac{s + w_z}{s + w_p} \tag{5.5}$$

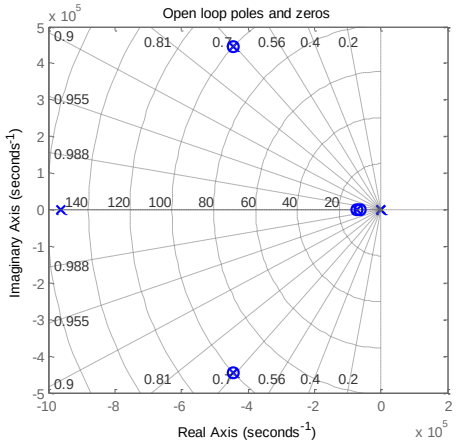
As the ADPLL incorporates a digital loop filter, the digital equivalent of this filter needs to be built in a form given in Eq. 5.6. In [99], a bi-linear transformation is carried out and the digital loop filter gain, poles, and zeros are represented in terms of their analog counterparts as in Eq. 5.6.

$$H[z] = \frac{K_{lf}}{1 - z^{-1}}.\frac{1 - b_1 z^{-1}}{1 - a_1 z^{-1}}$$

$$K_{lf} = \frac{T_{res_{eff}}}{T_{out}}\frac{K}{K_v}\frac{w_p}{w_z}\frac{a_1}{b_1}T_s$$

$$a_1 = \frac{1}{1 + w_p T_s}, b_1 = \frac{1}{1 + w_z T_s} \tag{5.6}$$

The parameters required for the calculation of the digital loop filter pole a_1 and zero b_1 using Eq. 5.6 are already known from the previous steps. The loop filter gain K_{lf} is calculated by dividing the calculated open loop transfer function by the transfer function of all the remaining elements as given in Eq. 5.6. At this point, K_v and $T_{res_{eff}}$ are used from the previously generated DCO and TDC HLS results. CellPLL carries out all the calculations and reports the details of the transfer functions from the GUI and prints open and closed loop pole-zero maps as given in Figure 5.3. The phase noise model of the digital loop filter is updated with the results, thereby completing



(a) Closed loop pole-zeros

(b) Open loop pole-zeros

(c) Open loop transfer function A(s)

(d) Loop filter $H[z]$

Figure 5.3. Characteristics of the loop

the design of the loop. For any given user specification, transfer functions can be generated. However, the possible set of implementations that can be generated are bounded by the set of realizations that the embedded ADPLL template can support.

5.3. Implementation of all-digital PLL

To compare the performance of the MIMO quantization noise suppression with the conventional SIMO method and also create a fully synthesizable standard cell AD-PLL, a design with the specifications given in Table 5.1 is implemented and simulated in 55 nm CMOS technology. Design of the remaining sub-blocks and top-level PLL control are presented in Subsection 5.3.1 and Subsection 5.3.2.

Table 5.1. Parameters for PLL design example 1

Parameter	**Value**
VCO frequency	0.65-1.35 GHz
Clock Multiplication Range	16-30
TDC resolution	20 ps/LSB
DCO/TDC oscillation mismatch	15%
DCO phase noise	-100 dBc/Hz at 1 MHz
PLL bandwidth	100 kHz

5.3.1. Digital loop filter

The loop filter is implemented digitally as shown in Figure 5.4a. With the help of digital scaling, accumulation, and IIR filtering operations, proportional and integral paths of the loop are created and the structure digitally imitates a type-2 order-2 PLL analog loop filter (Figure 5.4b). The IIR loop filter is a 1^{st} order circuit similar to the one in Figure 5.4c with the characteristics given in Eq. 5.7.

$$H[z] = \frac{1-\alpha}{1-\alpha z^{-1}} \tag{5.7}$$

In order to calculate the loop filter parameters, CellPLL is used with the specifications resulting in open loop parameters K, F_p and F_z for use with an analog filter. K is the open loop gain, F_p is the pole and F_z is the zero frequency. For the specified system, the analog equivalent transfer function and its calculated parameters are given in Eq.

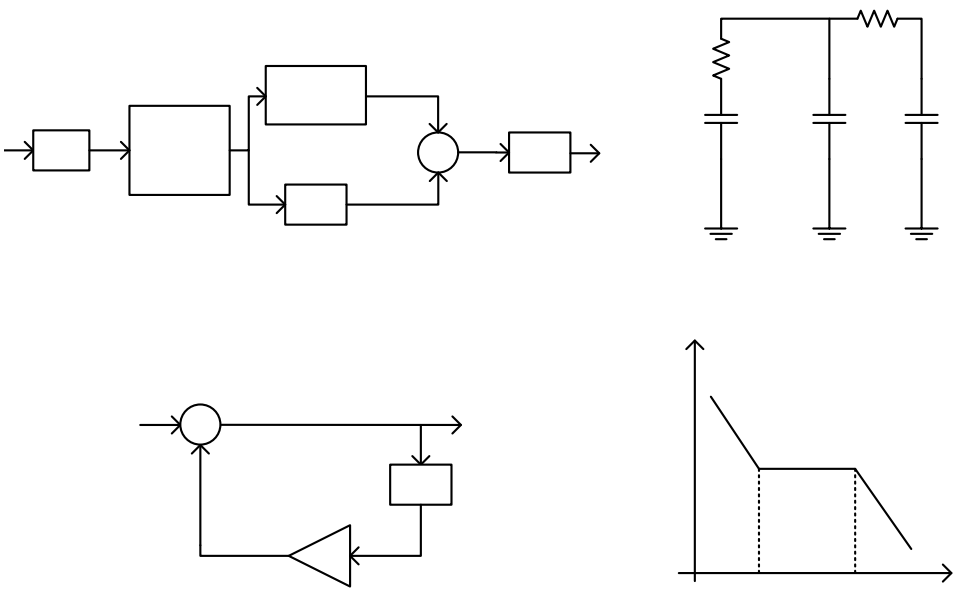


Figure 5.4. a. Z domain model of the digital filter, b. Analog equivalent of loop filter, c. First order IIR structure, d. Frequency response of the digital loop filter

5.8 and Eq. 5.9. The corresponding loop filter is with gain of K, pole and zero frequencies F_p and F_z.

$$A_{calc}(s) = \frac{K}{s^{type}} \frac{1 + s/w_z}{1 + s/w_p} \tag{5.8}$$

$$K = 3,004 \times 10^{10},\ f_p = 1,531 \times 10^5,\ f_z = 10^4 \tag{5.9}$$

Frequency response of the loop filter is shown in Figure 5.4d. This analog filter transfer

$$a_1 = \frac{1}{1 + w_p T_s},\ b_1 = \frac{1}{1 + w_z T_s} \tag{5.11}$$

$$K_{LF} = T_s \frac{T_{res_{eff}}}{(T_s/N)} \frac{K}{K_v} \frac{w_p}{w_z} \frac{a_1}{b_1} \tag{5.12}$$

When Eq. 5.11 and Eq. 5.12 are solved with the values provided in Eq. 5.13, a_1, b_1, K_{LF} are calculated as in Eq. 5.14, Eq. 5.15, and Eq. 4.2:

$$T_{res_{eff}} = 12\ ps,\ K_v = 1\ MHz/LSB,\ T = 10\ ns,\ N\ = 10 \tag{5.13}$$

$$b_1 = \frac{1}{1 + 2\pi.10.10^3.100\ MHz^{-1}} = 0,999372 \tag{5.14}$$

$$a_1 = \frac{1}{1 + 2\pi.153.10^3.100\ MHz^{-1}} = 0,990478 \tag{5.15}$$

$$K_{LF} = \frac{12.10^{-12}.3.004.10^1 0.153.0.990478}{10^{-8}.10^{-1}.1.10^6.10.0,999372.10^8} = 54,6.10^{-6} \tag{5.16}$$

The digital transfer function approximation is realized with a circuit similar to the one in Figure 5.4a. Transfer function of the circuit is given in Eq. 5.17 and solved into the same format as the desired digital filter response in Eq. 5.18.

$$H[z] = K_1 \frac{1 - \alpha}{1 - \alpha z^{-1}} \frac{K_2 - K_2 z^{-1} + 1}{1 - z^{-1}} \tag{5.17}$$

$$H[z] = K_1 (1 - \alpha)(1 + K_2) \frac{1}{1 - z^{-1}} \frac{1 - \frac{K_2}{1 + K_2} z^{-1}}{1 - \alpha z^{-1}} \tag{5.18}$$

The desired transfer function is matched to the transfer function of the actual implementation to give the following parameters for use during the implementation as given

5.3.2. MASH11 digital Sigma-Delta modulator

ADPLL uses $\Sigma\Delta$ modulator shown in Figure 5.2b for two purposes. Firstly, a fractional multiplication of the input frequency is obtained. Secondly, a priori known output sequence of the modulator is leveraged to estimate the resolution in TDC blocks.

A 2^{nd} order MASH11 digital $\Sigma\Delta$ modulator topology similar to the one in [104] is implemented by cascading first order digital $\Sigma\Delta$ blocks given in Figure 5.2a. These 1^{st} order $\Sigma\Delta$ cores with 8-bit inputs and 1-bit output are implemented using delay, compare to zero, and add operations. The output of the modulator is a 4-bit signed vector and it varies between -3 and 2 depending the fractional part F clock multiplication value. Output of the divider is the feedback clock and it is used as the update clock for the fractional modulator. The digital $\Sigma\Delta$ modulator's core and top-level functionality is simulated as shown in Figure 5.6 and Figure 5.7 to verify that the bit stream average matches the desired input fraction.

