The Dissertation Committee for Giri NK Rangan Certifies that this is the approved version of the following dissertation:

A Fractional N Frequency Synthesizer for an Adaptive Network Backplane Serial Communication System

Committee:

Earl E. Swartzlander, Jr., Supervisor

Jacob A. Abraham

Anthony P. Ambler

Donald S. Fussell

Eric J. Swanson

A Fractional N Frequency Synthesizer for an Adaptive Network Backplane Serial Communication System

by

Giri NK Rangan, BE, MSEE

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December, 2005

Dedication

To Amma and Anna for teaching me to value education

And to Anu for ensuring that I have the patience and the perseverance

to take up this task and complete it.

Acknowledgements

I would like to express my thanks to Prof. Earl E. Swartzlander, Jr., for his invaluable encouragement, support, understanding and guidance. Being a part time student implies a relatively drawn out doctoral program and to have an advisor that understands the various external factors that can potentially derail the program is very crucial. I am very happy and fortunate that I had one in Prof. Swartzlander.

I also wish to thank Mr. Eric Swanson at this time. I learned a lot about system modeling and simulation from his great course *Introduction to System Design and Modeling*, which dealt with circuit design issues and modeling of the same in a high level behavioral simulation. I got a great headstart in refining the scope of my dissertation based on this course. I also learned the right way to approach any circuit and system design project from the invaluable information that Mr. Swanson provided in this course.

I express my thanks and gratitude to Prof. Anthony Ambler for agreeing to be the chairman of my committee. I also wish to thank Prof. Jacob Abraham and Prof. Donald Fussell for graciously agreeing to be a part of my committee.

I want to acknowledge the inputs of Mr. Jules J Jelinek and Dr. Kishore Kota that helped greatly in addressing some of the architectural issues. Their vast experience in the field of Frequency Synthesizers, PLLs and communications was a great resource for me.

I also want to thank my parents, my sisters, my parents-in-law and all my friends in Austin for their support and encouragement. Lastly and most importantly, I would like to thank my wife, Dr. Anu Rangan and my son Raghav for their love, affection and support. This work would not have been possible without their understanding and encouragement.

A Fractional N Frequency Synthesizer for an Adaptive Network Backplane Serial Communication System

Publication No._____

Giri NK Rangan, Ph.D.

The University of Texas at Austin, 2005

Supervisor: Earl E. Swartzlander, Jr.

An architecture and design of a Phase Locked Loop based frequency synthesizer is developed in this dissertation. Using multiple phases generated by a ring oscillator, this synthesizer is able to generate non-integer multiples of the incoming, high quality clock signal. The design is done for a nominal target frequency of 3.125 GHz for application in a serial communication system such as a network backplane. Using a fully differential design, the architecture is able to achieve the stringent timing jitter requirements of a network backplane system.

Advancements in the content and the coverage of the Internet have tremendously increased the need for high speed data transport over very long and very short distances. The long distance bandwidth and speed requirements have been addressed by the use of optical links. The shorter distances, such as serial communications in a network backplane are still in the realm of copper lines drawn on printed circuit boards. Thus the medium of communication places design constraints on the electronic devices operating on either side of the medium. As symbol frequencies approach 3.125 GHz and beyond,

architectural modifications must be made to alleviate the channel effects. This dissertation presents a phase locked loop for a network backplane system application where the two transceivers can communicate with each other to determine a particular line code that they will use for the most optimal communication between them. The selection of the line code determines the symbol rate and in turn the transmit clock frequency. A particular line code may be chosen such that a non-integer multiple of a low frequency input clock is required. The incoming clock is usually fixed at standard frequencies like 312.5 or 625 MHz to operate the parallel data path. It would then be beneficial to have a fractional-N frequency synthesizer which can generate the necessary fractional frequency multiples.

The synthesizer presented in this research work is designed in a standard 0.13 μm CMOS technology with a 1.5 V power supply. It dissipates 112 mW of power and occupies an estimated silicon area of 0.2 sq. mm. The nominal peak to peak jitter of this design is approximately 43.7 ps and the maximum peak to peak jitter is 48 ps.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

This research addresses the design of a fractional N clock synthesizer that can generate non-integer multiples of an incoming clock frequency with low timing jitter. In a high speed serial communication environment like a network backplane, the characteristics of the communication channel between two transceivers may not be determined until the time of installation and may need to be changed during the life of the system. As a result, it is desirable to be able to adapt the data coding schemes of the transceivers based on the actual conditions in the network backplane. The choice of a particular line code determines the rate of the serial communication and is a major factor in the transceiver design. Therefore, in many situations a serial transmit clock is required that can be set to a wide variety of non-integer multiples of the on-board crystal oscillator.

Commercially available network backplane transceivers offer data rates from 2.5 GBPS to 10 GBPS. The higher speed versions employ multi-level coding schemes to reduce the symbol rate or the serial communication rate. This dissertation addresses design and development of the transmit clock synthesizer for a transceiver operating at a nominal data rate of 5 GBPS. Due to the line coding employed, the symbol rate may be as high as 6.25 GBPS. As a result, the frequency synthesizer developed in this work operates at a nominal center frequency of 3.125 GHz. To achieve this, an input crystal oscillator frequency of 312.5 MHz is used. The nominal multiply ratio is 10 with a targeted power consumption of 75 mW. For the most common configurations of quad transceivers using a single synthesizer, the per port power consumption is 18.75 mW. The targeted timing jitter is less than 50 ps peak to peak and less than 3.5 ps rms.

A new Phase Locked Loop (PLL) architecture for generating non-integer multiples of the input clock has been developed. This design is based on the observation that in a synchronous Phase Frequency Detector (PFD) inside the PLL, the divided version of the output clock need not have a 50% duty cycle. The PFD operates only on the rising edges of the reference clock and the feedback clock [1]. Thus, the feedback clock only needs to have the correct frequency and phase. In this design, the feedback clock is actually a pulse train whose frequency and phase is locked to the reference clock. This allows for signal manipulation such that non-integer frequency multiplication is achieved.

1-1. FREQUENCY SYNTHESIZERS

Frequency synthesizers are common building blocks of most communication systems. Most often, a clock synthesizer or a frequency synthesizer is a PLL based system that locks the phase and a divided version of the output frequency to an incoming clock source. In most applications, the incoming clock source is a crystal oscillator. Crystal oscillators, because of their high Q factors, can generate single frequency signals of very high spectral purity. However, due to system and circuit issues, the output of the PLL will not have the same quality as that of the crystal. The extent of reduction in the quality of the PLL output is quantified in terms of *timing jitter* or *phase noise*. These two metrics are used to measure the performance of a PLL and compare different architectures and designs. Although jitter and phase noise are related to each other, one or the other is mostly specified for a synthesizer based on the application. For example, in wireless communications, the frequency synthesizer functions as a Local Oscillator (LO) to do the mixing of the incoming RF signal down to an IF band. Here, the LO output

needs to be a sinusoid such that the mixing operation generates only sum and difference frequencies. In such applications, the performance of the synthesizer is measured in terms of its *phase noise*. On the other hand, in wire line applications such as Ethernet, the clock source is usually a square wave and its *timing jitter* is the most critical performance parameter. Timing jitter and the phase noise are related to each other and knowing one, the other can be estimated. This dissertation is focused on generating a high frequency clock with low timing jitter.

When a PLL is required to generate an output frequency that is not an integer multiple of the incoming clock, most architectures achieve this by modulating the divide factor of the feedback divider such that on the *average*, the PLL output has the desired frequency. However, the cycle to cycle behavior of such architectures may contain large timing variations that can lead to significant timing jitter. Delta Sigma modulators are also used to randomly select the divide factor on a cycle to cycle basis to achieve fractional multiplication. In such architectures, the loop bandwidth of the PLL should be set low enough to filter the out of band quantization noise produced by the Delta Sigma Modulator. A steady state small signal analysis of the PLL, presented later in this dissertation, shows that any noise introduced at the VCO, experiences a highpass function to the output. Hence, a loop with a relatively large bandwidth is desirable such that the VCO noise can get filtered out. This dissertation presents an architecture where the loop filter bandwidth can be set relatively high.

1-2. ORGANIZATION

This dissertation is organized as follows. Chapter 2 explains the components of a serial communications link and the performance metrics of such links. The concept of Bit Error Rate (BER) is introduced and the effects of amplitude and phase distortion as

captured on an eye diagram are explained. Timing jitter, its effects on system performance, its classification and specification are discussed. The chapter then presents network backplanes and their channel characteristics, line coding and the resulting data/symbol rate ratios. An adaptive network backplane is introduced. In such a system, the serial communication rate between two transceivers can be modified based on the channel conditions. This is achieved by choosing one out of several different line codes. The resulting transmit clock frequency requirements are summarized in this chapter.

Chapter 3 presents a review of frequency synthesizer architectures. The PLL fundamentals are introduced along with analysis of the jitter sources inside the PLL. Other frequency synthesis architectures like analog mixers and DDFS are also discussed. Then the concept of fractional N frequency synthesis is presented with a review of the architectures published in the literature.

Chapter 4 presents the fractional N synthesizer architecture developed in this research work. This design incorporates a technique called the Direct Digital Period Synthesis and a description of the same is presented. Then the architecture that uses the DDPS in the feedback loop of a PLL is presented with a discussion of the possible sources of jitter introduced by the DDPS circuit into the overall system. High level simulation results and calculation of the circuit level parameters are presented.

Chapter 5 describes in detail the circuit design of the individual blocks of the new fractional N synthesizer. Starting with a brief discussion on the CMOS process chosen for this design, the chapter presents a new Voltage Controlled Oscillator (VCO) circuit that uses a negative trans-conductance stage for fully differential tuning of the VCO frequency and uses continuous time common mode feedback to maintain the output DC level. Performance results of the VCO are also summarized. The chapter also explains the design of the Phase Frequency Detector (PFD), the differential charge pump/loop filter,

the feedback divider, the transition selector and the phase accumulator. A performance summary for each of the blocks is also presented.

For achieving high performance and high frequency operation, the layout of the blocks and the overall chip is as important as the design of the blocks. Chapter 6 discusses the issues relevant to the layout of this design. A potential chip level floor plan and a candidate floor plan for the VCO are explained in this chapter.

Chapter 7 presents conclusions concerning this design. Possible enhancements that can improve the performance of this architecture in terms of power, silicon area, flexibility are discussed. Comparisons of the performance and the estimated silicon area of this design with a few other designs published in the literature are summarized.

Chapter 2: Background

Demand for bandwidth in serial communication links has been increasing as the communications industry demands higher quantity and better quality of information. In addition, the ever increasing computational capability of processors is driving the need for very high bandwidth communication. Such links are an important part of multi-processor interconnection [2], processor to memory interconnection and serial network interfaces such as FireWire, Ethernet, SONET/Fiber channel and backplane transceivers. The research and design of transmit (TX) and receive (RX) circuits for these links currently targets speeds exceeding several gigabits per second (GBPS) [3]. Until recently, these circuits have been implemented in high speed but expensive processes like bipolar silicon, GaAs or SiGe technologies. However, commercial viability of these devices is dictated mostly by reduction in cost of manufacturing and being able to achieve greater levels of functional integration. This requirement has driven research work to explore the implementation of very high speed communication links in CMOS processes. It is indeed advantageous that CMOS technologies with feature sizes of 0.13 μm or smaller are available today. Such technologies provide extremely fast transistors so that a serial communications link running at 2.5 GBPS or higher can be implemented in CMOS [3], [4]. In most of these designs, single bit data stream is transmitted across a communication channel using a two level signaling technique like NRZ. Each data bit is represented as a +1 or a -1 and is transmitted using two corresponding voltage levels centered carefully around a DC level. Alternatively, a more sophisticated multi-level signaling scheme such as 4-PAM can be used to decrease the symbol rate with respect to the data rate. In [5], a multi-GBPS 4-PAM serial link transceiver has been implemented in 0.3 μm CMOS

technology. In a multi-level coding scheme such as 4-PAM, two data bits are coded into one of four possible data symbols. Different voltage levels are assigned to each of the four data symbols. This method reduces the line rate requirement by half for a given data rate.

2-1. SERIAL LINK SYSTEM DESCRIPTION

A typical serial link is comprised of three primary components: a transmitter, a communications channel and a receiver. The transmitter converts digital data bits into a serial bit stream and transmits the bit stream to the receiver through the channel. The receiver converts the received bit stream into digital data bits. The bit stream is subjected to the characteristics of the channel and when it arrives at the receiver inputs, it may appear substantially different from when it was transmitted. Figure 2-1 illustrates a typical serial communications link. Notice that the data stream appears *distorted* at the input of the receiver. It is the job of the receiver to process this input so that the transmitted data is recovered with minimal errors. An important part of this process is that the local receiver clock needs to be synchronized with respect to the transitions observed in the incoming data stream. This process is called clock recovery.

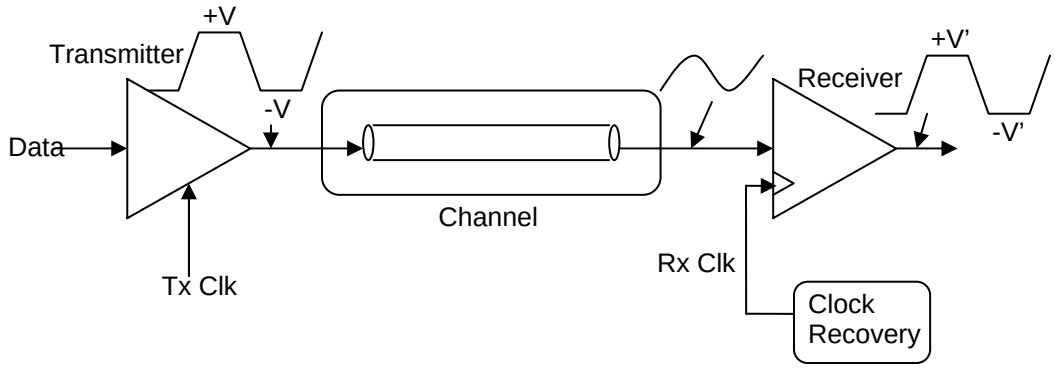


Figure 2-1. Basic serial link: the transmitter, the channel and the receiver

There are two distinct types of physical media used for serial communication channels: optical fibers and copper. Optical fibers provide extremely large communications bandwidth over very long distances [6]. However, the cost of laying the fiber along with the associated components and electronics at either end of the fiber makes this approach expensive and area-inefficient. As such, optical links are deployed in long-haul networks carrying large amounts of data. On the other hand, copper cables are much cheaper to deploy. Also, copper has been used as a medium for such a long time that a vast amount of already laid copper is available readily as a communication medium. However, copper as a medium offers limited data bandwidth and suffers from pronounced transmission line effects like amplitude distortion and non-linear group delay causing severe inter-symbol interference (ISI). As a result, copper is used mostly in short distance applications like board-board communications in a network backplane or computer to computer communications in a LAN environment [6].