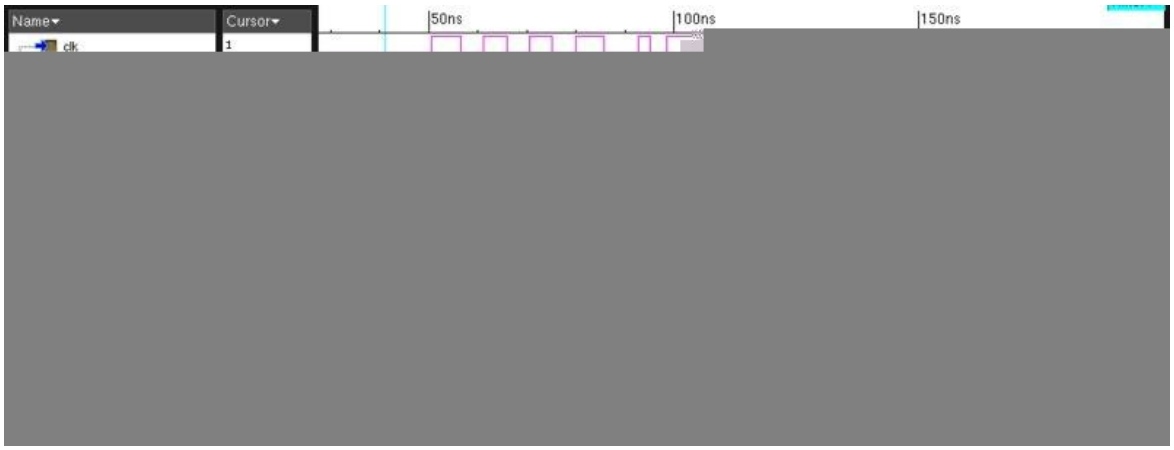


Figure 5.7. MASH11 top-level simulation results

designs, the implemented design is significant in two aspects.

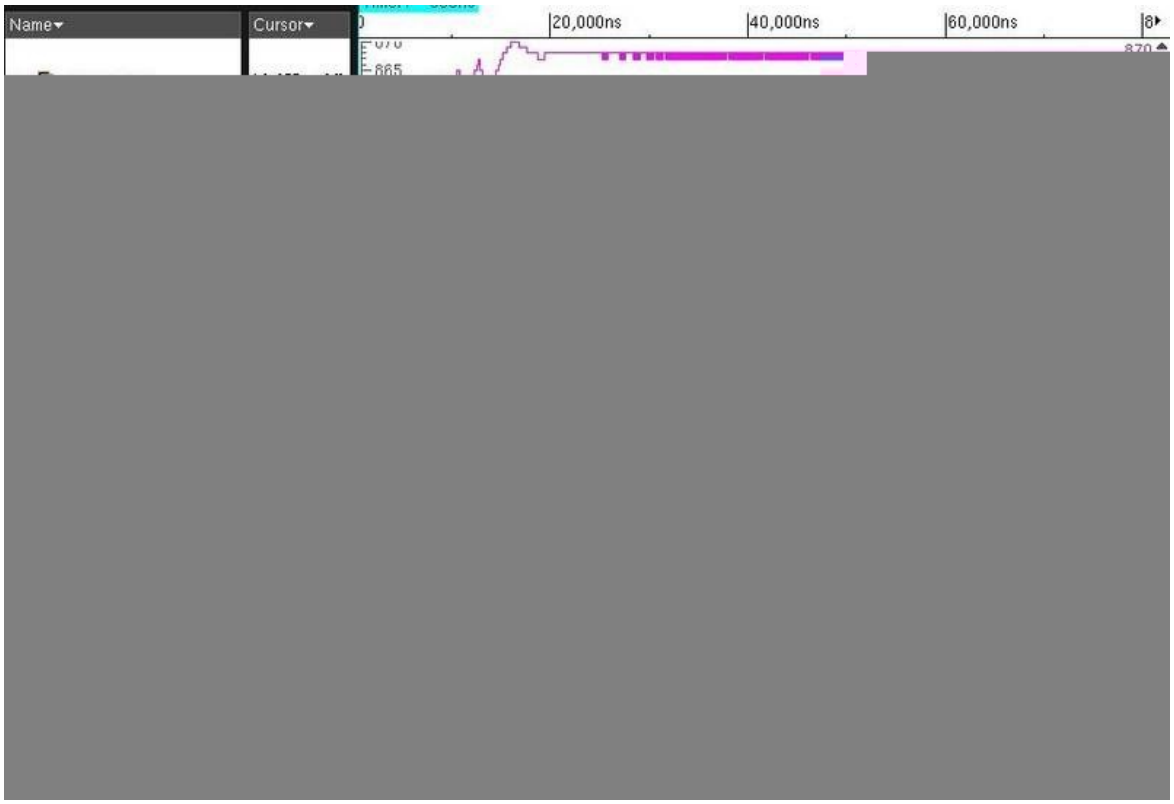


Figure 5.8. Important signals ın the top-level locking simulation

The TDC uses the proposed MIMO quantization noise suppression method and reduces the quantization noise by a factor of $\sqrt{2}$ compared to the SIMO case presented in [97] while using the same number of gates and power as shown in Figure 4.6. Compared to similar designs [1, 6, 7], better jitter performance is obtained with the

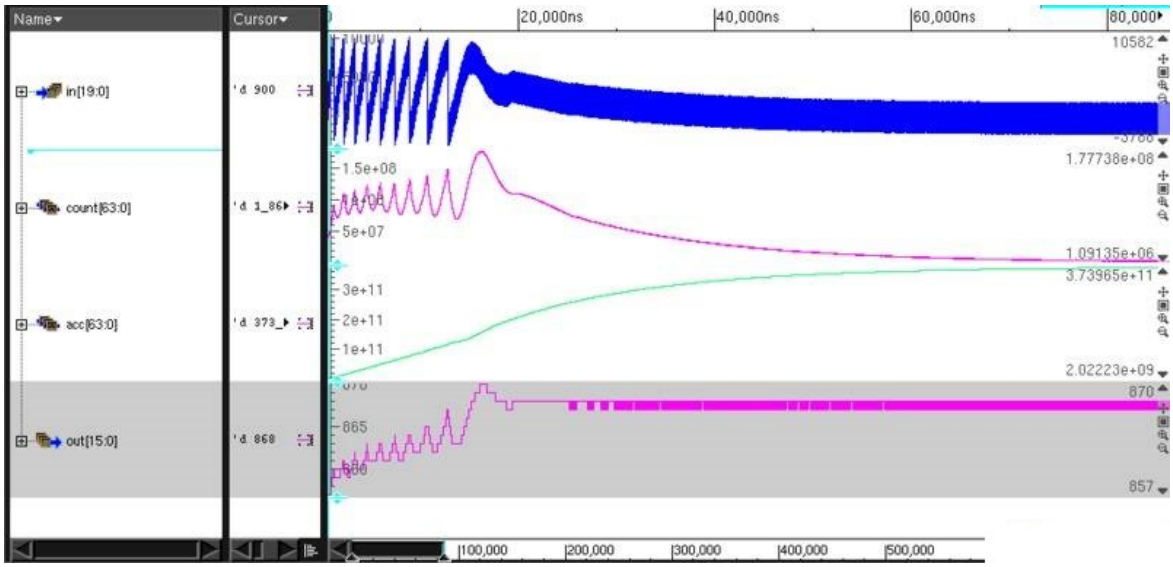


Figure 5.9. Loop filter dynamics during top-level locking simulations

help of improved TDC resolution as shown in the simulated phase noise profile in Case 1 of Figure 6.1b. There is room for improvement in area and power consumption when compared to [1,6,7]. While it would limit the tuning range, power and area reduction is possible by reducing the number of rings in the DCO.

Table 5.2. ADPLL implementation results and comparison

Parameter	**This Work**	[1]	[7]	[6]	[105]
Type	**Std Cell**	**Custom**	**Custom**	**Std Cell**	**Custom**
Process (nm)	55	65	28	65	65
Supply (V)	1	1.1	1	1.1	1.2
Frequency (GHz)	0.65 - 1.35	0.6-0.8	0.01-0.63	1.5-2.7	0.6-0.8
Long Term Jitter	1.78 ps_{rms}	193 ps_{pp}	30 ps_{rms}	36 ps_{pp}	30 ps_{rms}
Area (mm^2)	0.09	0.03	0.03	0.04	0.03
Power (mW@MHz)	17.5@800	5@800	3@250	13.7@2500	3.2@800

5.4. Phase Modeling

The phase model of the digital loop filter is represented by its z-domain transfer function. On the other hand, a $\Sigma\Delta$ modulator phase model is created by generating the phase noise from a frequency noise as shown in Figure 5.2c. As the output of the modulator is either high or low, the quantization noise due to this noise source will have a variance of $1/12$. As the PSD from discrete domain noise source to continuous domain output requires PSD mapping [2], the defined variance is divided by T_s. Quantization noise profile in frequency domain for this modulator is well known to have a shape with 40 dB/dec slope after cut-off frequency as given in Figure 5.2d. This frequency profile can be obtained by shaping a white noise source RNG with mentioned variance through a high-pass filter. Finally, the frequency profile is integrated in order to obtain phase noise with 20 db/dec slope and the phase noise is added to the input of the feedback divider.

Another first order $\Sigma\Delta$ modulator is employed at the output of the digital loop filter which, oversamples the reference clock with a clock generated by dividing the output clock and dither the 12 bit wide input of the DCO in order to improve DCO resolution. The phase model calculates the phase noise contribution due to the quantization in DCO with the phase model given in Figure 5.1. The width of the DCO control word is fixed in the embedded DCO template hence, limiting the space of possible output frequency ranges.

Using Simulink, all the modules in the design are modeled in phase domain as explained in previous sections. To the extent of author's knowledge, implementation of the phase model simulation in Simulink is new and it will allow flexibility and ease of use compared to equation based analytic methods when analyzing complex ADPLLs that have dual, cascaded, and hybrid loop architectures. User specifications such as input reference frequency f_{ref}, worst TDC resolution $\Delta T_{tdc_{max}}$, fractional feedback divider value NF, loop bandwidth f_o, transfer function's zero frequency f_z, and DCO noise level at a specified offset frequency are set by the user from the GUI. Other important design parameters such as the actual TDC resolution ΔT_{tdc} and DCO gain

per LSB K_v are calculated by the HLS algorithms. These parameters are used to update the parameters of the phase model for phase noise simulation. CellPLL uses T_s as the sampling time for discrete steps of the phase model.

At the top-level of the phase model, the TDC compares the phases of the feedback and reference clocks and generates an output $Err[k]$ which represents the phase error in digital vector form. The loop filter filters the output of the TDC and generates the frequency control word for the DCO. The DCO integrates the frequency control word into phase and generates the phase ramp at its output. The feedback divider divides the high-speed DCO output by $N + F$ and adds the quantization noise due to the fractional division as shown in Figure 5.1. The DCO output works in continuous time domain, while the rest of the model is in discrete domain; therefore, relative domain crossing blocks are added between boundaries to account for the discrete to continuous time domain phase conversion. The reference clock's phase is assumed constant in order to simulate phase noise perturbations around the carrier frequency without having to wait for the ADPLL to lock. While this lets the phase noise simulation in CellPLL be fast, the settling behavior of the system cannot be observed. If the reference clock referred output noise is desired for analysis, a noise source with zero mean can be added to the reference clock input. Details on how CellPLL runs phase noise simulations is explained in Figure 5.6.

5.5. Noise transfer functions

Three noise sources are modeled in the phase noise simulation environment. The profiles of the noise sources have been illustrated in the previous sections. This section provides the details on how the noise sources are referred to the output by observing the noise transfer functions (NTFs). Using a linearization analysis in MATLAB, the NTFs for the quantization noise of the TDC, DCO, and $\Sigma\Delta$ modulator are generated as seen in Figure 5.10.

The TDC quantization noise has a flat power spectral density (PSD) at the noise source as seen in Figure 4.7. This noise profile is shaped by the NTF which is a scaled

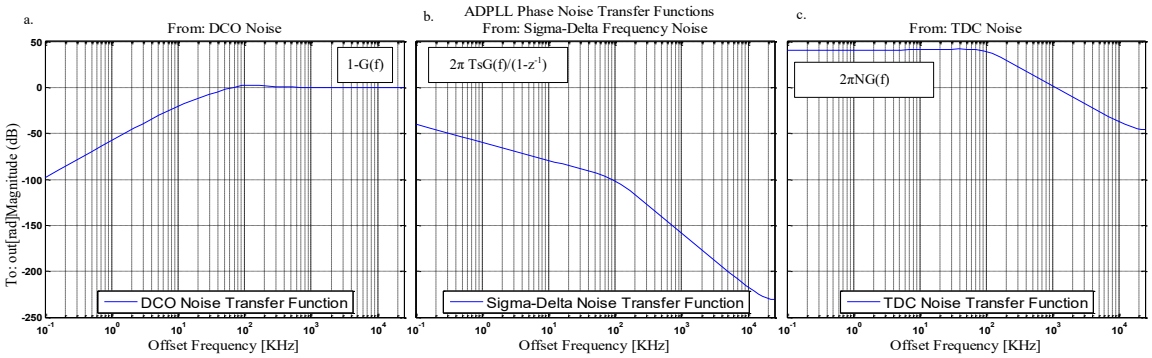


Figure 5.10. Noise transfer functions for all noise sources

version of $G(f)$ as seen on Figure 5.10c. The observed amount of scaling is in line with a $2\pi N$ scaling given in [2]. As the bandwidth of the loop is increased, the filtering of this noise source is reduced and the jitter increases correspondingly.

In Figure 3.8, the DCO phase noise profile at the source was presented. The NTF for the DCO is a high-pass filter with a unity pass band as given on Figure 5.10a. The observed NTF for the DCO matches the analytic result $1 - G(f)$ from [2]. The high pass filter filters the elevated noise at low offset frequencies of the DCO noise source. This shapes the decaying noise profile of the DCO such that the increased bandwidth of the ADPLL results in a reduced contribution from this noise source to the jitter. Furthermore, DCO quantization noise is also modeled with details explained in Figure 5.4 and Figure 5.6. It is seen that the noise is shaped with a high pass filter function when referred to the output.

The $\Sigma\Delta$ modulator quantization noise defined in Figure 5.2d is shaped by the NTF given in Figure 5.10b. The NTF is a superposition of an integrator and the closed loop transfer function $G(s)$. The NTF roll off starts with -20 dB/dec and after the cut-off frequency continues at roll off with -60 dB/dec. This NTF includes the integration that converts the noise profile defined in the frequency domain to the phase domain as well. In other words, the NTF includes the integrator seen on Figure 5.2c and the noise source is defined to be at the input of this integrator. The given $NTF[z] = 2\pi T_s G(f)/(1-z^{-1})$ in [2] correlates with CellPLL results. The contribution to jitter from this noise source can be reduced by lowering the bandwidth of the ADPLL

specifications. Additionally, reference phase noise is transferred with the closed loop transfer function to the output.

5.6. High-level synthesis and phase noise analysis

CellPLL needs to parse the standard cell libraries before first use. This operation has to be done only once per library and the results are stored internally in the tool's database. After the standard cell libraries are parsed, the tool can be used to specify ADPLL parameters and generate the ADPLL design, analyze the design for phase noise performance, and generate Verilog code output together with the synthesis scripts. The user enters the ADPLL specifications in the GUI and executes the flow. When executed, the tool follows the procedure given in Figure 5.11.

CellPLL contains only one flexible circuit topology. This template can be swapped by alternate designs by users, but the tool does not automatically change the circuit templates depending on power and area targets or try to identify the trade-off between performance, noise and power. However, for the embedded circuit template, the tool actually inherently minimizes area and power consumption for the specified performance by the user. HLS starts from minimum and enlarges the cell sizes for the gate level portions of the DCO and TDC only as much as needed. And the remaining RTL code of the design is optimized by the RTL synthesizer. This ensures that we implement the desired performance with minimum power and area allowed under this circuit topology. The user can get early power and area estimates after RTL synthesis.

Verilog code for the feedback divider is generated using the multiplication range designated by the minimum and maximum N value. The bit vector widths are determined from the N_{min}, N_{max} and the divider code is generated. A 2^{nd} order 1-bit $\Sigma\Delta$ modulator is employed in the ADPLL architecture. The Verilog code for the modulator is generated by the tool. Lock detector logic in the ADPLL architecture is predefined and it does not change depending on the parameters of the PLL. Its Verilog code is generated by the tool. Digital loop filter architecture is predefined by the ADPLL architecture. The gains and cut-off frequencies used in the sub blocks of the

LPF is calculated during transfer function design by the tool. Finally, the loop filter is written out with updated parameters for the calculated transfer function. Verilog code for ADPLL top-level block is generated using the parameters specified in the GUI. The bit vector widths are determined from parameters specified in the GUI and the corresponding code is written out. The divider related vectors widths are determined by $N_{min/max}$. Along with the ADPLL design and synthesis scripts, some digital behavioral models and digital test bench is also provided for digital fast-simulations along with simulation scripts. These are useful for rapid iteration of design verification before proceeding with the time consuming mixed-signal simulations.

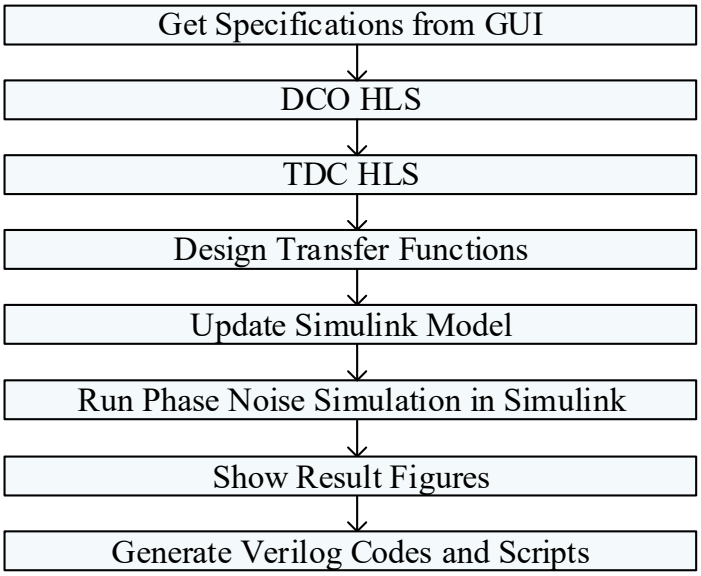


Figure 5.11. CellPLL top-level execution flow

After the transfer function design has been completed, the open and closed loop transfer functions, and pole-zero locations are reported graphically. Next, loop filter circuit implementation parameters K_1, K_2, and α are calculated using the calculated digital loop filter transfer function's gain K_{lf}, pole a_1, and zero b_1.

The Simulink model parameters are updated using all of the design parameters of the generated ADPLL and the model preparation for phase noise simulation runs are completed. The phase model simulation is run four times with each noise source enabled separately and with all of the noise sources enabled. This allows analysis

Figure 5.12. Effect of increasing TDC non-linearity on the phase noise profile for $BW = 100kHz$ $(0\%, 1\%, 2\%)$

and reporting of each noise source separately. Increasing the length of the simulation increases the PSD resolution at low offset frequencies while increasing the run time. By default, 2^{10} samples are simulated resulting in good resolution at as low as 1 kHz offset frequency. After the simulations are completed, the phase noise results are

CPPSIM as shown in Figure 5.13.