2-2. SYSTEM METRICS AND REQUIREMENTS

Two metrics are most commonly used to evaluate the performance of a serial link. They are: the bit-rate (or its inverse, the bit-time) and the bit-error rate (BER). The bit-time is usually expressed in terms of nanoseconds or picoseconds. The bit-error rate is a statistical measure of the reliability of the link [7]. Although the information being communicated across the channel is digital, the electrical signal is analog in nature and the receiver has to make a decision about the data based on the received signal. There is always a probability of error during this process. The smaller the probability of making an error, the better is the quality of the overall system. BER is the statistical measure of the probability of an erroneous decision at the receiver. The data rate of the system is specified such that it satisfies a particular BER. For applications such as network

backplanes, BER requirements are on the order of 10^{-12} to 10^{-15} at data rates of 3.215 or 6.25 GBPS [3],[8]. This means that a serial transceiver operating at a speed of 6.25 GBPS can only make a single erroneous bit decision in 10^{12} or 10^{15} bits (a serial link transmitting at a rate of 6.25 GBPS will have to transmit for 45 hours continuously without making a single bit error to achieve a BER of 10^{-15}). That is a very stringent specification to meet and dictates most of the design decisions.

2-2-1. Bit-Error Rate

Referring back to Figure 2-1, a bit error occurs when the receiver's decision is different from the corresponding transmitted symbol. The error can occur due to imperfections in the transmitter design, excessive transmission line effects in the channel and imperfections in the receiver design. The biggest source of imperfections adversely affecting the performance is noise. Sources of noise can be in the transmitter and the receiver circuits in addition to the noise introduced by the channel. Broadly, the noise can be categorized into phase (timing) noise and amplitude noise [2]. These can further be classified into static and dynamic noise. Static noise effects introduce phase and voltage offsets and in general, can be handled easily. On the other hand, dynamic noise can severely affect the performance of the system and must be kept to a minimum.

An "eye" diagram is often used as a tool to ascertain the effect of noise in the system. Figure 2-2 shows how an eye pattern is formed by folding a continuous time data shown in Figure 2-2a into a single bit time as shown in Figure 2-2b. Noise in the system causes the shape of the eye to be distorted.

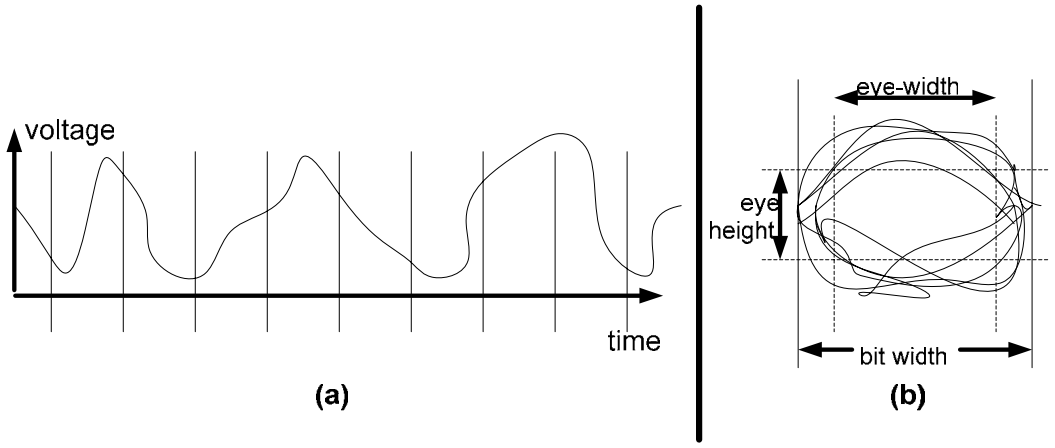


Figure 2-2. Illustration of a data eye: (a) continuous time data stream, (b) data eye formed by overlapping bit times

To further illustrate the specific effects of amplitude and phase noise on a data pattern, refer to the Figures 2-3(a) and (b). Figure 2-3(a) shows the effect of amplitude noise in the form of a reduction in the height of the eye. In the receiver, a decision circuit determines if the received data is a '+1' or a '-1.' This decision is made by determining if the received signal is above or below a particular threshold. In the presence of excessive amplitude noise, the received signal, even though it was a '+1,' may not exceed the receiver threshold causing the receiver to make a wrong decision. Similarly, a static voltage offset in the decision level can make the reception more sensitive to error by reducing the voltage noise margin [9].

The second source of error, timing error, can similarly affect the performance. The two main sources of timing errors that can cause bit errors are static and dynamic errors, collectively referred to as jitter. In an ideal system, the receiver clock edge lines up perfectly at the ideal sampling point as shown in the Figure 2-3b where the opening of the data eye is the widest. In reality, this may not be the case. The incoming data pattern suffers from transmission line effects and other non-idealities in the system and the data

edges arrive at different times as shown in the figure. In the figure, time t_{jd} represents the jitter in the data pattern. At the same time, the clock at the receiver end will have a finite amount of jitter, represented by t_{jc}, caused by circuit imperfections and noise sources on the power supply and substrate. As a result, the sampling of the data does not always happen at the widest part of the eye. Bit errors can occur if the deviation of the actual sampling point from the ideal one is excessive.

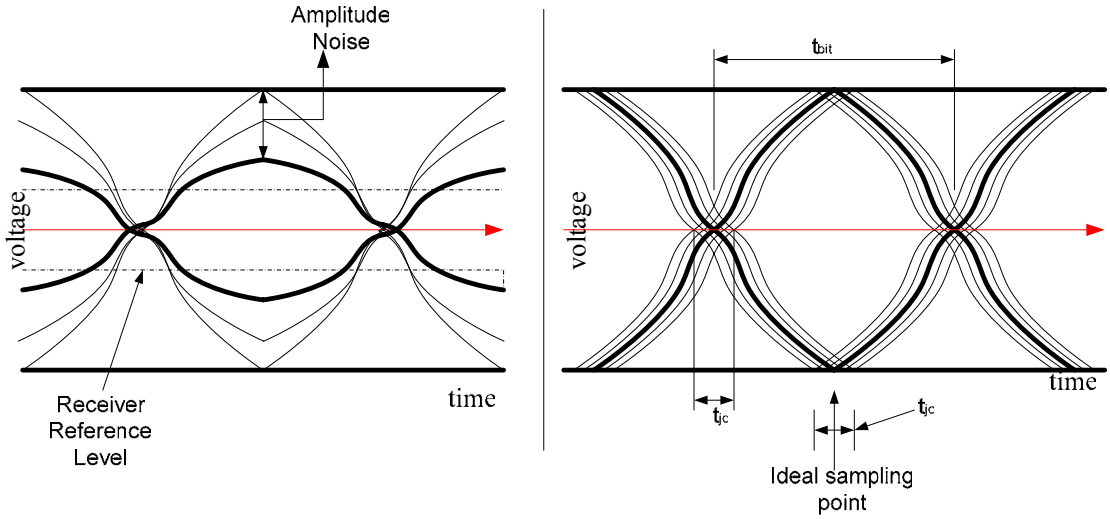


Figure 2-3. Effect of noise on a data eye: (a) Amplitude noise and (b) Phase or timing noise

The statistical properties of noise are used to estimate the performance of a system. If the noise is assumed to be uncorrelated with the transmitted signal, then it can be modeled as Additive White Gaussian Noise (AWGN). At the point where the receiver samples the incoming signal, there is a finite probability that the noise amplitude will be greater than the signal amplitude, causing a wrong decision. A measure of this probability is the bit-error rate (BER). In order to ensure that the opening of the eye is large enough to reliably sample the received signal, one can either A: increase the signal amplitude (signal power) and/or B: decrease the amount of noise in the system. Both of these can be

characterized by the Signal to Noise Ratio (SNR) of the system. Then the error probability function of a Gaussian distribution can be used to calculate the BER as a function of the SNR as per Equation 2-1.

$$Pe = \int_{S}^{\infty} \frac{1}{\sqrt{2\pi\sigma_s^{\,2}}} \exp\left(-\frac{y^2}{2\sigma_s^{\,2}}\right) dy \qquad (2\text{-}1)$$

Where: S is the peak signal amplitude, y is the signal and σ_s is the standard deviation of the noise. Figure 2-4 shows the plot of the Bit Error Rate as a function of the SNR expressed in decibels. Increasing system SNR, either by increasing signal power or by decreasing the noise in the system, improves BER as seen in the plot [10],[11].

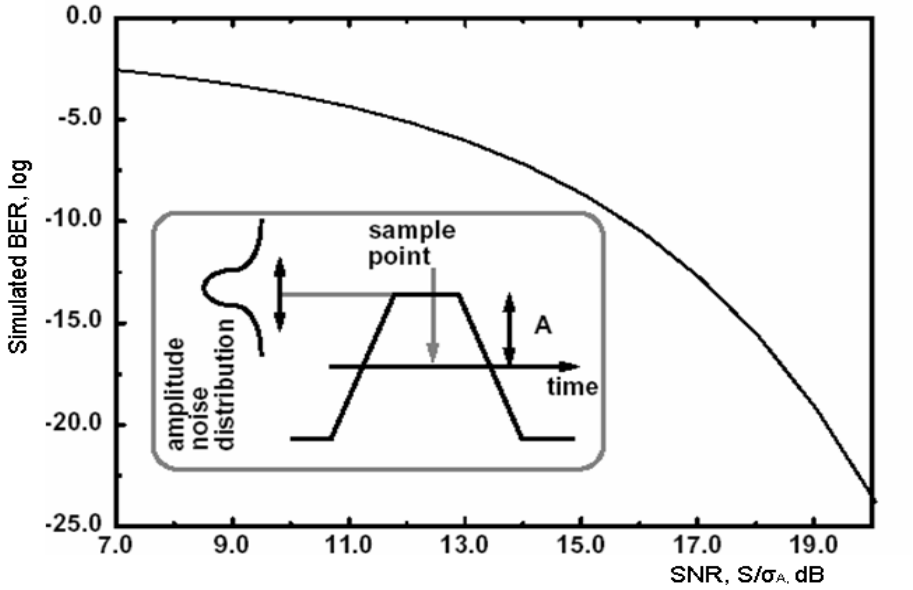


Figure 2-4. Plot of Bit Error Rate versus SNR.

2-2-2. Jitter

Jitter is broadly defined as the deviation in the period of a timing signal from its ideal period. There are several different ways jitter is quantified for a signal such as a clock or an NRZ data pattern. One classification divides jitter into deterministic and random groups [7].

Deterministic jitter is generally bounded in amplitude, non-Gaussian and expressed as a peak to peak value. Examples of deterministic jitter are, A: Duty cycle distortion caused by asymmetric rise and fall times, B: Inter-Symbol Interference (ISI) caused by finite channel bandwidth effects, C: Sinusoidal jitter caused by power supply noise and D: Uncorrelated jitter caused by interference from nearby signals [7].

On the other hand, random jitter is assumed to be Gaussian in nature and accumulates from thermal noise sources. Because peak to peak measurements take a long time to achieve statistical significance, random jitter is measured as an RMS value. Multiple random jitter sources add in RMS fashion governed by the characteristics of random processes and the central limit theorem. However, in order to fully specify the jitter performance of a system, such as a serial communications link, the random jitter is often converted into an equivalent peak to peak number and added to the deterministic jitter to get a total peak to peak jitter value. Although Gaussian statistics imply "infinite" peak to peak amplitude, a useful peak to peak number can be calculated from the RMS value given a probability of exceeding the peak to peak value. For example, the peak to peak random jitter having less than 10^{-12} probability of being exceeded is 14.1 times the RMS value. It turns out that this method of quantifying random jitter ties well into the BER specification of a communications link. Table 2-1 below tabulates the BER and the equivalent peak to peak/RMS ratio of the random jitter in the system [7].

It should be noted that the BER specification for a system implies that the total jitter, expressed as a peak to peak value, can not exceed a particular value. The total jitter is the sum of the random and the deterministic jitter. Knowing the deterministic portion of the jitter in a system, Table 2-1 provides a means to estimate the jitter bounds for the random jitter sources [12].

Table 2-1. BER vs RMS jitter, expressed as peak to peak jitter value.

PROBABILITY OF ERROR (BER)	P-P= N*RMS
10^{-10}	12.7 * RMS
10^{-11}	13.4 * RMS
10^{-12}	14.1 * RMS
10^{-13}	14.7 * RMS
10^{-14}	15.3 * RMS

The presence of jitter affects the performance and operation of both the transmitter and the receiver in a serial communications link. At the transmitter end, any jitter on the clock used to transmit the data on to the channel will cause jitter in the transitions of the transmitted data. At the receiver, the local clock used to sample the data can have jitter and that can significantly alter the sample point in the data eye. Beyond the above mentioned effects, presence of jitter can have an effect on the SNR as well. Due to cost, area and reliability issues, most systems have a maximum bound on the usable signal power. As a result, improvement in BER is possible only by reducing the noise in the system. To achieve that, frequencies above the data bandwidth are filtered to reduce the amount of wideband noise like thermal noise. However, this filtering has an adverse effect on the data pattern. The higher frequencies that give the data transitions sharp square edges are filtered out. Thus the data pattern no longer has well defined periods. If such a data pattern is sampled at an instant where the signal amplitude is not at its peak, the resulting *non-optimal* sampled value can be within the window of the decision threshold causing a bit error. The slewing portion of the signal effectively translates phase noise or jitter into amplitude noise. By taking into account the increase in

noise contributed by the jitter, one can calculate the BER using Equation (2-1). In the equation, σ_A is adjusted to include timing jitter as shown in Equation (2-2).

$$\sigma_{total} = \sqrt{\sigma_A^{\,2} + g(\sigma_\phi)^2} \tag{2-2}$$

Where $g(\sigma_\varphi) = S|\sigma_\varphi/\pi|$, is expressed in radians normalized to 2π. Figure 2-5 plots BER vs SNR for different values of σ_φ. Notice that as the noise approaches half the bit-time, the BER effectively flattens and increasing SNR does not help at all.

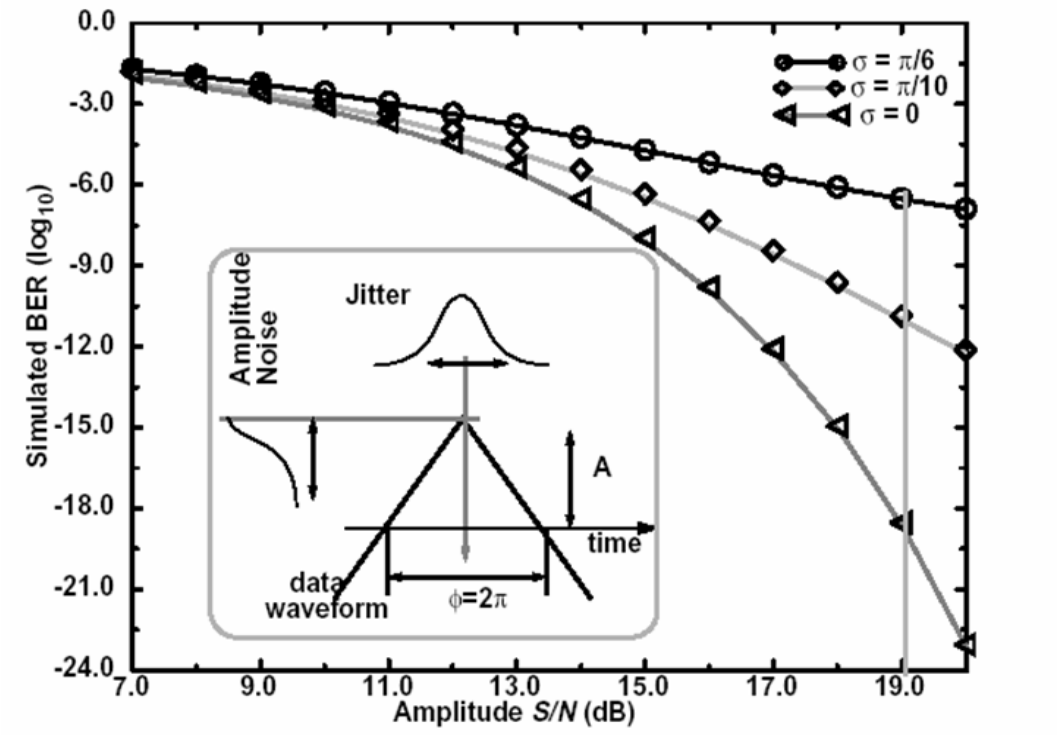


Figure 2-5. BER versus SNR for different timing jitter σ_φ values.