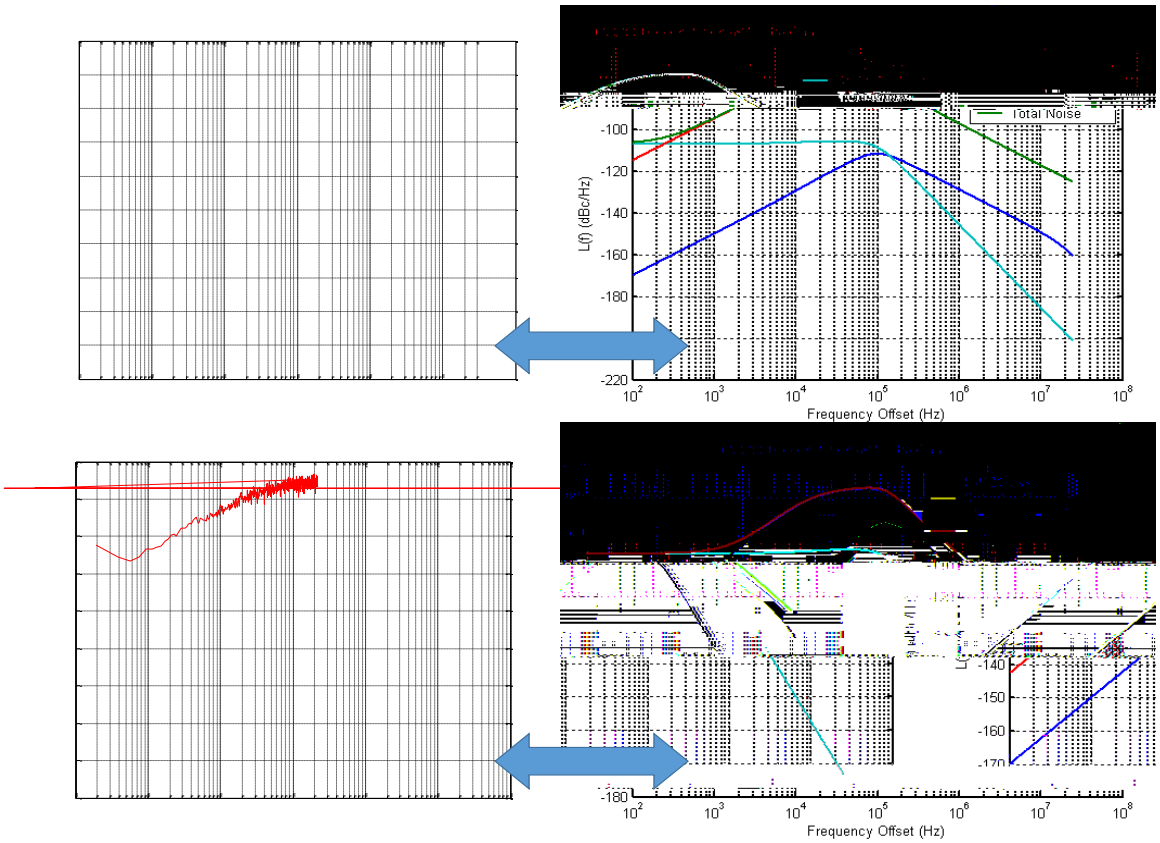


Figure 5.13. Phase noise for two design cases and correlation to CPPSIM (No TDC non-linearity)

6. RESULTS AND DISCUSSION

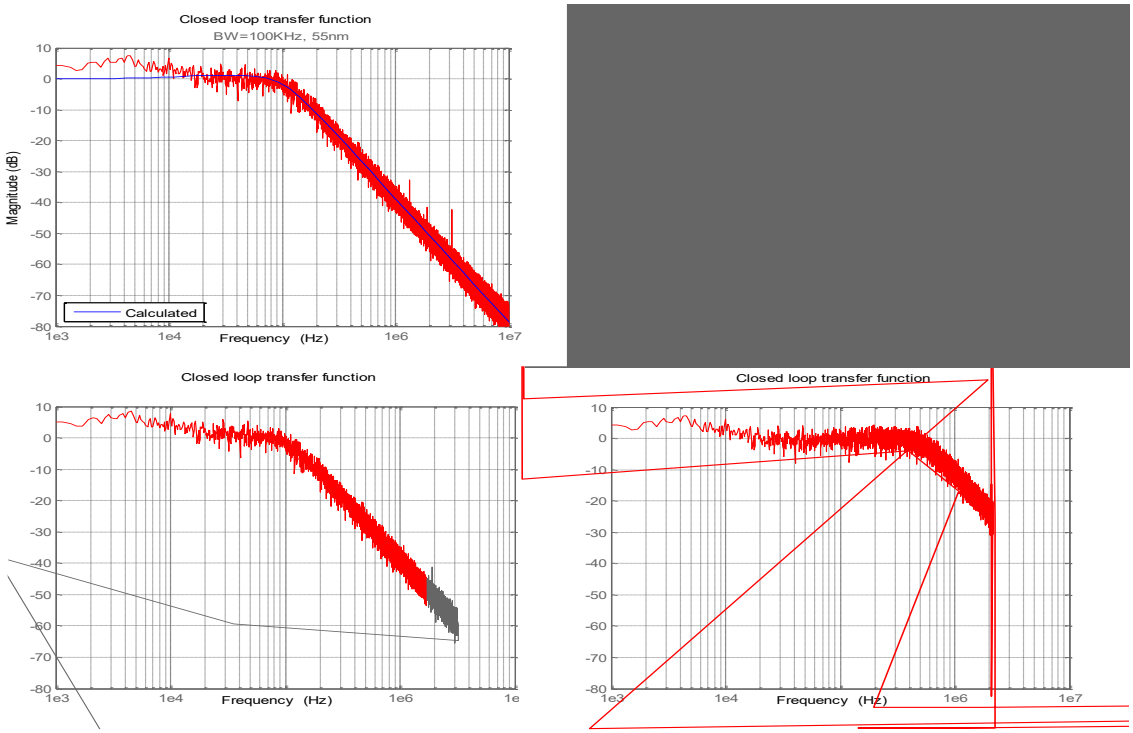
Four design examples have been generated using the tool with specifications given in Table 6.1. All of the case parameters are picked such that the spectral mask of the OLDI protocol can be satisfied. Two applications are chosen for these parallel to serial clock generation PLLs. For the purpose of filtering jitter, a 100 kHz bandwidth is chosen in the first application. On the other hand, for the spread spectrum tracking application, a 500 kHz bandwidth is selected to provide greater input tracking capability. In order to show rapid process migration capability, cases 3 and 4 are selected as the 65 nm counterparts of the first two cases implemented on 55 nm.

Table 6.1. Implemented design examples using CellPLL

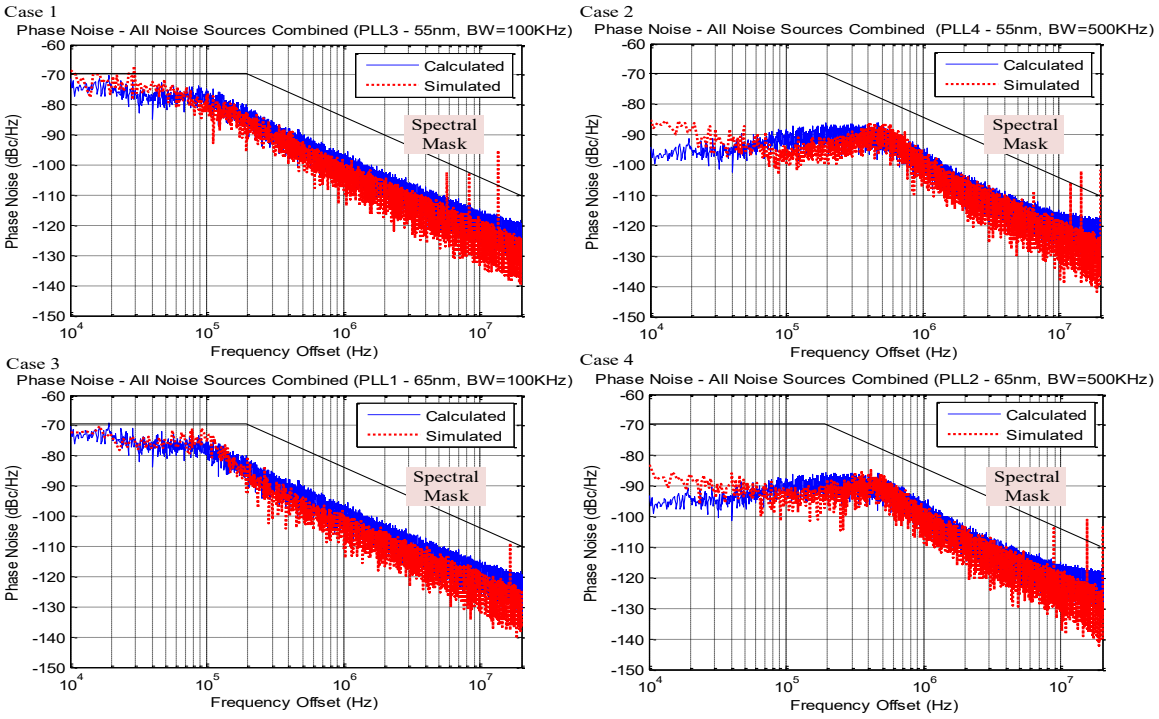
Case	Node $[nm]$	BW(f_o) $[kHz]$	NF	$\Delta T_{del_{max}}$ $[ps]$	Dco_{PN}@20MHz $[dBc]$	f_{ref} $[MHz]$	*Power* [mW]	*Area* $[mm^2]$
1	55	100	16-30	20	-123	50	17.4	0.102
2	55	500	16-30	20	-123	50	18.1	0.108
3	65	100	16-30	20	-123	50	17.9	0.141
4	65	500	16-30	20	-123	50	19.3	0.149

With an i5 Intel Processor laptop, the CellPLL tool takes about 1 minute to generate the loop, analyze its phase noise, and run the high-level synthesis. Synthesizing the gates from Verilog output codes lasts 2 hours per design with a Intel Xeon processor.

Area and power numbers for these cases are reported in Table 6.1. The area occupation seems to be dominated primarily by DCO (79%) and TDC (18%) while the rest of the circuits only consume 3% of the design.



(a) Simulation of bandwidth for generated ADPLLs



The example designs have been generated using the CellPLL and the generated Verilog codes have been synthesized. VSR space based router is used for auto placement and routing. The extracted post layout designs have been imported into Spectre APS transistor level simulator and a noise enabled transient simulation has been run to verify proper locking and phase tracking, and also to collect data for phase noise PSD generation. Data-sets for the instantaneous periods of ADPLL input and output are compared to generate the closed loop transfer function of the designs. This result is compared to the closed loop transfer function of the intended design from the CellPLL and matching is observed as shown in Figure 6.1a.

For each design, with $NF = 17.25$ the instantaneous periods of the ADPLL output are recorded to a data-set and parsed for generating phase noise figures as shown in Figure 6.1b. The phase model models 1% TDC non-linearity and the associated frequency spurs. However the divide value variations are not modeled and an average divide value is used, which prohibits the estimation of fractional spurs due to fractional-N operation. Transient simulations show the combined frequency spur performance of the loop. In Figure 5.13, it is seen that the phase noise is estimated to be dominated by the DCO for all of the cases. When compared to the phase noise estimation of CellPLL, it can be seen that the tool can predict the phase noise accurately and quickly. Our HLS tool runs in under a minute; the RTL synthesis, space based VSR auto place and routing and, parasitic extraction takes 1 day; while the transient post-layout simulations that we use to report our results last 4 days. Compared to very slow transient simulations, the tool provides a dramatic improvement during design iterations when the designer is trying to decide on the tradeoff between noise components.

The tool does not check if the phase noise results satisfy the spectral mask requirements. However, the designer can easily see the status of the overall phase noise PSD together with the spectral mask as seen on Figure 6.1b and tweak design specifications in order to satisfy the communication standard. From the results, it can be seen that the phase noise is dominated by the DCO and increasing the bandwidth of the ADPLL from case 1 to case 2 increases the suppression of DCO noise and result

in a lower phase noise profile. Additionally, results of cases 3 and 4 match cases 1 and 2 respectively, which illustrates that the technology migration can be handled properly. While CellPLL does not report area or power estimation for the generated designs, the RTL synthesis tool can be utilized for this estimation.

The tool was used to generate four ADPLL designs. Finally ADPLL was simulated in transient simulations and its locking behavior, calibration control and power analysis has been completed for the first case. For all of the cases, the correctness of the implemented closed loop ADPLL transfer functions and accuracy of the estimated phase noise profile are illustrated. Phase noise simulation engine is also verified by comparing it to a similar tool for analog PLLs. In other words, it was shown that the tool could accurately design, analyze, and implement the design examples while accurately estimating the phase noise performance.

7. CONCLUSION

In this thesis, a new standard cell PLL architecture based on Verilog codes and synthesis has been presented. The design includes several key items such as the novel MIMO quantization noise suppression technique. Additionally, a tool that is used for automatic design, analysis and design generation of proposed ADPLL architecture has been developed.

The design itself is significant in several aspects. Implemented ADPLL design successfully achieves superior jitter performance with the proposed MIMO quantization noise suppression method while staying in the ballpark for power and area consumption compared to other ADPLLs. This MIMO technique improves the effective resolution of the TDC. Therefore the phase noise component arising from the TDC quantization noise is suppressed by an additional order of $\sqrt{2}$ and its dominance on overall phase noise profile is suppressed.

Thanks to the use of standard cells, area hungry capacitors in LF, DCO and TDC are eliminated. Flexibility, portability and configurability targets for the design have been met with the synthesized standard cell digital design flow. Intended goals for the new ADPLL architecture have been achieved and the results prove that the ADPLLs' jitter and flexibility specifications are improved with the proposed methods.

Additionally, a new tool called CellPLL has been developed for generating ADPLLs and analyzing them. This tool emphasizes automatic design and analysis of ADPLL and digital loop filter transfer functions along with new methods for phase modeling.

CellPLL creates transfer functions directly for the closed loop system using the input specifications captured from the GUI. Loop filter transfer function is extracted by dividing the loop transfer function to the gains of other sub-blocks. Furthermore, a phase model is developed in order to estimate the phase noise performance of the loop

for every noise source independently. This phase model saves the user a considerable amount of simulation time by reducing the long time domain phase noise iterations.

Additionally, for the first time a tool that also provides the actual design for the calculated ADPLL with the help of HLS is presented. CellPLL uses the embedded design template during automatic design implementation. According to the user specifications, STA is run and sub-blocks are updated before the Verilog output code and synthesis scripts are written. When compared to $C++$ based tools, the programming and modeling environment established in MATLAB allows faster tool adoption and ease of modification.

During the period of this study, it has been gladly observed that the research community continued working on this topic in similar directions with this thesis and contributed to the development ADPLLs.

REFERENCES

1. Staszewski, R. B., K. Waheed, S. Vemulapalli, F. Dulger, J. Wallberg, C. M. Hung and O. Eliezer, "Spur-free all-digital PLL in 65nm for mobile phones", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 52–54, February 2011.

2. Perrott, M., M. Trott and C. Sodini, "A modeling approach for Sigma;- Delta; fractional-N frequency synthesizers allowing straightforward noise analysis", *Solid-State Circuits, IEEE Journal of*, Vol. 37, No. 8, pp. 1028–1038, August 2002.

3. Su, R., S. Lanzisera and K. S. J. Pister, "A 2.6psrms-period-jitter 900MHz all-digital fractional-N PLL built with standard cells", *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, pp. 455–458, September 2011.

4. Fanori, L., A. Liscidini and R. Castello, "3.3GHz DCO with a frequency resolution of 150Hz for All-digital PLL", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 48–49, February 2010.

5. Yang, S.-Y. and W.-Z. Chen, "A 7.1mW 10GHz all-digital frequency synthesizer with dynamically reconfigurable digital loop filter in 90nm CMOS", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 90–91,91a, February 2009.

6. Park, Y. and D. D. Wentzloff, "An all-digital PLL synthesized from a digital standard cell library in 65nm CMOS", *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, pp. 1–4, September 2011.

7. Kim, W., J. Park, J. Kim, T. Kim, H. Park and D. Jeong, "A 0.032mm2 3.1mW synthesized pixel clock generator with 30psrms integrated jitter and 10-to-630MHz DCO tuning range", *Solid-State Circuits Conference Digest of Technical*

Papers (ISSCC), 2013 IEEE International, pp. 250–251, February 2013.

8. Lau, C. and M. Perrott, "Fractional-N frequency synthesizer design at the transfer function level using a direct closed loop realization algorithm", *Design Automation Conference, 2003. Proceedings*, pp. 526–531, June 2003.

9. Mena, J., R. Deken, J. Coker, M. Johnstone, S. Ramirez and P. Frey, "High level synthesis of a Front End filter and DSP engine for analog to digital conversion: a case study", *VLSI Test Symposium (VTS), 2010 28th*, pp. 252–252, April 2010.

10. Hunter, R., T. Fuhrman and D. Thomas, "Working chips from high level synthesis: a case study from industry", *Custom Integrated Circuits Conference, 1994., Proceedings of the IEEE 1994*, pp. 144–147, May 1994.

11. Balcioglu, Y. and G. Dundar, "A synthesizable Time to Digital Converter (TDC) with MIMO spatial oversampling method", *New Circuits and Systems Conference (NEWCAS), 2015 IEEE 13th International*, pp. 1–4, June 2015.

12. Deng, W., A. Musa, T. Siriburanon, M. Miyahara, K. Okada and A. Matsuzawa, "A 0.022mm2 970 uW dual-loop injection-locked PLL with -243dB FOM using synthesizable all-digital PVT calibration circuits", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, pp. 248–249, February 2013.

13. Anand, T., M. Talegaonkar, A. Elshazly, B. Young and P. K. Hanumolu, "A 2.5GHz 2.2mW/25 on/off-state power 2psrms-long-term-jitter digital clock multiplier with 3-reference-cycles power-on time", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, pp. 256–257, February 2013.

14. Elkholy, A., A. Elshazly, S. Saxena, G. Shu and P. K. Hanumolu, "15.4 A 20-to-1000MHz 14ps peak-to-peak jitter reconfigurable multi-output all-digital clock generator using open-loop fractional dividers in 65nm CMOS", *Solid-State Cir-*

cuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International, pp. 272–273, February 2014.

15. Chen, Z. Z., Y. H. Wang, J. Shin, Y. Zhao, S. A. Mirhaj, Y. C. Kuan, H. N. Chen, C. P. Jou, M. H. Tsai, F. L. Hsueh and M. C. F. Chang, "14.9 Sub-sampling all-digital fractional-N frequency synthesizer with -111dBc/Hz in-band phase noise and an FOM of -242dB", *Solid- State Circuits Conference - (ISSCC), 2015 IEEE International*, pp. 1–3, February 2015.

16. Shen, K. Y. J., S. F. S. Farooq, Y. Fan, K. M. Nguyen, Q. Wang, A. Elshazly and N. Kurd, "19.4 A 0.17-to-3.5mW 0.15-to-5GHz SoC PLL with 15dB built-in supply noise rejection and self-bandwidth control in 14nm CMOS", *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 330–331, January 2016.

17. Kim, W., J. Park, J. Kim, T. Kim, H. Park and D. Jeong, "A 0.032mm2 3.1mW synthesized pixel clock generator with 30psrms integrated jitter and 10-to-630MHz DCO tuning range", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, pp. 250–251, February 2013.

18. Deng, W., D. Yang, T. Ueno, T. Siriburanon, S. Kondo, K. Okada and A. Matsuzawa, "15.1 A 0.0066mm2 780 uW fully synthesizable PLL with a current-output DAC and an interpolative phase-coupled oscillator using edge-injection technique", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pp. 266–267, February 2014.

19. Kim, H., J. Sang, H. Kim, Y. Jo, T. Kim, H. Park and S. H. Cho, "14.4 A 5GHz -95dBc-reference-Spur 9.5mW digital fractional-N PLL using reference-multiplied time-to-digital converter and reference-spur cancellation in 65nm CMOS", *Solid-State Circuits Conference - (ISSCC), 2015 IEEE International*, pp. 1–3, February 2015.

20. Ahmad, F., G. Unruh, A. Iyer, P. E. Su, S. Abdalla, B. Shen, M. Chambers and

I. Fujimori, “19.1 A 0.5-to-9.5GHz 1.2ps-lock-time fractional-N DPLL with 1.25 UI period jitter in 16nm CMOS for dynamic frequency and core-count scaling in SoC”, *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 324–325, January 2016.

21. Yeh, C. W., C. E. Hsieh and S. I. Liu, “19.5 A 3.2GHz digital phase-locked loop with background supply-noise cancellation”, *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 332–333, January 2016.

22. Dalt, N. D., P. Pridnig and W. Grollitsch, “An all-digital PLL using random modulation for SSC generation in 65nm CMOS”, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, pp. 252–253, February 2013.

23. Liu, J., T. K. Jang, Y. Lee, J. Shin, S. Lee, T. Kim, J. Park and H. Park, “15.2 A 0.012mm2 3.1mW bang-bang digital fractional-N PLL with a power-supply-noise cancellation technique and a walking-one-phase-selection fractional frequency divider”, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pp. 268–269, February 2014.