2-3. NETWORK BACKPLANES AND CHANNEL CHARACTERISTICS

This research work addresses serial communication links in network backplanes. A backplane is simply a connection between shelves and racks of networking equipment inside a telecommunication network [3],[13],[14]. It is constructed as a collection of large

electrical circuit boards with many sockets into which smaller networking boards, called line cards, are plugged. The backplane is constructed such that data are switched between the individual line cards. Each line card typically contains 8-16 serial data ports, with each port operating at serial rates of 2.5 to 6.25 GBPS. A conceptual visualization of a network backplane is shown in Figure 2-6.

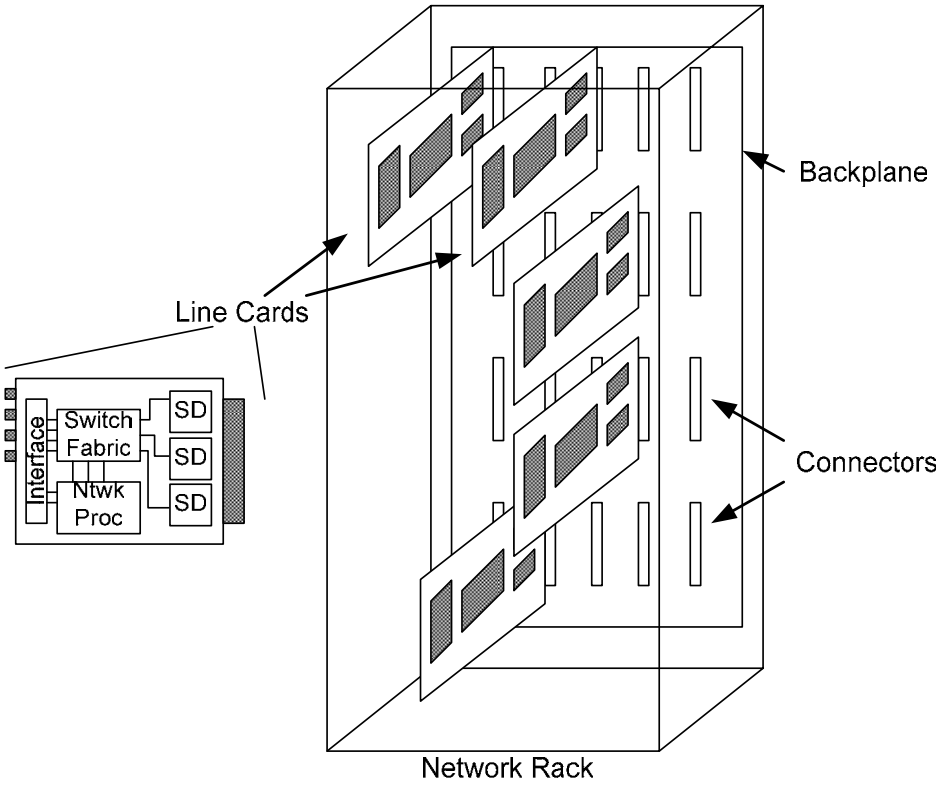


Figure 2-6. A typical network backplane system

Thus a backplane provides a massive pipeline for high speed data transport. The electronic device that actually performs the serial communication transmission and reception is called a **Ser**ializer-**Des**erializer (SerDes). As the name suggests, a SerDes takes in parallel, low speed data, serializes it into a single bit high speed data stream to be transmitted on to a channel. This is the "*Ser*"ializer part. On the "*Des*"erializer side, a

high speed single bit data is received from the channel and is converted into a low speed parallel data. The functional block diagram of a commercially available SerDes is shown in Figure 2-7, courtesy of Vitesse Semiconductor [15]. The key blocks that are relevant to this discussion are the parallel to serial converter, the clock multiply unit (CMU), the TX clock generator, the serial to parallel converter and the RX clock generator. This device takes in 32 bits of parallel data at 62.5 Mbps on its TXIN[31:0] inputs, converts this parallel data into a 34 bit serial data and transmits it on to a backplane at 2.125 GBPS (34x62.5e6) on the differential TXSA+/TXSA- outputs. Serial data is received on the RXSA+/RXSA- inputs differentially. After clock and data synchronization, serial to parallel conversion and word alignment, the 32 bit parallel data at 62.5 MBPS is output on the RXOUT[31:0] outputs. The incoming reference clock to the chip is 62.5 MHz, with a nominal rms jitter of 7 ps. The CMU multiplies this clock by 17 to generate a clock signal at 1.0625 GHz. The total analog power consumed by this device is 200 mA from a single 3.3 V power supply.

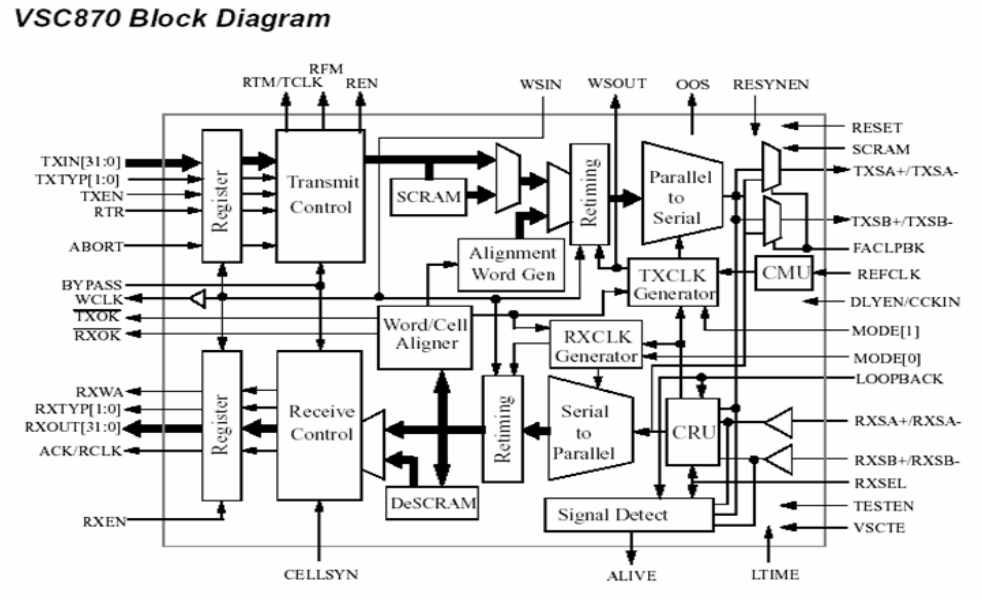


Figure 2-7. Functional block diagram of the VSC870 Backplane SerDes.

The communication medium in a backplane environment is usually printed circuit board (PCB) traces. In other words, it is copper traces on a PCB. Depending on the placement of the line cards in a backplane, the distance between two communicating SerDes' could be as little as an inch to as large as 36 inches. This large distance variation puts severe design constraints on the building blocks of the SerDes. Figure 2-8 shows the simulated attenuation profile for a line card constructed with the common FR4 material. A signal transmitted at a frequency of 3.125 GHz undergoes significant attenuation as shown in Figure 2-8. For example, at a distance of 30 inches, which is common in a backplane environment, a signal will suffer up to 15 dB of attenuation. To emphasize the severity of this effect, a 500 mV peak to peak signal will be received as a 100 mV peak to peak signal. Arbitrary amplification of the received signal will not work because the noise present in the channel will also be amplified. Additionally, since the channel is a limited bandwidth medium, different frequency components of the transmitted signal undergo different delays when they arrive at the receiver. This causes phase or timing distortion and introduces Inter-Symbol Interference (ISI) and data eye closure [16]. To alleviate these problems, some form of equalization is required in the receiver. An equalizer is a filter with controlled amounts of high frequency gain that is intended to complement the attenuation effects of the channel. It is important to observe that the frequency of the transmitted data is *half* that of the effective data rate. Data is transmitted and received on both edges of the clock, thus requiring that the clock frequency need only be half the required data rate. This feature reduces the design constraints on the clock generator. Furthermore, since fully differential design and communication is used in high speed systems, using both edges of the clock introduces more uniform data dependent supply and ground bounce effects. Noise management becomes less of a problem.

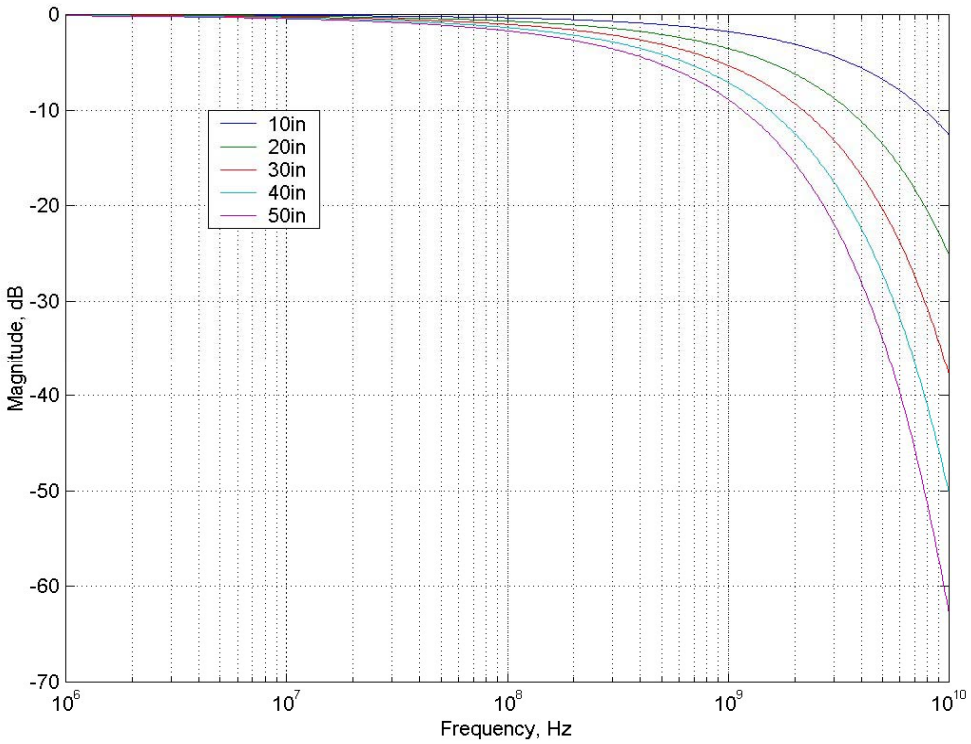


Figure 2-8. Attenuation profile for an FR4 material line card for several lengths of copper traces.

On the transmitter side, three different techniques are incorporated to improve the reliability of communication. The first one is the use of Error Correcting Codes (ECC). A pre-determined redundancy is added to the data such that any bit errors occurring at the receiver end can be corrected. There are numerous codes used for error correction in today's communication systems. The choice of a particular code depends upon on the kinds of errors in the system. For example, wireless communication systems, where errors can occur in bursts due to external conditions, use some form of Reed-Solomon codes which have excellent burst error correction properties. The performance of a particular code is measured in terms of *coding gain*, indicating the difference in the system SNR to achieve a specified BER with and without the ECC. Typical coding gain

numbers are 2-6 dB. Although ECC improves the BER performance of a communication link tremendously, it adds significant overhead in terms of silicon area and power consumption. For example, a decoder for a Reed Solomon code is a non trivial signal processing system requiring significant silicon area and power. However, the improved BER performance using ECC is an attractive proposition for serial communication links. As a result, inclusion of ECC in a design is usually a subject of trade-off.

The second technique used is transmit pre-emphasis. Up to 12-20 dB of gain is applied to the transmitted data to amplify the high frequency components. Implementing transmit pre-emphasis at high frequencies like 3.125 GHz or more is also a non-trivial design task. Even in deep sub-micron processes like 0.13 μm, a lot of power has to be dissipated in the transmit amplifier in order to achieve the required gain and bandwidth. Moreover, analog circuits have to be operated in open loop at these frequencies which can lead to undesirable DC offset and stability issues.

The third technique is to use line coding. Here, the transmitted data is encoded and redundant bits are added to the data stream to ensure a minimum percentage of transition density in the data stream. Transitions in the data stream are needed for robust clock recovery and synchronization on the receive side. The topic of line coding is explored in detail in the section below.

2-4. LINE CODING

High speed data is encoded with a line-code to ensure a certain transition density in the transmitted data. The code is designed such that the transition density is close to 50%. This ensures a good DC balance in the transmitted data and reliable clock recovery and data synchronization at the receiver. However, the line code adds data overhead. For example, one of the popular codes is an 8b/10b scheme. In this code, 8 bits of user data

are encoded into 10 bits of transmitted data. This scheme adds a 25% data overhead. If a usable data rate of 5 GBPS is required, the line data rate must be 6.25 GBPS. Another line code that is used frequently is a 64b/66b scheme. Here, 64 data bits are encoded into 66 transmitted bits. A third method is to use data scramblers. Linear feedback shift registers are used to scramble the incoming data, add pre-determined overhead bits (to identify start/end of frames, etc.) and transmit the resulting data. A de-scrambler is used on the receive side to recover the data. Both the data scrambling method and the 64b/66b method have less overhead than the 8b/10b coding scheme. Table 2-2 below summarizes the various coding schemes, the serial line rate for a given user data rate and the resulting transmit clock frequency. If a fixed crystal oscillator reference is used to generate the transmit clock, using the 64/66 or a scrambler based 32/34 scheme will require non-integer multiples of the crystal frequency.

Table 2-2. Summary of line coding schemes and resulting TX clock frequencies

LINE CODES	USER DATA RATE	SERIAL LINE RATE	TX CLOCK FREQUENCY	MULTIPLY RATIO FOR A 312.5MHZ CLOCK
8b/10b	5 GBPS	6.25 GBPS	3.125 GHz	10
64b/66b	5 GBPS	5.15625 GBPS	2.578125 GHz	8.25
32b/34b Scrambler	5 GBPS	5.3125 GBPS	2.65625 GHz	8.5

It is conceivable that a serial link might have to support more than one type of line coding scheme so that the user can switch between them based on the channel conditions. For example, for a low noise, short distance backplane, the 8b/10b scheme would be preferable due to popularity and legacy issues. On the other hand, for a noisy channel of

long lengths where transmission line effects are significant, it is preferable to reduce the line speed as much as possible and still achieve the necessary user data rate. Line codes with less overhead are preferred in such situations. The above discussion highlights the need for a scheme wherein an output signal of arbitrary frequency can be synthesized using a single high quality input reference frequency. The examples listed in Table 2-2 are only a small number of line codes that can be used. That means the TX clock synthesizer should ideally be able to synthesize any one of a range of frequencies. A straightforward approach is to implement multiple dividers and pre-scalers that can be switched corresponding to the required frequency [17]. This method is hardware intensive and very jitter sensitive. A better approach is to synthesize the output frequency under full digital control such that non-integer multiples of the input frequency can be generated. This is referred to as fractional-N frequency synthesis [18]. Several fractional-N type frequency synthesizers have already been reported in literature [19]-[24]. However, most of these designs make use of ΔΣ modulators leading to loop bandwidth constraints on the system.

2-5. TRANSMIT CLOCK REQUIREMENTS

The previous sections explained in detail the components of a serial link, the metrics by which a serial link performance is measured, backplane designs and coding schemes that affect the serial line data rate. For a backplane SerDes that needs to support two or more serial line speeds, there is a need for the generation of very low jitter transmit serial clock of a frequency not necessarily an integer multiple of the input crystal reference source. This section briefly summarizes some of the requirements of the transmit frequency synthesizer.

As studied earlier, jitter is a significant contributor to the performance of the backplane SerDes. Jitter causes timing or phase distortion causing higher bit error rates. For a 6.25 GBPS serial link, the period of a single data bit is only 160 ps. Even in a 0.13 μm technology, typical transmit rise and fall times tend to be on the order of 40-45 ps. That leaves only about 70 ps to align the receive clock edge such that ideal sampling of the data can be made. Any jitter introduced by the transmit clock directly reduces the jitter budget. As a result, peak to peak jitter requirements in the range of 45-55 ps are common. For this research a 50 ps peak to peak jitter requirement is targeted at a BER of 10^{-14}. Referring to Table 2-1, the RMS jitter number is calculated to be about 3.26 ps.

2-6. SUMMARY

An adaptive line rate network backplane architecture is introduced in this chapter. Using one of several possible line coding schemes, the serial communication rate in a network backplane is minimized while maintaining the same data throughput. Such an architecture imposes certain design constraints on the transceiver modules that are interfacing with the communication medium. The design constraints imposed on the transmit clock synthesizer are summarized in this chapter.

Chapter 3: Review of Frequency Synthesis Architectures

The requirements for the transmit clock synthesizer are established in Chapter 2. A frequency synthesizer with the capability of generating non-integer multiples of the input reference is desirable for an adaptive line rate network backplane. In order to pick the right architecture for this work, a review of the existing frequency synthesizer types is done while studying their relative advantages and disadvantages. A review of the several different fractional-N frequency synthesizers is presented in this chapter. Fractional-N synthesizers are capable of generating fractional multiples of the input reference clock. They are mainly used in wireless communication systems where individual channels are closely spaced in frequency and the receiver locks on to a particular channel.

3-1. FREQUENCY SYNTHESIZER TYPES

The three common types of frequency synthesis are: the direct analog synthesizer, the indirect or phase locked loop (PLL) synthesizer and finally, the direct digital frequency synthesizer (DDFS) [25]. Of these, the PLL based synthesizers have been used most commonly. However, the DDFS based synthesizers have gained popularity significantly as they address certain drawbacks of the PLL based synthesizers very well. A brief description of these synthesizer types is given below. PLL and DDFS based synthesizers are explored in greater detail in the following sections.

The direct analog synthesizer uses analog multiplication, mixing and division to generate a desired frequency from a reference [26]. This method is illustrated in Figure 3-1. The mixer is essentially a multiplier that generates the sum and difference frequencies by multiplying two sinusoids of different frequencies F_{LO} and F_{ref}. At the output of the mixer, both $F_{LO} + F_{\mathrm{ref}}$ and $F_{LO} - F_{\mathrm{ref}}$ frequencies are present. By adjusting the center

frequency of the bandpass filter, either of the two frequencies can be obtained. In this diagram, the sum frequency is shown to be output by the filter. In practice, this concept can be extended to multiple stages with suitable frequency dividers such that a frequency of arbitrary accuracy can be synthesized [26].

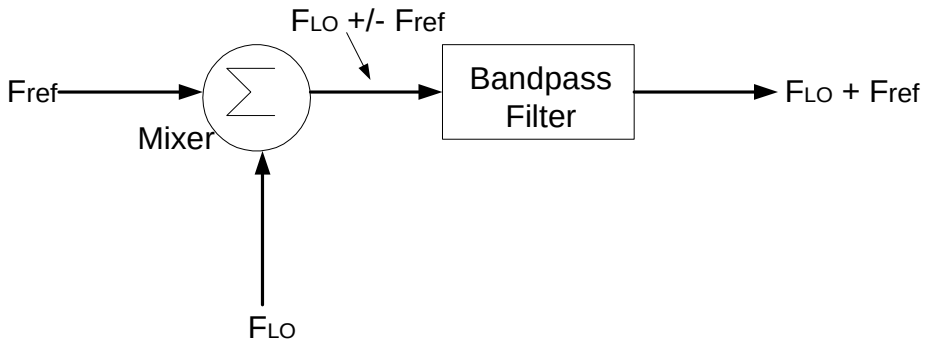


Figure 3-1. Conceptual diagram of a direct analog Frequency Synthesizer

Note that by changing F_{LO}, different frequencies can be generated. A switch network chooses one of several different frequencies available and by suitable tuning of the bandpass filter, the required frequency is generated. A detailed scheme of a practical implementation is given in [26]. This type of synthesizer can change output frequency rapidly. The time to make a change is dictated by the speed at which the local oscillator frequencies can be switched and the rate at which the bandpass filter can be retuned.

The output spectrum close to the synthesized frequency is very clean, essentially a replica of the local oscillator, but with frequency modulation (FM) sidebands due to imperfections in the multiplication process. This type of synthesizer can have very fine frequency resolution. However, often a fine frequency resolution requires a large number of stages. Thus, this type of synthesizer very quickly becomes bulky, power hungry and expensive. Moreover, many different frequencies need to be generated and silicon integration is impractical due to substrate and power supply noise reasons.

The second type of frequency synthesizer uses phase lock techniques to generate an output frequency from a reference frequency. Figure 3-2 shows a simplified block diagram of a PLL based frequency synthesizer. The functional blocks of such a synthesizer are the Phase-Frequency Detector (PFD), the charge pump, the loop filter, the Voltage Controlled Oscillator (VCO) and the divider [27].

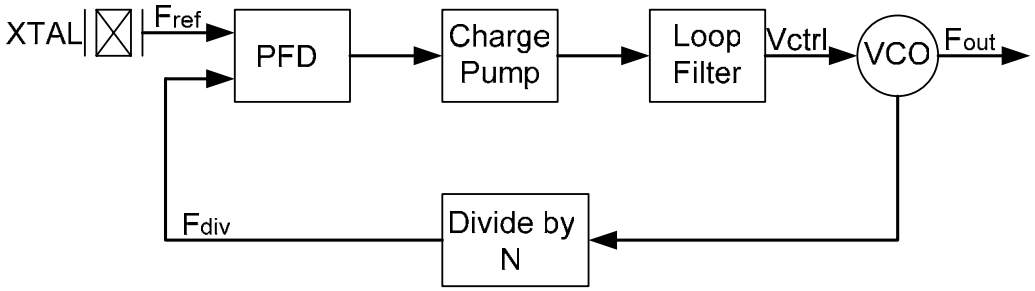


Figure 3-2. Simplified block diagram of a PLL based Frequency Synthesizer.

The basic functionality is as follows. The PFD compares the phase of the reference frequency F_{ref} with the divided version of the VCO frequency, F_{div}. It then generates a signal that is proportional to the phase error between the two signals. This signal is converted into an equivalent voltage by the charge pump which is lowpass filtered by the loop filter so that the loop filter voltage, V_{ctrl}, corresponds to the average value of the phase error [1],[28],[29],[33]. The loop filter voltage adjusts the frequency of the VCO. Since the phase of a signal is proportional to the time integral of frequency, the adjustment in the frequency of the VCO also adjusts the phase of the VCO output. When the loop reaches a stable condition, referred to as *lock*, the two signals to the PFD are either *in-phase* or have a fixed *static phase offset* and the output frequency is equal to the reference frequency times the feedback divide ratio. That is $F_{out} = N \text{x} F_{ref}$. The divide ratio

N is usually an integer (with exceptions to be introduced later) and the frequency resolution of a PLL frequency synthesizer is F_{ref}.

Earlier implementations of a PLL were mostly in the analog domain. The phase detector was an analog mixer followed by a loop filter. However, with the advantages that a digital implementation has over an all analog approach, many of the PLL functions are now implemented digitally. Furthermore, completely digital implementations are also reported in the literature. Here, the VCO is replaced by a Numerically Controlled Oscillator (NCO) and the loop filter is just an up/down counter [26].

The third common type of frequency synthesizer is the Direct Digital Frequency Synthesis (DDFS) scheme. A DDFS is a look-up table based sine-wave synthesizer. A block diagram of a DDFS system is shown in Figure 3-3. At the clock frequency, F_{clk}, a number representing the phase change per clock period, is added to the previous contents of the phase accumulator. The output of the accumulator is used as the address for a memory that contains the numerical representations of the *cosine* of the address. The output of the memory is then converted to an analog signal by a Digital to Analog Converter (DAC) and to a smooth sine-wave by the lowpass filter [30],[31].

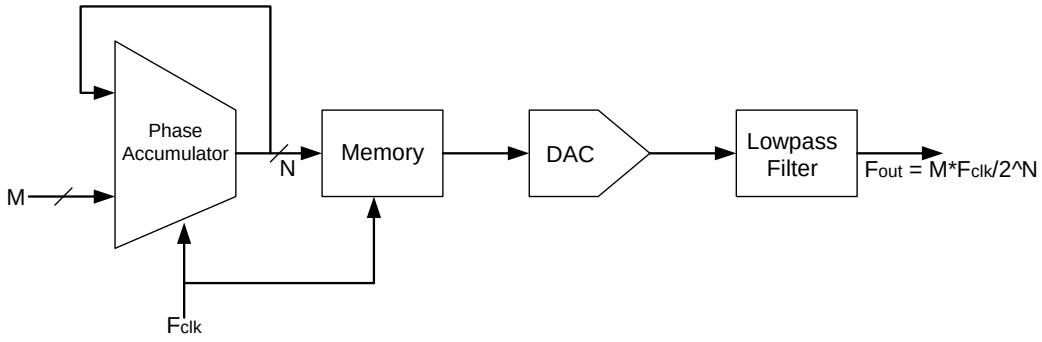


Figure 3-3. Block diagram of a Direct Digital Frequency Synthesizer.

Equation (3-3) shows the relationship between the synthesized frequency and the different variables controlling the operation of the DDFS system. In the equation, M is the input tuning word to the phase accumulator, F_{clk} is the update frequency of the phase accumulator and N is the resolution or the bit precision of the phase accumulator. Clearly, for large values of N, extremely fine frequency resolution can be achieved.

$$Fout = \frac{M * Fclk}{2^N} \tag{3-3}$$

DDFS architectures offer the advantages of very fine frequency resolution, fast frequency switching, complete digital control of the output frequency, great potential for integrated circuit implementations and finally, when used as a quadrature synthesizer in communication systems, DDFS offers unparalleled matching between I and Q channels [32]. The disadvantages are the need for a relatively high quality DAC and filter and also that the output frequency is only a fraction of F_{clk} and not a multiple like other synthesizer architectures.

3-2. PHASE LOCKED LOOP FREQUENCY SYNTHESIS

Section 3-1 presented a brief description of the PLL based frequency synthesizer. A more detailed look is presented in this section. A mathematical analysis of the PLL begins by observing the small signal model of the loop. It should be noted that the small signal model is applicable when the loop is the locked condition such that the effect of small variations in the input phase and the VCO phase can be studied accurately.

Figure 3-4 shows a linearlized model of a PLL frequency synthesizer along with the transfer function of each block. In the figure, K_{PD} denotes the combined transfer function of the PFD and the charge pump. K_{PD} is expressed in the units of V/Radian. The

output of the charge pump is filtered by the loop filter whose transfer function is given by $G_{LF}(s)$. The filtered voltage, V_{ctrl}, varies the frequency of the VCO. The VCO has a gain of K_{VCO}, expressed in Radians/s/V. In the Laplace domain, the VCO is also represented as an integrator, denoted by the 1/s factor. This is because the phase of a signal is an accumulation of frequency and in continuous time, it is represented as integration. The feedback divider has a transfer function of $1/N$ [27],[28].

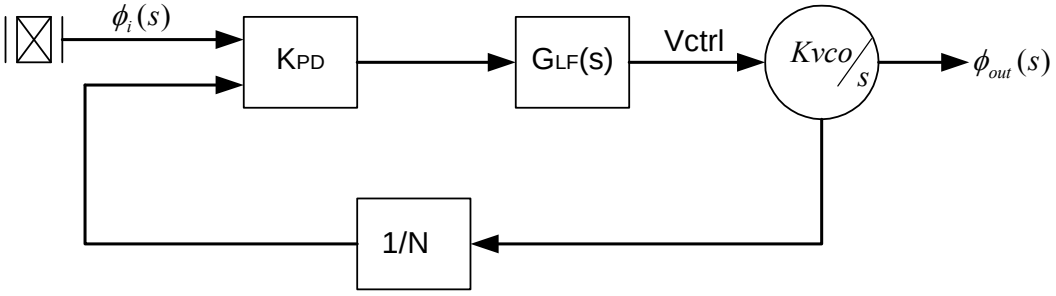


Figure 3-4. Small signal linearized model for a PLL frequency synthesizer.

The charge pump is a three-position electronic switch that is controlled by the three states of the PFD. The PFD output will have three states depending on whether the reference phase is ahead of the divider phase, behind the divider phase or equal to the divider phase. A simplified version of the charge pump is shown in Figure 3-5 below [29]. When the switch is set in the U or the D position, the charge pump delivers a current of +/- I_p to the loop filter impedance Z_{LF}. In the N position, the switch is open isolating the loop filter from the charge pump. The loop filter can be either passive or active. Most common implementations use simple passive loop filters due to the simplicity of their implementation. Due to the inherent switching behavior of the charge pump based PLL, it is a time-varying network.

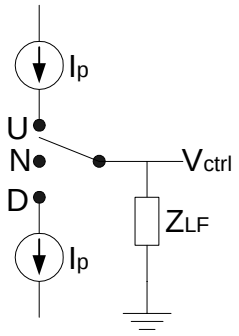


Figure 3-5. Conceptual version of a charge pump.