24. Tsai, T. H., M. S. Yuan, C. H. Chang, C. C. Liao, C. C. Li and R. B. Staszewski, “14.5 A 1.22ps integrated-jitter 0.25-to-4GHz fractional-N ADPLL in 16nm FinFET CM0S”, *Solid- State Circuits Conference - (ISSCC), 2015 IEEE International*, pp. 1–3, February 2015.

25. Kundu, S., B. Kim and C. H. Kim, “19.2 A 0.2-to-1.45GHz subsampling fractional-N all-digital MDLL with zero-offset aperture PD-based spur cancellation and in-situ timing mismatch detection”, *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 326–327, January 2016.

26. Sai, A., S. Kondo, T. T. Ta, H. Okuni, M. Furuta and T. Itakura, “19.7 A 65nm CMOS ADPLL with 360uW 1.6ps-INL SS-ADC-based period-detection-free TDC”, *2016 IEEE International Solid-State Circuits Conference (ISSCC)*,

pp. 336–337, January 2016.

27. Jang, T. K., X. Nan, F. Liu, J. Shin, H. Ryu, J. Kim, T. Kim, J. Park and H. Park, "A 0.026mm2 5.3mW 32-to-2000MHz digital fractional-N phase locked-loop using a phase-interpolating phase-to-digital converter", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, pp. 254–255, February 2013.

28. Huang, Y. C., C. F. Liang, H. S. Huang and P. Y. Wang, "15.3 A 2.4GHz ADPLL with digital-regulated supply-noise-insensitive and temperature-self-compensated ring DCO", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pp. 270–271, February 2014.

29. Song, M., T. Kim, J. Kim, W. Kim, S. J. Kim and H. Park, "14.8 A 0.009mm2 2.06mW 32-to-2000MHz 2nd-order analogous bang-bang digital PLL with feed-forward delay-locked and phase-locked operations in 14nm FinFET technology", *Solid- State Circuits Conference - (ISSCC), 2015 IEEE International*, pp. 1–3, February 2015.

30. Kim, H., Y. Kim, T. Kim, H. Park and S. Cho, "19.3 A 2.4GHz 1.5mW digital MDLL using pulse-width comparator and double injection technique in 28nm CMOS", *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 328–329, January 2016.

31. Zhu, J., R. K. Nandwana, G. Shu, A. Elkholy, S. J. Kim and P. K. Hanumolu, "19.8 A 0.0021mm2 1.82mW 2.2GHz PLL using time-based integral control in 65nm CMOS", *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 338–340, January 2016.

32. Zhang, X. and A. B. Apsel, "A low variation GHz ring oscillator with addition-based current source", *Solid State Device Research Conference, 2009. ESSDERC '09. Proceedings of the European*, pp. 233–236, September 2009.

33. Chung, Y. M. and C. L. Wei, "An all-digital phase-locked loop for digital power management integrated chips", *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 2413–2416, May 2009.

34. Vengattaramane, K., J. Craninckx and M. Steyaert, "Analysis of fractional spur reduction using noise cancellation in digital-PLL", *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 2397–2400, May 2009.

35. Tangudu, J., S. Gunturi, S. Jalan, J. Janardhanan, R. Ganesan, D. Sahu, K. Waheed, J. Wallberg and R. B. Staszewski, "Quantization noise improvement of Time to Digital converter (TDC) for ADPLL", *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 1020–1023, May 2009.

36. Shin, D., J. Koo, W. J. Yun, Y. Choi and C. Kim, "A fast-lock synchronous multi-phase clock generator based on a time-to-digital converter", *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 1–4, May 2009.

37. Samori, C., M. Zanuso, S. Levantino and A. L. Lacaita, "Multipath adaptive cancellation of divider non-linearity in fractional-N PLLs", *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pp. 418–421, May 2011.

38. Brandonisio, F. and F. Maloberti, "An all-digital PLL with a first order noise shaping Time-to-Digital Converter", *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 241–244, May 2010.

39. Brandonisio, F. and M. P. Kennedy, "First order noise shaping in all digital PLLs", *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pp. 161–164, May 2011.

40. Ouda, M., E. Hegazi and H. F. Ragai, "Digital enhancement of frequency synthesizers", *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 973–976, May 2010.

41. Staszewski, R. B., "All-digital RF frequency modulation", *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pp. 426–429, May 2011.

42. Javidan, M., E. Zianbetov, F. Anceau, D. Galayko, A. Korniienko, E. Colinet, G. Scorletti, J. M. Akré and J. Juillard, "All-digital PLL array provides reliable distributed clock for SOCs", *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pp. 2589–2592, May 2011.

43. Klumperink, E., R. Dutta, Z. Ru, B. Nauta and X. Gao, "Jitter-Power minimization of digital frequency synthesis architectures", *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pp. 165–168, May 2011.

44. Pellerano, S., P. Madoglio and Y. Palaskas, "A 4.75GHz fractional frequency divider with digital spur calibration in 45nm CMOS", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 226–227,227a, February 2009.

45. Shanan, H., G. Retz, K. Mulvaney and P. Quinlan, "A 2.4GHz 2Mb/s versatile PLL-based transmitter using digital pre-emphasis and auto calibration in 0.18 um CMOS for WPAN", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 420–421,421a, February 2009.

46. Ali, T. A., A. A. Hafez, R. Drost, R. Ho and C. K. K. Yang, "A 4.6GHz MDLL with -46dBc reference spur and aperture position tuning", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 466–468, February 2011.

47. Jang, S., H. Song, S. Ye and D. K. Jeong, "A 13.8mW 3.0Gb/s clock-embedded video interface with DLL-based data-recovery circuit", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 450–452, February 2011.

48. Wu, C. P., S. S. Wang, H. W. Tsao and J. Wu, "A 300KHz bandwidth 3.9GHz

0.18 um CMOS fractional-N synthesizer with 13dB broadband phase noise reduction", *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, pp. 451–454, September 2011.

49. Weltin-Wu, C., E. Temporiti, D. Baldi, M. Cusmai and F. Svelto, "A 3.5GHz wideband ADPLL with fractional spur suppression through TDC dithering and feedforward compensation", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 468–469, February 2010.

50. Chen, M. S. W., D. Su and S. Mehta, "A Calibration-Free 800 MHz Fractional-N Digital PLL With Embedded TDC", *IEEE Journal of Solid-State Circuits*, Vol. 45, No. 12, pp. 2819–2827, December 2010.

51. Tokairin, T., M. Okada, M. Kitsunezuka, T. Maeda and M. Fukaishi, "A 2.1-to-2.8GHz all-digital frequency synthesizer with a time-windowed TDC", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 470–471, February 2010.

52. Zanuso, M., S. Levantino, C. Samori and A. Lacaita, "A 3MHz-BW 3.6GHz digital fractional-N PLL with sub-gate-delay TDC, phase-interpolation divider, and digital mismatch cancellation", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 476–477, February 2010.

53. Grollitsch, W., R. Nonis and N. D. Dalt, "A 1.4psrms-period-jitter TDC-less fractional-N digital PLL with digitally controlled ring oscillator in 65nm CMOS", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 478–479, February 2010.

54. Gao, X., E. A. M. Klumperink, G. Socci, M. Bohsali and B. Nauta, "Spur-reduction techniques for PLLs using sub-sampling phase detection", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 474–475, February 2010.

55. Borremans, J., K. Vengattaramane, V. Giannini and J. Craninckx, "A 86MHz-to-12GHz digital-intensive phase-modulated fractional-N PLL using a 15pJ/Shot 5ps TDC in 40nm digital CMOS", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 480–481, February 2010.

56. Kondou, M., A. Matsuda, H. Yamazaki and O. Kobayashi, "A 0.3mm2 90-to-770MHz fractional-N Synthesizer for a digital TV tuner", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 248–249, February 2010.

57. Pavlovic, N. and J. Bergervoet, "A 5.3GHz digital-to-time-converter-based fractional-N all-digital PLL", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 54–56, February 2011.

58. Tasca, D., M. Zanuso, G. Marzin, S. Levantino, C. Samori and A. L. Lacaita, "A 2.9-to-4.0GHz fractional-N digital PLL with bang-bang phase detector and 560fs-rms integrated jitter at 4.5mW power", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 88–90, February 2011.

59. Liang, C. F. and K. J. Hsiao, "An injection-locked ring PLL with self-aligned injection window", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 90–92, February 2011.

60. Elshazly, A., R. Inti, W. Yin, B. Young and P. K. Hanumolu, "A 0.4-to-3GHz digital PLL with supply-noise cancellation using deterministic background calibration", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 92–94, February 2011.

61. Jee, D. W., Y. Suh, H. J. Park and J. Y. Sim, "A 0.1-fref BW 1GHz fractional-N PLL with FIR-embedded phase-interpolator-based noise filtering", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 94–96, February 2011.

62. Lee, H. J., A. M. Kern, S. Hyvonen and I. A. Young, "A scalable sub-1.2mW 300MHz-to-1.5GHz host-clock PLL for system-on-chip in 32nm CMOS", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 96–97, February 2011.

63. Sai, A., T. Yamaji and T. Itakura, "A 570fsrms integrated-jitter ring-VCO-based 1.21GHz PLL with hybrid loop", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 98–100, February 2011.

64. Takinami, K., R. Strandberg, P. C. P. Liang, G. L. G. de Mercey, T. Wong and M. Hassibi, "A rotary-traveling-wave-oscillator-based all-digital PLL with a 32-phase embedded phase-to-digital converter in 65nm CMOS", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pp. 100–102, February 2011.

65. Lee, J., H. Wang, W.-T. Chen and Y.-P. Lee, "Subharmonically injection-locked PLLs for ultra-low-noise clock generation", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 92–93,93a, February 2009.

66. Rylyakov, A., J. Tierno, H. Ainspan, J. O. Plouchart, J. Bulzacchelli, Z. T. Deniz and D. Friedman, "Bang-bang digital PLLs at 11 and 20GHz with sub-200fs integrated jitter for high-speed serial communication applications", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 94–95,95a, February 2009.

67. hyoun Kim, K., D. M. Dreps, F. D. Ferraiolo, P. W. Coteus, S. Kim, S. V. Rylov and D. J. Friedman, "A 5.4mW 0.0035mm2 0.48psrms-jitter 0.8-to-5GHz non-PLL/DLL all-digital phase generator/rotator in 45nm SOI CMOS", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 98–99,99a, February 2009.

68. Tsai, K. H. and S. I. Liu, "A 43.7mW 96GHz PLL in 65nm CMOS", *Solid-*

State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International, pp. 276–277,277a, February 2009.

69. Chien, T. H., C. S. Lin, Y. Z. Juang, C. M. Huang and C. L. Wey, "An edge-missing compensator for fast-settling wide-locking-range PLLs", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 394–395,395a, February 2009.

70. Yu, X., W. Rhee, Z. Wang, J. B. Lee and C. Kim, "A 0.4-to-1.6GHz low-OSR DLL with self-referenced multiphase generation", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 398–399,399a, February 2009.

71. Hedayati, H., B. Bakkaloglu and W. Khalil, "A 1MHz-bandwidth type-I fractional-N synthesizer for WiMAX applications", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 390–391,391a, February 2009.

72. Gao, X., E. A. M. Klumperink, M. Bohsali and B. Nauta, "A 2.2GHz 7.6mW sub-sampling PLL with -126dBc/Hz in-band phase noise and 0.15psrms jitter in 0.18 um CMOS", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 392–393,393a, February 2009.

73. Lu, L., Z. Gong, Y. Liao, H. Min and Z. Tang, "A 975-to-1960MHz fast-locking fractional-N synthesizer with adaptive bandwidth control and 4/4.5 prescaler for digital TV tuners", *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 396–397,397a, February 2009.

74. Perrott, M. H., S. Pamarti, E. Hoffman, F. S. Lee, S. Mukherjee, C. K. Lee, V. Tsinker, S. Perumal, B. Soto, N. Arumugam and B. W. Garlepp, "A low-area switched-resistor loop-filter technique for fractional-N synthesizers applied to a MEMS-based programmable oscillator", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 244–245, February

2010.

75. Lee, S.-K., Y.-H. Seo, Y. Suh, H.-J. Park and J.-Y. Sim, "A 1GHz ADPLL with a 1.25ps minimum-resolution sub-exponent TDC in 0.18 um CMOS", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 482–483, February 2010.

76. Mandai, S., T. Iizuka, T. Nakura, M. Ikeda and K. Asada, "Time-to-digital converter based on time difference amplifier with non-linearity calibration", *ESSCIRC, 2010 Proceedings of the*, pp. 266–269, September 2010.

77. Zanuso, M., S. Levantino, A. Puggelli, C. Samori and A. L. Lacaita, "Time-to-digital converter with 3-ps resolution and digital linearization algorithm", *ESSCIRC, 2010 Proceedings of the*, pp. 262–265, September 2010.

78. Mandai, S. and E. Charbon, "A 128-channel, 9ps column-parallel two-stage TDC based on time difference amplification for time-resolved imaging", *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, pp. 119–122, September 2011.

79. Lu, P., P. Andreani and A. Liscidini, "A 90nm CMOS gated-ring-oscillator-based Vernier time-to-digital converter for DPLLs", *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, pp. 459–462, September 2011.

80. Devita, G., A. C. W. Wong, N. Kasparidis, P. Corbishley, A. Burdett and P. Paddan, "A 0.9mW PLL integrated in an ultra-low-power SoC for WPAN and WBAN applications", *ESSCIRC, 2010 Proceedings of the*, pp. 158–161, September 2010.

81. Chao, T. S., Y. L. Lo, W. B. Yang and K. H. Cheng, "Designing ultra-low voltage PLL Using a bulk-driven technique", *ESSCIRC, 2009. ESSCIRC '09. Proceedings of*, pp. 388–391, September 2009.

82. Drago, S., D. Leenaerts, B. Nauta, F. Sebastiano, K. Makinwa and L. Breems, "A 200 A duty-cycled PLL for wireless sensor nodes", *ESSCIRC, 2009. ESSCIRC*

'09. Proceedings of, pp. 132–135, September 2009.

83. Wang, P. Y. and C. H. Fu, "All digital modulation bandwidth extension technique for narrow bandwidth analog fractional-N PLL", *ESSCIRC, 2010 Proceedings of the*, pp. 270–273, September 2010.

84. von Bueren, G., D. Barras, H. Jaeckel, A. Huber, C. Kromer and M. Kossel, "Design and phase noise analysis of a multiphase 6 to 11 GHz PLL", *ESSCIRC, 2009. ESSCIRC '09. Proceedings of*, pp. 384–387, September 2009.

85. Kobayashi, Y., S. Amakawa, N. Ishihara and K. Masu, "A low-phase-noise injection-locked differential ring-VCO with half-integral subharmonic locking in 0.18 um CMOS", *ESSCIRC, 2009. ESSCIRC '09. Proceedings of*, pp. 440–443, September 2009.

86. Murphy, D., Q. J. Gu, Y. C. Wu, H. Y. Jian, Z. Xu, A. Tang, F. Wang and M. C. F. Chang, "A Low Phase Noise, Wideband and Compact CMOS PLL for Use in a Heterodyne 802.15.3c Transceiver", *IEEE Journal of Solid-State Circuits*, Vol. 46, No. 7, pp. 1606–1617, July 2011.

87. Huang, Y. C., C. F. Liang, H. S. Huang and P. Y. Wang, "15.3 A 2.4GHz ADPLL with digital-regulated supply-noise-insensitive and temperature-self-compensated ring DCO", *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 270–271, February 2014.

88. August, N., H. J. Lee, M. Vandepas and R. Parker, "A TDC-less ADPLL with 200-to-3200MHz range and 3mW power dissipation for mobile SoC clocking in 22nm CMOS", *2012 IEEE International Solid-State Circuits Conference*, pp. 246–248, February 2012.

89. Song, M., T. Kim, J. Kim, W. Kim, S. J. Kim and H. Park, "14.8 A 0.009mm2 2.06mW 32-to-2000MHz 2nd-order analogous bang-bang digital PLL with feed-forward delay-locked and phase-locked operations in 14nm FinFET technology",

2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers, pp. 1–3, February 2015.

90. Liu, J., T. K. Jang, Y. Lee, J. Shin, S. Lee, T. Kim, J. Park and H. Park, "15.2 A 0.012mm2 3.1mW bang-bang digital fractional-N PLL with a power-supply-noise cancellation technique and a walking-one-phase-selection fractional frequency divider", *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 268–269, February 2014.

91. Tsai, T. H., M. S. Yuan, C. H. Chang, C. C. Liao, C. C. Li and R. B. Staszewski, "14.5 A 1.22ps integrated-jitter 0.25-to-4GHz fractional-N ADPLL in 16nm FinFET CM0S", *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, pp. 1–3, February 2015.

92. Kong, L. and B. Razavi, "25.7 A 2.4GHz 4mW inductorless RF synthesizer", *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, pp. 1–3, February 2015.

93. Balcioglu, Y. and G. Dundar, "A synthesizable digitally controlled oscillator with only standard cells: A 810 to 1400 MHz digital back-end design flow compatible design with PVT calibration", *Electronics and Telecommunications (ISETC), 2014 11th International Symposium on*, pp. 1–4, November 2014.

94. Balcioglu, Y. and G. Dundar, "Improved standard cell synthesizable Digitally Controlled Oscillator", *Signal Processing and Communications Applications Conference (SIU), 2015 23th*, pp. 355–358, May 2015.

95. Wu, C.-T., W.-C. Shen, W. Wang and A.-Y. Wu, "A Two-Cycle Lock-In Time ADPLL Design Based on a Frequency Estimation Algorithm", *Circuits and Systems II: Express Briefs, IEEE Transactions on*, Vol. 57, No. 6, pp. 430–434, June 2010.

96. Guler, U. and G. Dundar, "Modeling CMOS Ring Oscillator Performance as a

Randomness Source", *Circuits and Systems I: Regular Papers, IEEE Transactions on*, Vol. 61, No. 3, pp. 712–724, March 2014.

97. Vengattaramane, K., J. Borremans, M. Steyaert and J. Craninckx, "A standard cell based all-digital Time-to-Digital Converter with reconfigurable resolution and on-line background calibration", *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, pp. 275–278, September 2011.

98. Mandai, S. and E. Charbon, "A 128-Channel, 8.9-ps LSB, Column-Parallel Two-Stage TDC Based on Time Difference Amplification for Time-Resolved Imaging", *IEEE Transactions on Nuclear Science*, Vol. 59, No. 5, pp. 2463–2470, October 2012.

99. Straayer, M. and M. Perrott, "A Multi-Path Gated Ring Oscillator TDC With First-Order Noise Shaping", *Solid-State Circuits, IEEE Journal of*, Vol. 44, No. 4, pp. 1089–1098, April 2009.

100. Temporiti, E., C. Weltin-Wu, D. Baldi, R. Tonietto and F. Svelto, "A 3 GHz Fractional All-Digital PLL With a 1.8 MHz Bandwidth Implementing Spur Reduction Techniques", *Solid-State Circuits, IEEE Journal of*, Vol. 44, No. 3, pp. 824–834, March 2009.

101. Lee, M. and A. A. Abidi, "A 9 b, 1.25 ps Resolution Coarse Fine Time-to-Digital Converter in 90 nm CMOS that Amplifies a Time Residue", *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 4, pp. 769–777, April 2008.

102. Sai, A., S. Kondo, T. T. Ta, H. Okuni, M. Furuta and T. Itakura, "19.7 A 65nm CMOS ADPLL with 360uW 1.6ps-INL SS-ADC-based period-detection-free TDC", *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 336–337, January 2016.

103. Hsu, C. M., M. Z. Straayer and M. H. Perrott, "A Low-Noise Wide-BW 3.6-GHz Digital Delta Sigma Fractional-N Frequency Synthesizer With a Noise-Shaping

Time-to-Digital Converter and Quantization Noise Cancellation", *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 12, pp. 2776–2786, December 2008.

104. Ye, Z. and M. Kennedy, "Reduced Complexity MASH Delta Sigma Modulator", *Circuits and Systems II: Express Briefs, IEEE Transactions on*, Vol. 54, No. 8, pp. 725–729, August 2007.

105. Chen, M. S. W., D. Su and S. Mehta, "A calibration-free 800MHz fractional-N digital PLL with embedded TDC", *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 472–473, February 2010.