However, in most applications, the internal states of the PLL change very little in each cycle of the reference frequency. That is, the loop bandwidth of the PLL is quite small compared to F_{ref}. In these cases, instead of analyzing the PLL behavior in every cycle, an average behavior over large number of cycles of the reference clock is analyzed. By applying this averaged analysis, the small signal model of Figure 3-4 can be used. From Figure 3-4, a current of $I_p*|\Phi_e|$ is delivered to the loop filter impedance for the time t_p on each cycle of the reference. Each cycle has a duration $2\pi/\omega_{ref}$. Thus the average current delivered to the loop filter impedance over one cycle is given by

$$i_{av} = \frac{I_p \Phi_e}{2\pi} \text{ A} \tag{3-4}$$

Note that i_{av} is also the error current averaged over many cycles provided that both inputs to the PFD are periodic [29]. This condition is always satisfied in the case of a frequency synthesizer. The control voltage to the VCO, V_{ctrl}, is then given by,

$$V_{ctrl}(s) = I_{av}(s) Z_{LF}(s) = \frac{I_p Z_{LF}(s) \Phi_e(s)}{2\pi} \tag{3-5}$$

Where: $I_{av}(s)$ is the Laplace transform of $i_{av}(t)$, $Z_{LF}(s)$ is the Laplace transform of $z_{lf}(t)$ and $\Phi_e(s)$ is the transform of $\Phi_e(t)$. During lock, the phase of the VCO output is given by,

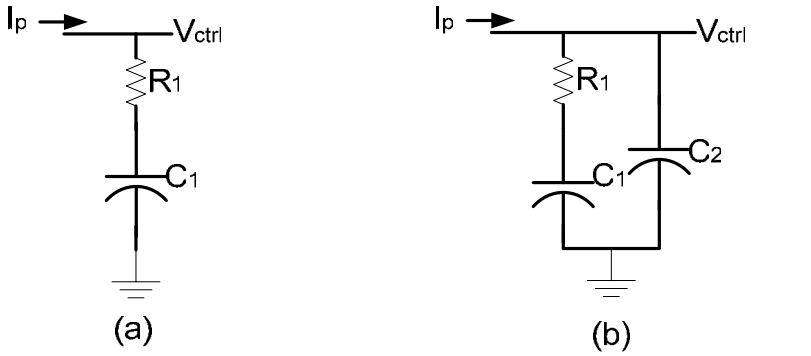
$$\Phi_o(s) = \frac{K_{VCO} V_{ctrl}(s)}{N.s} \tag{3-6}$$

Where: K_{VCO} is the VCO gain expressed in Radians/s/V. Since the PLL based frequency synthesizer includes frequency division of N, the value of K_{VCO} has been divided by N [33]. From Equations (3-5) and (3-6) and knowing that $\Phi_e(s) = \Phi_i(s) - \Phi_o(s)$, the loop transfer functions of the loop can be calculated to be,

$$\frac{\Phi_o(s)}{\Phi_i(s)} = \frac{K_{VCO} I_p Z_{LF}(s)/N}{2\pi s + K_{VCO} I_p Z_{LF}(s)/N} \tag{3-7}$$

and

$$\frac{\Phi_e(s)}{\Phi_i(s)} = \frac{2\pi s}{2\pi s + K_{VCO} I_p Z_{LF}(s)/N} \tag{3-8}$$



$$\frac{V_{ctrl}(s)}{I_{cp}(s)} = \frac{1 + sR_1C_1}{sC_1} \qquad \frac{V_{ctrl}(s)}{I_{cp}(s)} = \frac{1 + sR_1C_1}{s[(C_1 + C_2) + sR_1C_1C_2]}$$

Figure 3-6. Typical loop filter implementations: a. Series R-C with a pole at origin and a zero. b. Series R-C with a ripple suppressing capacitor.

Common implementations of the loop filter $Z_{LF}(s)$ use passive elements like resistors and capacitors due to their simplicity and ease of implementation. Two loop

filter topologies are shown the Figure 3-6 [28]. For the circuit shown in Figure 3-6(a), the closed loop transfer function is

$$\frac{\Phi_o(s)}{\Phi_i(s)} = \frac{\frac{K'}{2\pi C_1}(sR_1C_1 + 1)}{s^2 + s\frac{K'R}{2\pi} + \frac{K'}{2\pi C_1}} \tag{3-9}$$

Where: $K' = K_{VCO}.I_p/N$. Equation (3-9) indicates that this is a classic second order closed loop system. Its natural frequency ω_n and damping factor ζ are readily calculated to be

$$\omega_n = \sqrt{\frac{I_p K_{VCO}}{N 2\pi C_1}} \tag{3-10}$$

and

$$\zeta = \frac{R_1}{2}\sqrt{\frac{I_p C_1 K_{VCO}}{N 2\pi}} \tag{3-11}$$

The transfer function of Equation (3-9) is of a low-pass nature, indicating that the loop will respond to any low frequency phase noise present at the input. Also, since the DC gain of this function is 1, any static phase shift in the input reference will show up at the VCO output unchanged [33]. One drawback of the loop filter of Figure 3-6(a) is that it can introduce '*ripple*' in the control voltage even when the loop is locked [29]. Since the switches in the charge pump turn on at every phase comparison instant, any mismatch between the current sources of the charge pump flows through R_1 causing a step at V_{ctrl}. Over successive reference frequency cycles, this will appear as ripple in the control voltage. The ripple modulates the VCO frequency causing timing jitter. For frequency synthesizers, this effect is particularly undesirable.

One way to suppress the ripple is to add a second pole into the loop filter, albeit at a high frequency. This can be done by simply connecting a second capacitor in parallel with the R-C combination. Such a loop filter is shown in Figure 3-6(b), where capacitor

C_2 introduces the second pole. The effect of the loop filter having two poles is that the overall closed loop transfer function becomes a third order system [29]. The value of C_2 is chosen such that $C_1 >> C_2$. In such cases, only high frequency effects should be expected from the additional pole. The low frequency properties of the loop essentially remain the same as in the second order case.

3-3. JITTER IN PLL SYNTHESIZERS

Since PLLs operate on the phase of signals, they are susceptible to phase noise or jitter. The source of the jitter can be at the input to the synthesizer or any of the building blocks of the PLL. If each of the sources of jitter is assumed to be statistically independent, then their contribution to the overall performance of the system can be calculated using the small signal models presented earlier. To do this, various noise sources are added to Figure 3-4 producing Figure 3-7.

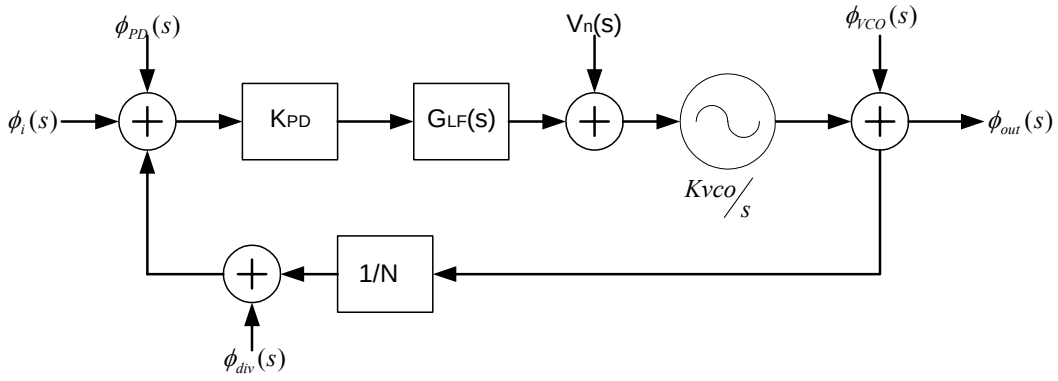


Figure 3-7. Small signal model of the PLL with various noise sources included.

In Figure 3-7, noise sources associated with the PFD and the charge pump, noise associated with the feedback divider and the noise due to the VCO are represented as phase noise sources whereas the noise associated with the loop filter is shown as a

voltage noise. For simplicity the loop filter is assumed to be of the type shown in Figure 3-6(a). The transfer function to the output due to the noise sources $\Phi_{PD}(s)$ and $\Phi_{div}(s)$ are the same and equal to

$$\frac{\Phi_o(s)}{\Phi_{PD}(s)} = \frac{\Phi_o(s)}{\Phi_{div}(s)} = \frac{\frac{K'}{2\pi C_1}(sR_1C_1+1)}{s^2 + s\frac{K'R_1}{2\pi} + \frac{K'}{2\pi C_1}}.N = \frac{\frac{K_{VCO}I_p}{2\pi C_1}(sR_1C_1+1)}{s^2 + s\frac{K'R_1}{2\pi} + \frac{K'}{2\pi C_1}} \tag{3-12}$$

The response of the loop to the noise injected at the output of the loop filter is given by,

$$\frac{\Phi_o(s)}{V_n(s)} = \frac{s^2}{s^2 + s\frac{K'R_1}{2\pi} + \frac{K'}{2\pi C_1}} \cdot \frac{K_{VCO}}{s} = \frac{s.K_{VCO}}{s^2 + s\frac{K'R_1}{2\pi} + \frac{K'}{2\pi C_1}} \tag{3-13}$$

and finally, the transfer function to the output for the VCO phase noise is given by,

$$\frac{\Phi_o(s)}{\Phi_{VCO}(s)} = \frac{s^2}{s^2 + s\frac{K'R_1}{2\pi} + \frac{K'}{2\pi C_1}} \tag{3-14}$$

The behavior of the PLL, in the small signal sense, to the noise sources indicated in Figure 3-7 can be observed from the above transfer functions. Any phase noise due to the PFD or charge pump or the feedback divider is *amplified* by the feedback factor *N*. This indicates for low jitter performance, it is not desirable to have large divider ratios. The response of the sytem to the noise introduced at the loop filter is of a *bandpass* nature with center frequency at the loop bandwidth of the PLL. Finally, the phase noise introduced by the VCO experiences a *highpass* function with its corner frequency at the loop bandwidth of the PLL. That is, any low frequency phase noise (or any long term jitter) in the VCO is filtered by the loop. However, high frequency noise (cycle to cycle jitter) of the VCO will appear unfiltered at the output of the PLL.

From the above analyses, some conclusions can be drawn with respect to choosing the parameters of a PLL based frequency synthesizer. A: The frequency resolution of the PLL is equal to the input reference frequency. If a fine frequency resolution at the output is desired, then F_{ref} must correspondingly be reduced. B: The loop bandwidth of the PLL should be smaller than the input frequency for stability reasons. In such a case, a low reference frequency implies a narrow loop bandwidth. This in turn means poor suppression of the VCO noise and slow response of the loop. The root locus of the third order PLL is plotted in Figure 3-8 where the feedback divide ratio has been varied from 1 to 100. It can be observed that as *N* increases, the complex poles move towards the *jω* axis indicating instability.

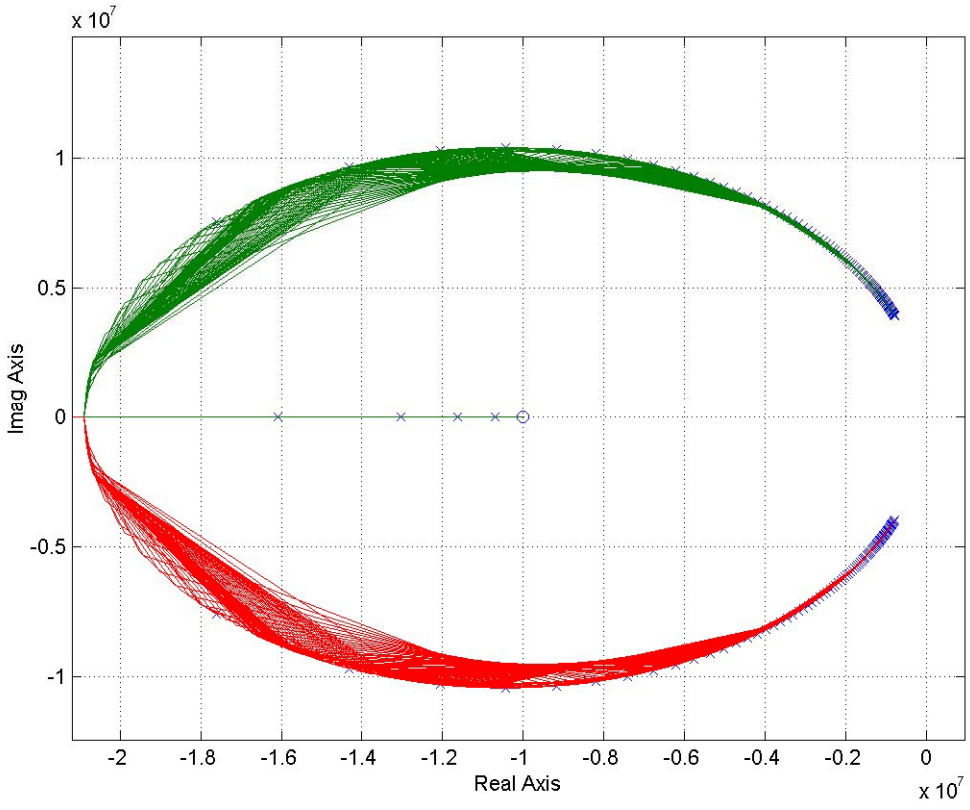


Figure 3-8. Root locus of the third order PLL showing effect of increasing *N*.

The above conclusions dictate the architectural specifications of PLLs to a large degree and for best performance, the feedback divide factor must be minimized and the input reference frequency must be maximized. This allows for a wide bandwidth PLL that can suppress a lot of the VCO jitter.

3-4. DIRECT DIGITAL FREQUENCY SYNTHESIS (DDFS)

The concept of DDFS was introduced in Section 3-1. This architecture offers several significant advantages over a PLL based frequency synthesizer. Most important of these is, of course, the extremely fine frequency resolution. Also, the frequency resolution is completely under digital control which makes DDFS very attractive for IC implementations. In practice, DDFS systems are used most commonly in wireless communication systems where many finely spaced carrier frequencies need to be synthesized. In addition, DDFS also has the advantage of very fast frequency switching. This characteristic is utilized in spread-spectrum or frequency-hopping systems, commercial spectroscopy and automatic test equipment (ATE) [31].

The operation of a DDFS system can be explained with the help of Figure 2-8. The figure also depicts the waveforms at the output of each of the blocks of the DDFS system. The underlying principle in a DDFS is that a fixed amplitude, fixed frequency and a fixed phase offset sine wave can be represented by $A_o sin(\omega t+\varphi)$. The signal phase is a linear function, as shown in the Figure 3-9. The slope or the gradient of the phase $d\varphi/dt$ is nothing but the angular frequency ω. To generate an output sinusoidal waveform, it is necessary to transform the phase $\varphi(t)$ into $\sin[\varphi(t)]$ and this operation is done by the ROM [32]. The output of the ROM is thus a digital representation of the sinewave. These digital samples of the sine-wave are converted into a sampled and held analog stair case by the Digital to Analog Converter (DAC). The DAC thus performs a zero order hold

operation which means that the spectrum of the output of the DAC has a sin(x)/x response. It is then the job of the analog filter that follows the DAC to filter out all the higher frequencies and convert the DAC output into a smooth sine wave.

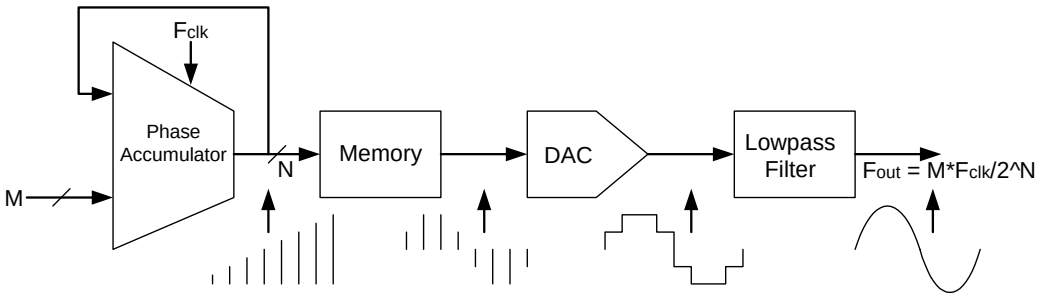


Figure 3-9. A DDFS system showing the signal processing at each block.

Since the state of the digital machine is known at every clock cycle, it is relatively straight forward to modify the output signal by adding phase shifting, amplitude modulation, frequency changes, etc., all in the digital domain. The phase accumulator performs the operation $S(n) = S(n-1) + M$, where M is the input control word. If the phase accumulator is N bits wide, then it will accumulate from 0 to 2^{N-1}. Beyond this, the accumulator overflows and starts over again. The rate of accumulation depends on F_{clk} and the phase resolution depends on M. If a zero output from the phase accumulator corresponds to a phase of 0 and 2^{N-1} corresponds to 2π, then the phase accumulator periodically generates phase angles from 0 to 2π. If the *cosines* of these phase angles are stored in a memory, it is a relatively simple exercise to access these values sequentially to complete the sine wave. The frequency of the output signal is given by Equation 3-3.

If the full precision of the phase accumulator is to be utilized, then the size of the memory needs to be 2^N. Therefore, the size of the memory grows exponentially for increasing phase accumulator precision. As a result, some form of truncation usually is done wherein only a fixed number of MSBs of the phase accumulator output are actually

used to address the memory. This process of *phase truncation* results in spurious content in the synthesized output. Several different methods have been proposed in the literature [32], [34], [35] to reduce the size of the memory. Some of these architectures take advantage of the symmetry of a sinusoid around $\pi/2$ and reduce the memory requirement by a factor of 4. Others use the polynomial expansion of the sine function to approximate the sinewave. Another method is to use the co-ordinate rotation techniques such as the CORDIC algorithm.

DDFS, at its most basic, is a digital signal processing (DSP) technique that is subject to the principles of Nyquist's sampling theorem as any other DSP technique like digital filters, FFTs etc. Because of this, the maximum value of F_{out} is equal to $F_{clk}/2$. However, in practice, F_{out} is limited to about $0.4*F_{clk}$ in order to keep the cutoff requirements of the analog low pass filter reasonable. This is a major drawback of the DDFS system when compared with a PLL. As presented in earlier sections, a PLL can lock the phase of the divided oscillator frequency to that of an input reference easily so that the oscillator output can be multiple of the input reference frequency. In a DDFS system, it is difficult but not impossible. Since the DAC does a zero order hold, the output of the DAC contains not only the fundamental F_{out} but also its images around F_{clk}, $2*F_{clk}$ and so on. The amplitudes of the images follow the *sinc* function. By placing a *bandpass* filter at the output of the DAC instead of a lowpass, it is possible to extract a frequency that is higher than F_{clk}. Use of images at higher than F_{clk} can make DDFS look like a PLL. However, such operation is restricted to the first three or four images beyond which the amplitude loss due to the *sinc* function will severely degrade the SNR of the signals.

The need for a DAC and an analog filter can also be a performance bottleneck for a DDFS system. The digital to analog conversion process introduces quantization error and the dynamic range (DR) of a DAC is given by

$$DR(dB) = 6.021 * N + 1.761 \quad (3\text{-}15)$$

Where: N is the resolution of the DAC and if the DAC has a differential non-linearity (DNL) of X (X <= 1 LSB), then in the dynamic range of the DAC shown in Equation (3-15) [36], N is replaced by (N-X). The DNL causes *harmonic distortion* in the output of the DAC and any harmonics higher than $F_{clk}/2$ will appear in the output spectrum as aliased images.

DDFS techniques have gained in popularity significantly, particularly in wireless communication systems. For portable applications, DDFS is considered power hungry at high clock frequencies due to large memory requirements. Many different techniques to reduce the memory size have been reported in literature. An architecture that eliminates the ROM altogether by using a non-linear DAC is reported in [37].

3-5. FRACTIONAL N FREQUENCY SYNTHESIS

Previous sections have described the different frequency synthesis techniques. One significant advantage that the DDFS scheme enjoys is the frequency resolution of the output, as governed by Equation (3-3). In a PLL based system, if a frequency resolution finer than F_{ref} is required, then a pre-scaler or pre-divider is used to divide F_{ref} into a smaller frequency and the feedback divider inside the PLL is correspondingly increased. Suppose that F_{ref} is pre-scaled by a factor M so that the input frequency to the PLL is actually F_{ref}/M. Then the output frequency $F_{out} = (N/M)F_{ref}$. By suitable selection of N and M, it is possible to synthesize a fractional multiple of F_{ref}. For synthesis of

relatively low frequencies, this method is widely used. Figure 3-10 shows a modified PLL for synthesizing fractional multiples.

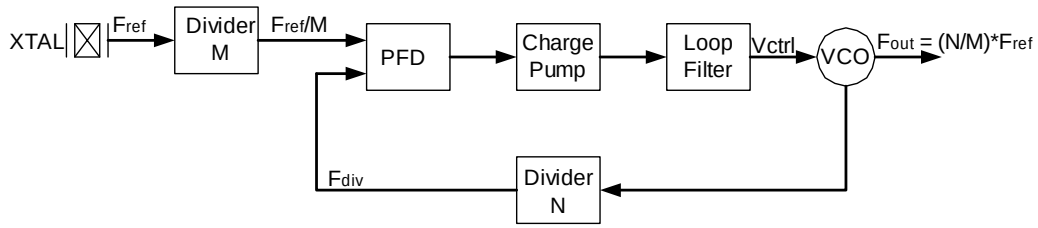


Figure 3-10. A PLL synthesizer using a pre-scaler to generate a fractional frequency.

The (*N*/*M*) method of synthesizing fractional frequencies presents several difficulties. Since the input frequency to the PLL is actually smaller by a factor of *M*, the loop bandwidth of the PLL must correspondingly be reduced to keep the loop stable. To generate an output frequency of 2.578125 GHz with a 312.5 MHz crystal, $M = 4$ and $N = 33$ is needed. As shown earlier, any noise originating at the feedback divider output gets *amplified* by *N* when it appears at the output. For low jitter applications, minimizing *N* is very important [38]. Increasing *N* can lead to another problem. In a PLL, the output of the loop filter is updated at the rate of the input reference frequency, F_{ref} (or in the case of Figure 3-10, F_{ref}/M). However, there are *N* cycles of the VCO in one cycle of the reference frequency. Any extraneous noise source introducing a noise spike into the VCO causes its output frequency to drift by a small amount until it is compensated by the loop. However, the compensation or update happens only at the next cycle of the reference frequency. If the divide ratio *N* is large, the input reference frequency is small relative to the PLL output and the output frequency can experience severe timing jitter until it is updated by the loop. In high speed communication systems, such bursts of jitter are not acceptable since they cause severe degradation in the quality of the link [38].

From the above discussion, it can be concluded that neither DDFS nor PLL in their simplest forms is capable of generating high fractional multiples of an input frequency with low timing jitter. Alternative architectures must be explored in order to achieve this functionality. Several such techniques are explored in the following sections.

3-6. HYBRID FREQUENCY SYNTHESIS

In a hybrid synthesis system, both DDFS and PLL are used. One method uses DDFS to generate an output signal to be used as the input reference to a PLL. The PLL locks to this signal to generate an output that can be several multiples of the DDFS output. Such a system is shown in Figure 3-11.

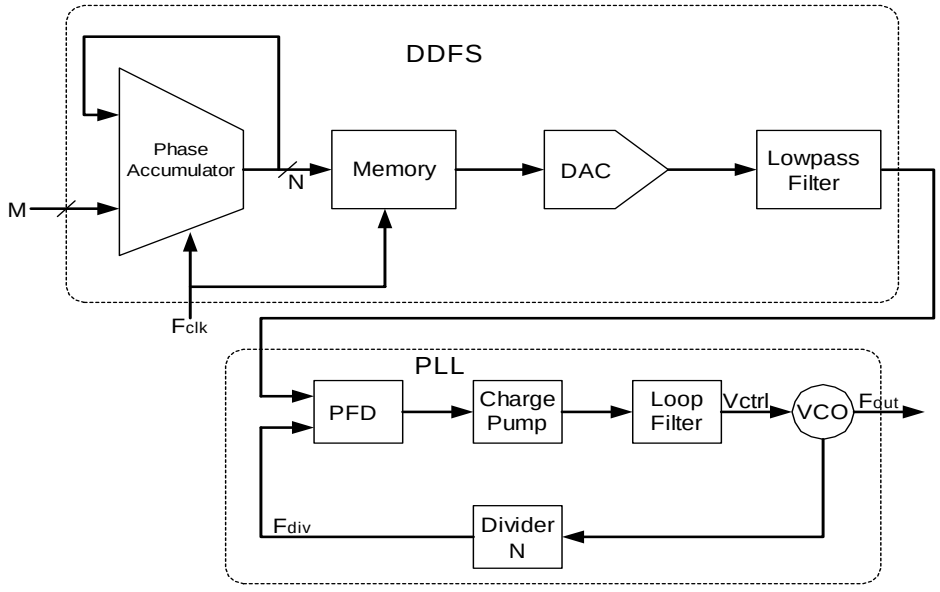


Figure 3-11. Hybrid Frequency Synthesis using cascaded DDFS/PLL stages. The DDFS generates the reference frequency for the PLL.

In the above system, the output of the DDFS is used as the input reference to the PLL [30]. The PLL locks to the phase of this signal and generates the output F_{out}, which

is *N* times the DDFS output. This system provides the advantages of fine frequency resolution in the DDFS and the high output frequency in the PLL. This is a good architecture to use when the spectral purity of the synthesized signal is not too critical.

In a second system, the DDFS is embedded inside the feedback loop of a PLL as shown in Figure 3-12. In the diagram, the DDFS, together with the divider, performs a fractional-N division inside the feedback loop of the PLL [30]. This technique achieves good frequency resolution and the frequency multiplication operation. However, one drawback of this technique is that the noise spurs introduced in the DDFS that are within the bandwidth of the PLL are increased by a factor of *N*.

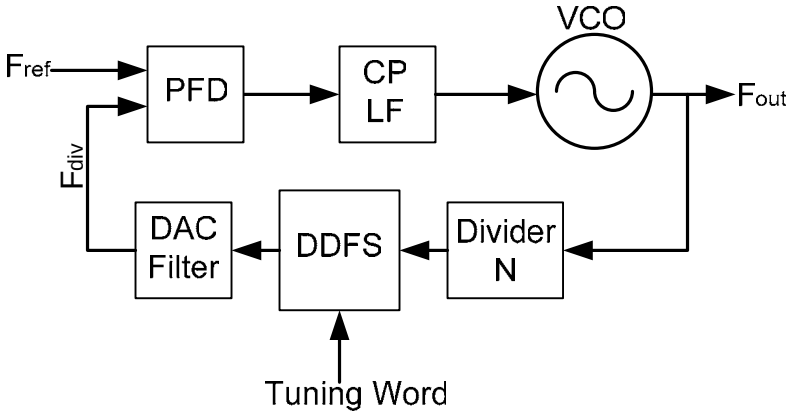


Figure 3-12. Hybrid Frequency Synthesis with DDFS in the feedback loop of a PLL.

Examining the above architecture further, if the divider were to be placed at the output of the DDFS, it would attenuate any phase noise/jitter/spurs introduced at the DDFS output. However, that would require the DDFS to run at the output frequency of the VCO, which can be very high. Also, to keep the phase noise of the output of the DDFS low, a high quality DAC and a high quality analog filter are needed.

3-7. FRACTIONAL-N SYNTHESIS WITH MULTI-MODULUS DIVIDERS

In this scheme, the feedback divider, instead of being an integer, is actually a fractional number. This is achieved by changing the feedback divide ratio between two integer values such that the *average* divide ratio over many cycles converges to the desired fractional number [23]. For example, suppose that the required transmit frequency is 2.578125 GHz and the input reference frequency is 312.5 MHz. Then N = 2.578125e9/312.5e6 = 8.25. Suppose that the feedback divider in the PLL is capable of dividing by 8 and also by 9. That is N can be set to either 8 or 9. Now, if $N = 8$ for three reference cycles and $N = 9$ for one reference cycle, then on average,

$$N = \frac{8*3+9}{4} = \frac{33}{4} = 8.25 \qquad \text{(2-16).}$$

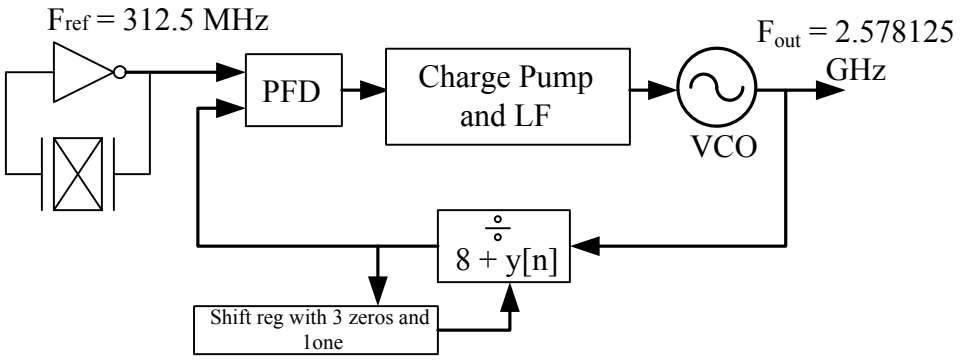


Figure 3-13. A fractional-N PLL that generates non-integer multiple of the reference frequency.

This is the fundamental principle behind most fractional-N PLLs [21]. This approach is depicted in Figure 3-13. While dynamically switching the divider modulus solves the fractional multiplication problem, a price is paid in the form of increased phase

noise. During each reference period the difference between the actual divider modulus and the average, i.e., the ideal divider modulus, represents an error that gets injected into the PLL and results in increased phase noise [21]. The main problem is that the switching of the divider modulus has periodicity that is mostly at low frequencies and within the bandwidth of the PLL. The periodic switching introduces a low frequency jitter into the PLL output. In the frequency domain, this will appear as spurious tones around the fundamental output frequency. Unfortunately, the only way to suppress this low frequency jitter is to make the PLL bandwidth very small [21].