106. Balcioglu, Y., *CellpII download website*, 2016, `http://sites.google.com/yalcinbalcioglu`, May 2016.

APPENDIX A: CELLPLL USER MANUAL

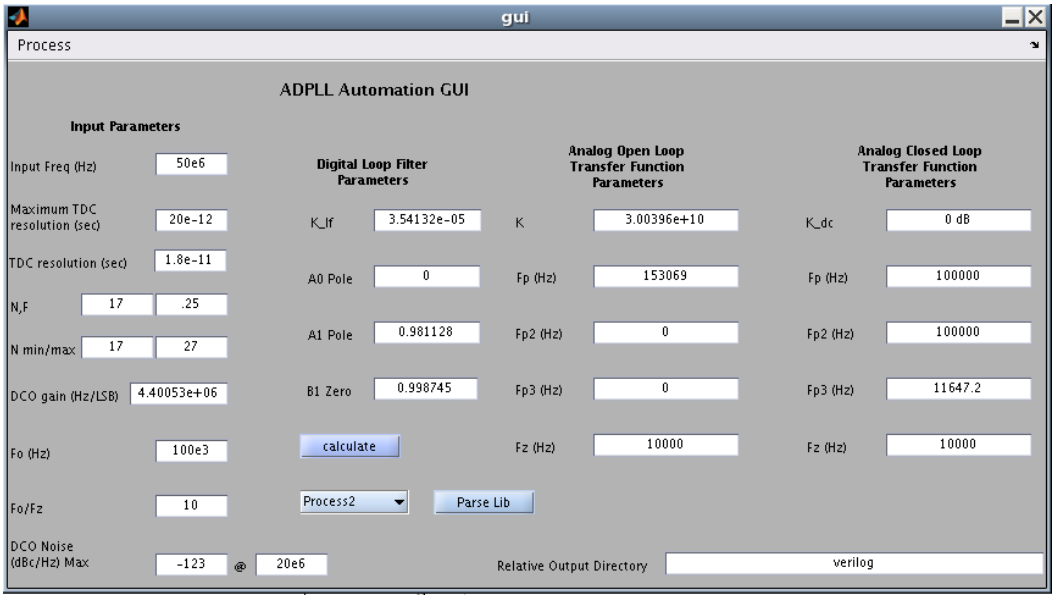


Figure A.1. Graphical User interface

The tool runs within MATLAB. GUI is started by running the "gui.m". Only for the first time use of each standard cell library, the parsing has to be completed. In order to select the desired library container for the library to be parsed, select the desired process container from process drop-down menu. Prepare the library template ".mat" file as explained in Appendix B. From the "Process" menu select the desired library template ".mat" file and click parse library button. This operation takes a considerable amount of time and the progress can be tracked from the MATLAB command line.

After the library is parsed, the user can fill in all the desired parameters, set the output folder to place the generated files and click generate button to run the tool. The possible parameters in the GUI are explained below:

- Input Frequency: Reference input of the PLL. Together with feedback divider range, determines the output frequency band of the ADPLL.
- MAX TDC resolution: Selects the desired resolution of the time-to-digital con-

verter. During TDC HLS, tool estimates if it can satisfy this requirement and gives an error if desired specification cannot be met.

- TDC resolution: Reports the estimated TDC resolution that the system will have after HLS for TDC is complete.
- $N.F$: Selects the fractional feedback divider value that is used for phase noise estimation.
- N_{min}, N_{max} : Sets the range for feedback divider value which is used for designing and verifying DCO ranges together with the input frequency. During DCO HLS, the available DCO tuning range is calculated for all corners and if the desired PLL output frequency range cannot be met, an error is asserted.
- DCO gain: After the HLS for DCO is complete, the DCO frequency steps are estimated and reported from this GUI item.
- F_0 selects the desired PLL bandwidth and F_0/F_z ratio selects the desired zero location in the closed loop transfer function with respect to the loop BW.
- DCO noise max: Selects the maximum allowed phase noise for the DCO referred noise. During DCO HLS, strengths are adjusted to meet this specification through estimation equations and an error is asserted if the specification cannot be met.
- Relative output directory: Selects the directory in which the generated ADPLL design, scripts and behavioral models will be written when the generation is triggered.

The other GUI items report the calculated open and closed loop transfer function gain, pole and zero calculation results specifically generated for specified ADPLL parameters. When the tool finishes the design, analysis, and high-level synthesis steps, several figures are printed. The synthesis scripts and design's Verilog codes are printed together with behavioral test-benches to the output folder. Figures include pole/zero locations, closed/open loop transfer functions, and phase noise profile results. After the run is complete, the process, the output folder, and the specifications can be modified as desired and the tools can be rerun. The source codes for this tool has been placed in [106].

APPENDIX B: STANDARD CELL LIBRARY PARSER

CellPLL tool provides a template for the ".mat" configuration file that the user needs to fill for each standard cell library. This configuration file contains entries for

(i) Standard cell library path, type, and corner
(ii) Some gates names, port names and possible strengths for a limited number of gate types that are used in ring oscillators

When the configuration file for technology library is ready, an only one time parse operation for each library needs to be performed as follows:

- Using the File $->$ Library drop down menu the user selects the configuration file
- Selects the desired container name for the process from the process drop down menu
- Clicks $ParseLib$ to let the tool analyze the library for the cells

During library parse, tool finds all the gates that it will use as gate level instances in the design and extracts their propagation delay, rise/fall time, pin cap, power tables for each strength and input/output pin combinations. When the parser completes, it saves all of the parsed data into a database that is loaded automatically in the future runs when this library is selected. A previously saved parsed library container can also be loaded later from an external file path using the "Process" menu.

This information is used to perform static timing analysis. STA results are used for strength adjustment during DCO and TDC's ring oscillator center frequency estimation and open loop phase noise estimation of the DCO.

APPENDIX C: STATIC TIMING ANALYSIS - PROPAGATION DELAY

As illustrated in Figure C.1, the switching of input signal A changes the state of Z; similar methods can be applied to other logic cells by applying proper input patterns to toggle the measured output. That is, by setting the proper values on other input pins, the output state becomes dependent on the trigger input only. Therefore, the propagation delay is a function of various inputs and may be expressed in a table.

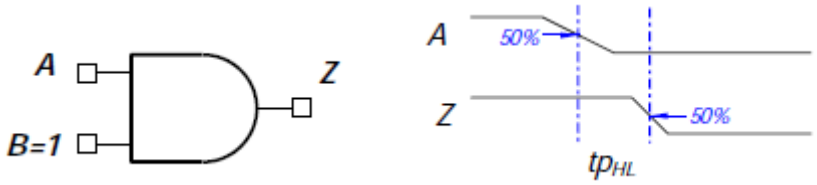


Figure C.1. AND gate propagation delay example

For each output pin, propagation delays $t_{p_{LH}}$ and $t_{p_{HL}}$ represent the state change delay from low-to-high and from high-to-low transitions. Propagation delay is measured from the 50% point of the input waveform to the 50% point of the output waveform as shown in Figure C.1.

Table C.1. Standard cell datasheet delay characterization example

Cell Name	**Path**	**Parameter**	**Group 1** $(< 0.00162)\ pf$	**Group 2** $(0.00162 - 0.03024)\ pf$	**Group 3** $(> 0.03024)\ pf$
BUFFD1	I to Z	$t_{p_{LH}}$	$0.0203 + 3.0211 * C_{load}$	$0.021 + 3.2856 * C_{load}$	$0.0213 + 3.2689 * C_{load}$
BUFFD1	I to Z	$t_{p_{HL}}$	$0.0244 + 3.0105 * C_{load}$	$0.0260 + 2.1197 * C_{load}$	$0.0279 + 2.0357 * C_{load}$

The propagation delay is a non-linear function of both the input slew rate and output loading. For ease of use during manual calculations, standard cell library expresses propagation delay as a simple linear equation based on output loading. In other words, this simple first order fitting provides only reference information about

the timing provided for each cell. For a more detailed and less simplified model, the design kits provide a two-dimensional, look-up timing table for each cell. The equation in Eq. C.1 models the propagation delay for each cell. Tables similar to the one in Table C.1 are used to make readable content for datasheets.

$$T_{typical} = T_{intrinsic} + F * C_{load} \tag{C.1}$$

$$T_{typical} = propagation\ delay\ at\ typical\ case\ (1.0V,\ 25°C)\ (ns)$$

$$T_{intrinsic} = intrinsic\ delay\ of\ each\ cell\ (ns)$$

$$F = load\ delay\ factor\ (ns/pF)$$

$$C_{load} = total\ output\ load\ capacitance\ (pF)$$

However, 2D tables that have rise/fall time and loading as two axes of the 2D table are used in real life similar to the one given in Figure C.2. Therefore, in HLS algorithms of CellPLL, these 2D mapping tables are utilized.

```
pin(Z) {
 direction : output;
 power_down_function : "!VDD + VSS";
 function : "((A1 A2) A3)";
 related_ground_pin : VSS;
 related_power_pin : VDD;
 max_capacitance : 0.03038;
 timing () {
  related_pin : "A1";
  timing_sense : positive_unate;
  timing_type : combinational;
  cell_rise (delay_template_7x7_0) {
   index_1 ("0.0024, 0.0079, 0.0189, 0.0408, 0.0847, 0.1724, 0.3478");
   index_2 ("0.00045, 0.00093, 0.00188, 0.00378, 0.00758, 0.01518, 0.03038");
   values ( \
    "0.03095, 0.03468, 0.04122, 0.05289, 0.07476, 0.1177, 0.2035", \
    "0.03163, 0.03537, 0.04189, 0.05355, 0.07544, 0.1185, 0.2043", \
    "0.03321, 0.03695, 0.04346, 0.05511, 0.077, 0.12, 0.2058", \
    "0.03695, 0.04064, 0.0471, 0.05871, 0.08057, 0.1236, 0.2095", \
    "0.04273, 0.0465, 0.05308, 0.06479, 0.08665, 0.1296, 0.2152", \
    "0.04991, 0.05386, 0.0607, 0.07258, 0.0945, 0.1375, 0.223", \
    "0.05847, 0.06281, 0.07031, 0.08299, 0.1055, 0.1486, 0.2343" \
   );
```

Figure C.2. Standard cell library delay characterization example