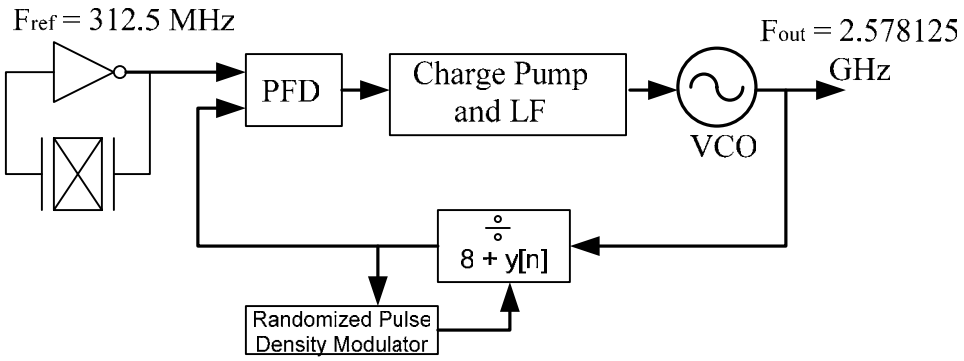


Figure 3-14. A Fractional-N PLL that generates non-integer multiple of the reference frequency with randomized divide sequence.

One way to eliminate the spurious tones is to randomize the periodicity of the divide sequence while still maintaining the average divide ratio. For example, as shown in Figure 3-14, a digital block can be used to generate a sequence $y[n]$, that approximates a sequence of samples of an independent random variable that takes on values of '0' and '1' with probabilities ¾ and ¼ respectively. During the n^{th} reference period, the divider modulus is set to $8 + y[n]$, such that the sequence has the desired average value but its power spectral density (PSD) is that of white noise [21]. Thus, instead of spurious tones or low frequency jitter, the modified technique introduces white noise or random jitter.

The portion of white noise within the loop bandwidth of the PLL shows up at the output. Mathematically, the randomized sequence $y[n]$ can be written as $y[n] = x[n] + e[n]$, where $x[n]$ is the desired fractional part of ¼ and $e[n]$ is the *quantization noise*. If the divide ratio between 8 and 9 is chosen sufficiently randomly, then $e[n]$ can be considered as zero-mean uncorrelated noise. It would then be advantageous if the spectrum of $e[n]$ can be *shaped* such that most of $e[n]$ lies *outside* the PLL's loop bandwidth.

3-8. PREVIOUS WORK

The use of a ΔΣ modulator to perform the shaping of the quantization noise mentioned above was first reported in [39]. Many implementations have since been reported in the literature [20], [23], [40]-[42], [47], [53]. Although a ΔΣ modulator *shapes* the quantization noise to lie out of band, the order of the modulator determines the extent of noise shaping and the rate at which the quantization noise increases out of band [36]. As a result, even the use of a ΔΣ modulator imposes restrictions on the loop bandwidth of the PLL. If higher order modulators are used, then their out of band quantization noise increases rapidly and sufficient filtering of this noise necessitates a small loop bandwidth.

In PLL implementations that use ring oscillators as VCOs, fractional multiplication of the input reference can be achieved by phase manipulation. A four stage ring oscillator generates eight phases of the same frequency, each phase separated by 45 degrees from the adjacent phase. By suitably manipulating these phase signals, fractional multiplication is achieved. Some implementations based on this scheme have been reported in the literature [43], [44].

In [43], a PLL that can synthesize a 1.8 GHz frequency is described for wireless digital data communication systems. A divided version of the VCO clock is latched

sequentially by the 8 phases of a four stage differential ring oscillator. The outputs of the latches are replicas of the divided clock separated by $1/8^{th}$ of a VCO period. Using an edge combining technique, an output frequency $F_{out} = (N + k/8)*F_{ref}$ can be synthesized. Although this design includes a calibration loop that tries to match the phases of the VCO precisely, any mismatches caused by the latches will introduce jitter. Since the divided frequency is sequentially latched, mismatch in the earlier stages will propagate into each latched phase.

In [44], a conventional PLL loop locks to a reference frequency and generates multiple phases of the same frequency using a ring oscillator. Then a digital subsystem operates on these phases based on an algorithm and generates an output frequency, which can be any fractional multiple of the VCO output and hence, the input reference. This architecture has a lower jitter bound set by the time delay between two adjacent VCO phases. Thus, for a VCO running at 3.125 GHz and using a 4 stage ring oscillator, time delay between adjacent phase is (320/8) ps = 40 ps.

Another architecture that uses a cascade of a multi-modulus divider and a fixed divider is published in [45]. This design describes a synthesizer at 800 MHz center frequency. Another interesting architecture is published in [46] where the input reference is first divided by a fractional divider. The divider's output can be expected to have significant jitter. A narrow band PLL is first used to *filter* out the jitter and another wideband PLL is used to multiply the output of the narrowband PLL to a frequency of 2.4 GHz.

Finally, there are also other reported synthesizer architectures based on Delay Locked Loops (DLL). A DLL is different from a PLL wherein the VCO is replaced by a Voltage Controlled Delay Line (VCDL). A few, high performance DLL based synthesizer architectures have been reported in literature [38], [48]-[52], [54], [55]. DLLs

have a significant advantage over PLLs in their phase accumulation behavior. In a PLL, the VCO phase is an integration or accumulation of its instantaneous frequency. This inherent integration behavior causes the VCO to accumulate any phase error introduced in the loop. On the other hand, in a DLL, the phase memory is reset once every reference frequency cycle [48]. Since the DLLs do not have an oscillator running at high frequency, some form of phase combining is needed to generate an output frequency higher than the input reference. A DLL based edge combining frequency synthesizer is described in [38]. The drawback of a DLL is that delay mismatches arising in the VCDL can cause severe jitter in the synthesized clock. In a PLL, the VCO is a high gain feedback system with a relatively high Q factor. The VCDL, however, just generates delayed replicas of the input clock. Any delay mismatch among the stages of the VCDL can lead to output jitter. As a result, DLL based circuits are usually not employed for clock synthesis applications. The circuit described in [49] generates output frequencies up to 1.6 GHz and this is among the highest output frequencies generated using a DLL.

3-9. SUMMARY

The three major frequency synthesis architectures are described in this chapter. Of these, PLL based synthesizers are most commonly used. Very high output frequencies can be synthesized with relatively inexpensive low frequency crystal oscillators using PLLs. However, non-integer frequency multiplication and requirement for fine frequency resolution impose severe constraints on the PLL parameters affecting the performance. DDFS architectures, on the other hand, offer excellent frequency resolution, fast frequency switching (there is no negative feedback loop to settle in a DDFS) and complete digital control. However, they are governed by Nyquist theorem and realistically synthesize frequencies that are significantly smaller than the clock frequency.

For network backplanes that require a very high frequency, but a flexible clock synthesizer, neither of the two techniques is directly suitable. Reported architectures that either combine the PLL and DDFS features or other forms of fractional-N multiplication were explored in detail. Fractional-N synthesizers using ΔΣ modulators to randomize the multi modulus feedback divider have been reported. These, however, constrain the loop bandwidth of the PLL thereby allowing VCO jitter to appear at the output. For network backplane applications where extremely fine frequency resolution is not required, a simpler architecture that does not restrict the PLL loop bandwidth is desirable.

Chapter 4: Architecture Design

The requirement for a fractional-N frequency synthesizer with low timing jitter was established in Chapter 2. It must be emphasized that for applications like a network backplane, the intention is to be able to generate one of many frequencies depending on the particular line code chosen. The selection of the line code, in turn, is dependent on the operating environment of the network backplane. A few example codes and the resulting clock frequency requirements were tabulated in Chapter 2.

Chapter 3 discussed frequency synthesizer architectures published in recent literature. Two major schemes are used for fractional multiplication. In one, a multi-modulus divider is used in the PLL whose divide ratio is manipulated such that on the average, a fractional multiple of the input clock is generated. This scheme is not suitable for serial communications applications as its cycle to cycle timing variation is high. The second scheme uses phase manipulation and combination to generate fractional multiples.

A new fractional-N frequency synthesizer architecture is presented in this chapter. This architecture is based on the phase manipulation and combination scheme. However, it differs from those presented in the literature. This architecture incorporates a scheme called Direct Digital Period Synthesis (DDPS). The DDPS scheme is combined with the feedback divider to achieve fractional multiplication. The concept of DDPS is briefly explained in the following section. The new frequency synthesizer architecture is described after that. Then the system design to select the PLL parameters is discussed.

4-1. Direct Digital Period Synthesis (DDPS)

A digital clock synthesis system is described in [56]. In this system, an output clock is synthesized using multiple phases of the same frequency of an input clock.

Although the architecture presented in [56] is similar to some that are discussed in Chapter 3, specifically, [43] and [44], it has the advantages of simplicity of implementation and potentially better jitter performance. Compared to DDFS, the DDPS scheme is simpler to implement due to the absence of critical analog elements like the DAC and the filter.

A block diagram of a DDPS system is shown in the Figure 4-1. It comprises of a transition generator (TGen) that produces several phases of the same frequency as the input clock. A transition selector (TSel) is responsible for selecting and propagating one of the transitions provided by TGen. This transition is propagated to the output clock F_{out}. The transition selection is done according to the output produced by the phase accumulator (Phase Acc).

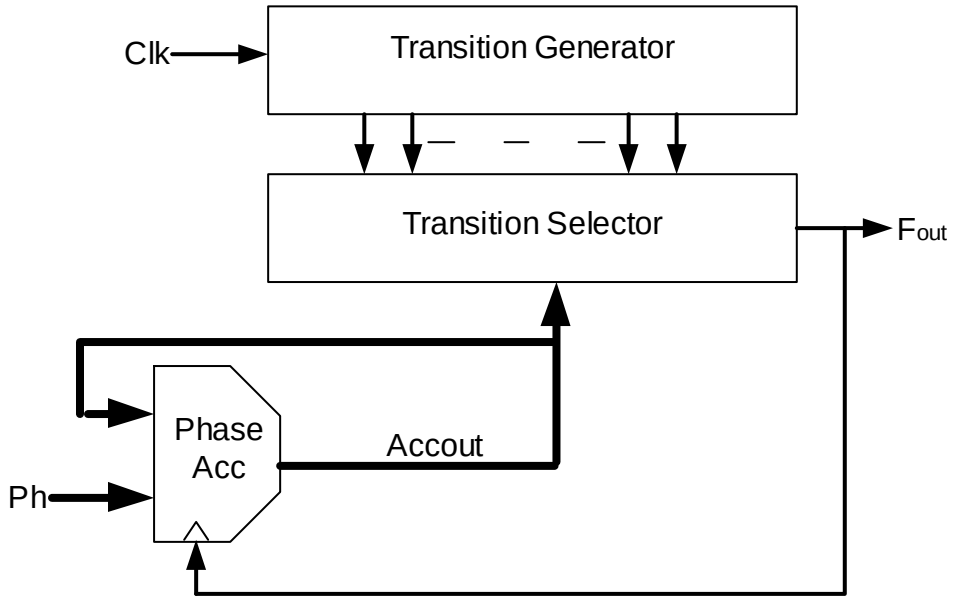


Figure 4-1. Architecture of a Direct Digital Period Synthesis system.

The phase accumulator computes a number that selects a particular phase of the input clock for the output. To illustrate the operation of this circuit, a simplified timing

diagram comprising of four clock phases and a 2-bit phase accumulator is shown in Figure 3-2. The input word to the Phase Accumulator in this example is Ph = 0.01_b. This value is the binary representation of the fraction ¼. Suppose that the initial value of the accumulator is 0.00_b. Then in response to the edges of the clock Fout, the accumulator will increment from 0.00_b to 0.01_b, 0.10_b, 0.11_b and the sequence repeats. In response to this sequence, the clock phases Φ_0, Φ_1, Φ_2 and Φ_3 are propagated to the output sequentially. As shown in the timing diagram, this creates a train of pulses at Fout such that the frequency of $F_{out} = F_{clk}$*4.

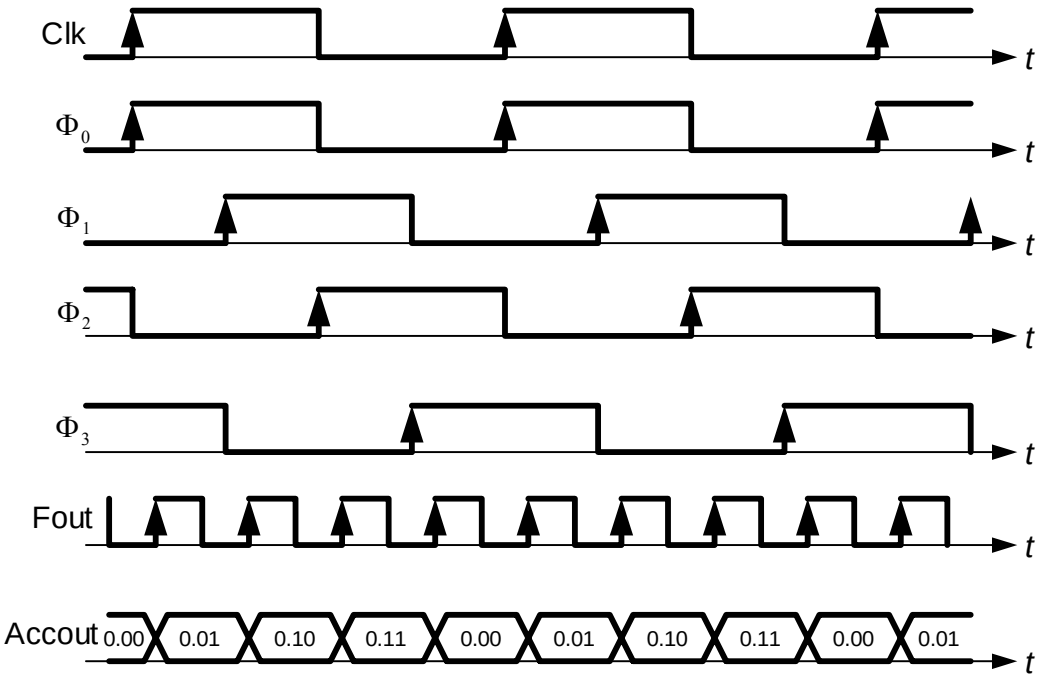


Figure 4-2. Timing diagram of the operation of the DDPS scheme.

If the input clock is F_{clk} and the phase accumulator input word is denoted as *Ph*, then the output clock frequency F_{out} is given by

$$F_{out} = \frac{F_{clk}}{Ph} \tag{4-1}$$

If $Ph < 1$ is the phase increment, then the output frequency is greater than the input frequency. Thus, DDPS in itself can perform the function of multiplicative frequency synthesis. Theoretically, the value of F_{out} can be really large. However, if Ph is too small, the next target clock phase from TGen arrives before the new value of the digital phase has updated the path from TGen to F_{out}. This causes a race condition leading to glitches if the path from TGen to F_{out} is modified too close in time to the transition selected by the digital output of the phase accumulator. The race condition is a function of process, temperature and voltage variations and can cause large and unpredictable jitter [56].

The timing problem can be overcome by updating F_{out} right after the transition on the next target phase of TGen. This results in a skipping of one full period of the input clock. In this case, the output clock frequency is given by the relation

$$F_{out} = \frac{F_{clk}}{1 + Ph} \tag{4-2}$$

The frequency resolution of this circuit is controlled by the resolution of the phase accumulator [56]. A 32-bit phase accumulator implementation allows sub-Hertz precision control in the output frequency. However, the time resolution of the output transitions and as a result, the output jitter, is determined by how well the different phases out of the TGen block are placed. That means, any jitter arising in the phases of the TGen block will significantly affect the critical loop timing. The timing diagram of the DDPS system that overcomes the race condition problem is shown in Figure 4-3. In this figure, the digital input number Ph is chosen to be ½ or 0.10_b. It can be observed that the synthesized output frequency now is $\frac{F_{clk}}{1.5}$ as predicted by Equation (4-2).

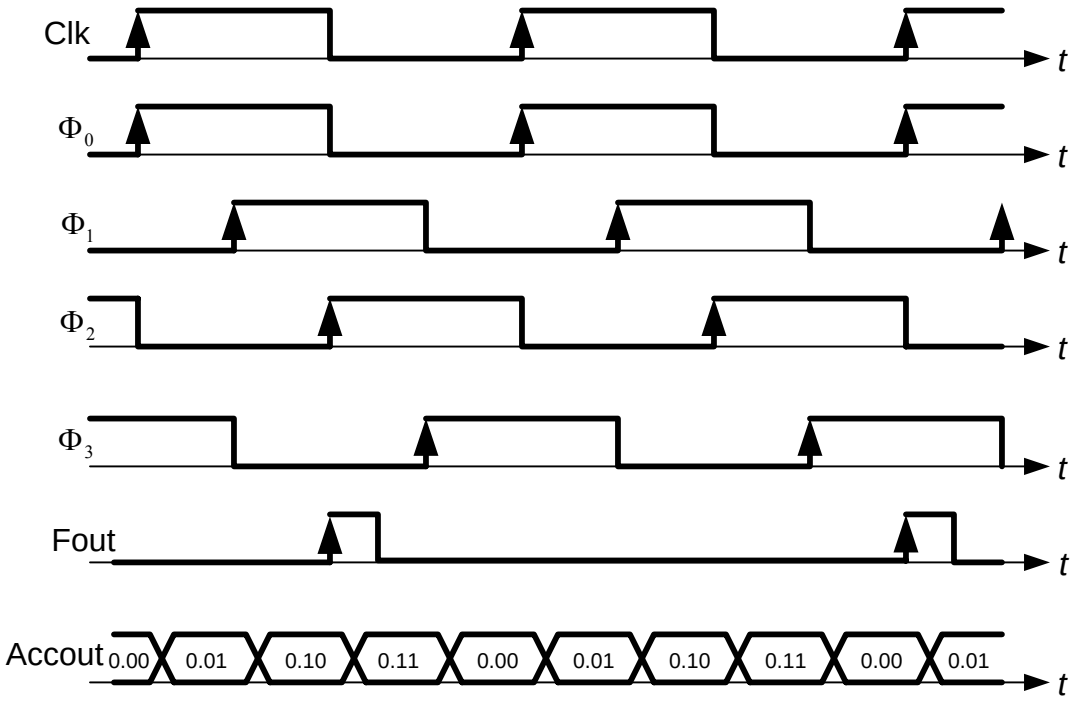


Figure 4-3. Timing diagram of the modified DDPS eliminating race condition

The total peak to peak output jitter of this circuit is the sum of the jitter coming from TGen block, the jitter in the TSel block and the phase jitter due to the finite precision of the accumulator. Of these, the phase jitter due to the accumulator may be reduced significantly by increasing its precision. The TSel jitter source is process dependent and needs to be looked at carefully. Equation (4-2) suggests that the penalty for solving the race condition problem is that F_{out} now can no longer be greater than F_{clk}. For the example of $Ph = 0.01_b$, $F_{out} = F_{clk}/1.25$. Thus, although the capability of fractional ratios was achieved, the frequency multiplication feature is lost. The DDPS architecture thus needs to be modified. The new architecture achieves this functionality.

4-2. NEW FRACTIONAL-N FREQUENCY SYNTHESIZER

A new architecture for generating fractional multiples of the input reference is shown in Figure 4-4. The multiple phases to the transition selector are provided by the ring oscillator outputs which are divided by a bank of dividers to provide equally spaced signals to the transition selector. Based on the digital input word, *Ph*, an output frequency that is a fractional multiple of F_{ref} can be synthesized.

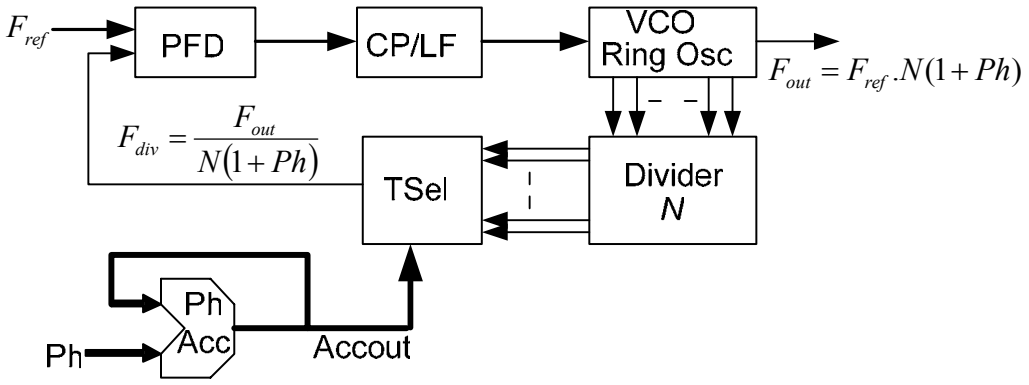


Figure 4-4. The new Fractional-N Frequency Synthesizer Architecture. A DDPS circuit in the feedback loop generates fractional multiples.

It can be simply established that $F_{out} = F_{ref}. N(1+Ph)$. Thus the digital input word *Ph* forms the fractional part of the divider modulus. As an example, referring to the frequencies listed in Table 2-2 in Chapter 2, setting $N = 8$ and $Ph = 0.25$, 0.0625 and 0.03125 provides the desired multiply ratios of 10, 8.5 and 8.25 respectively. Notice that a divide by 8 is a trivial circuit to design and the three values of *Ph* are all simple sub-multiples of 2 (2^{-2}, 2^{-4} and 2^{-5}). Generating any other frequency that is within the tuning range of the VCO but is a non-integer multiple of the reference frequency is very easily accomplished with this architecture. This architecture preserves the simplicity of the DDPS system while making use of a conventional PLL for frequency multiplication.

4-2-1. Discussion of the proposed architecture

It can be observed that in the proposed architecture, the DDPS is *not* required to generate a frequency *greater* than its input. This relaxation eliminates the potential race condition problem described in Section 4-1. The fractional part of the multiply ratio is under digital control and is limited only by the resolution of the phase accumulator. Unlike the ΔΣ based fractional-N synthesizer, there are no quantization error filtering issues to deal with. This relaxes the requirement of narrow PLL loop bandwidth. The PLL bandwidth can thus be set as high as the loop dynamics will allow so that the low frequency VCO jitter can be filtered.

It is important to examine the jitter properties of this architecture. In a PLL frequency synthesizer, the major sources of jitter are the VCO, the divider, the PFD and the charge pump. These jitter sources were depicted in Chapter 2 and their contribution to the jitter at the output were mathematically derived. In the proposed frequency synthesizer, additional jitter can be expected from the transition selector and the phase accumulator. Of these, the jitter due to the transition selector circuit is likely to be the dominant source. The time resolution of the output transitions of the transition selector and implicitly the output jitter, is determined by the time separation between the phases generated by the feedback divider. Suppose that the VCO is comprised of a 4 stage ring oscillator, running at a nominal frequency of 3.125 GHz. If the VCO is implemented as a fully differential ring oscillator, then the VCO generates 8 phases, each separated by 45^{o} in phase or 40 ps in time. When these 8 phases are divided, the divider output will have N*8 phases, each still separated by 40 ps. For a divide ratio of N=8, the divider output will have 64 phases separated by 40 ps. Hence, the peak to peak timing jitter of the system will have a lower bound of 40 ps. This lower bound may be significantly reduced by reducing the time resolution between the adjacent phases. This operation may be done

either with a phase interpolator [27] or by increasing the number of stages in the ring oscillator [63]. Both the options have their disadvantages. Increasing the number of stages in the ring oscillator is a better choice as the signals are generated inside of a high gain loop, but with increasing number of stages, mismatches and process variations can lead to the oscillator failing to get into sustained oscillations. A phase interpolator, on the other hand, can introduce mismatches between the interpolated phases. Both of the above options, while reducing the time resolution between the phases, can introduce random errors thereby degrading the jitter performance.

The lower bound of 40 ps means that the circuit implementation can not introduce more than 10 ps of jitter. Chapter 5 presents the circuit design aspects of this research with the predicted peak to peak jitter contributions. The simulated performance of the VCO shows that the jitter requirement can indeed be achieved.

4-2-2. System Design

In this section, a brief description of the high level design to arrive at the loop parameters of the system is presented. The system design begins with the determination of the most critical parameter, the achievable oscillator gain K_{vco}. From the linear analysis presented in Chapter 3, a large K_{vco} increases the noise sensitivity of the VCO and is not very desirable. Any small noise voltage at the VCO input will induce a large frequency change causing timing jitter. Another factor to consider is that during the locking process, the oscillator may experience voltages in excess of the nominal voltage. The excess voltage can drive the oscillator output to frequencies higher than the nominal. To accommodate for this process, the oscillator is designed to have about +/- 20% larger tuning range. The list of possible line codes and the resulting serial clock frequencies were tabulated in Table 2-2. Hence the oscillator must be designed such that it has a tuning range of 0.8*2.578125 = 2.0625 GHz to 1.2*3.125 = 3.7 GHz. Finally, the output

of the charge pump will have a minimum and maximum operating range based on the technology, charge pump structure and the power supply voltage. In this design, the nominal supply voltage is 1.5 V and assuming a safe operating range of 1 V, the desired K_{vco} of the oscillator is $\frac{1.6375e9}{1.0} = 1.6375GHz/V$. This value of the oscillator gain is relatively large and can lead to increased jitter sensitivity. Since the overall jitter of the synthesizer is governed mostly by the jitter of the oscillator outside the loop bandwidth of the PLL, it is desirable to minimize the K_{vco} of the oscillator. In Chapter 5, a new VCO design is described. This VCO uses a fully differential control voltage structure that allows for the above K_{vco} value to be halved. Performance results show that the achieved nominal K_{vco} is in the 800 MHz/V range, which is a significant improvement.

Figure 4-5 shows the output of a behavioral simulation to determine the loop filter parameters. The MATLAB code used in this simulation is included in the Appendix. A detailed description of the PLL parameter calculations is given in [28]. A K_{vco} of 915 MHz/V has been used in this simulation. This particular value of the oscillator gain is obtained from the circuit level simulation of the VCO. Figure 4-5 shows the values of the loop filter parameters needed for a given oscillator gain, to achieve a particular closed loop phase margin and unity gain frequency. In this simulation, a phase margin of 70 degrees is used. From the figure, for a Kvco of 915 MHz/V, the value of R1 = 2.7 kOhm, C1 = 27 pF and C2 = 0.5 pF. Chapter 5 lists the actual values used in the design for a more conservative loop behavior.

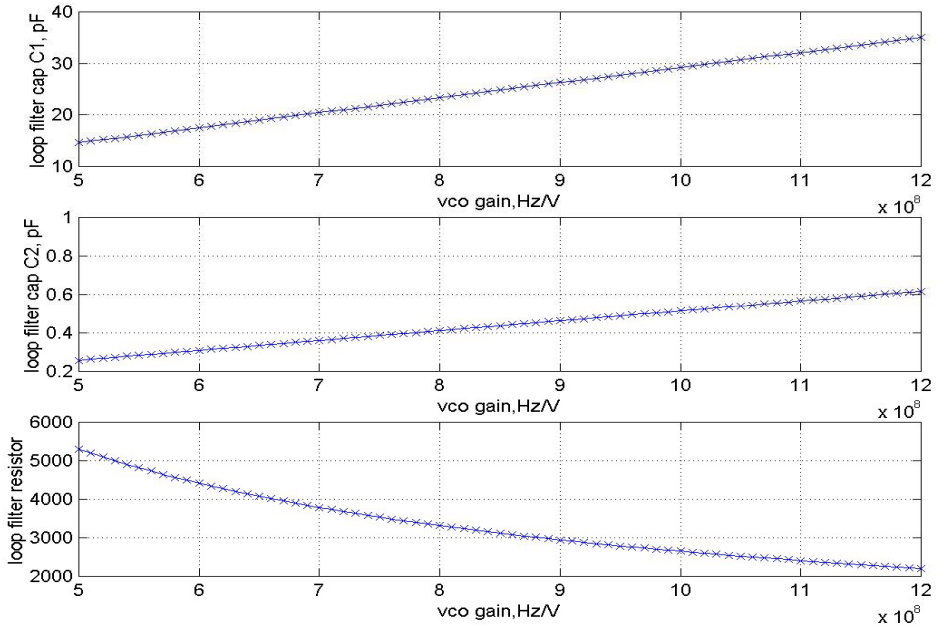


Figure 4-5. Behavioral simulation showing loop filter parameters versus K_{vco}

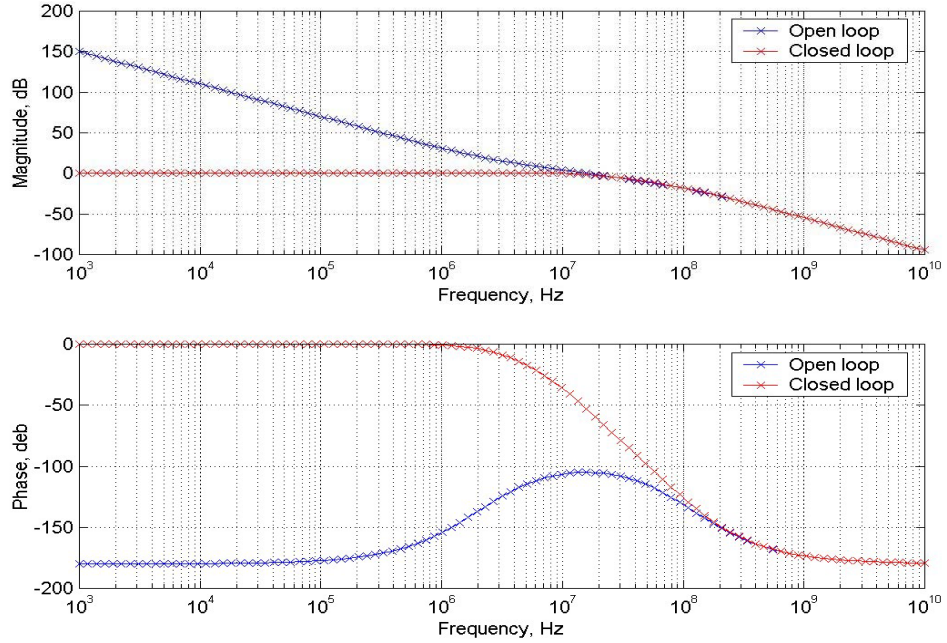


Figure 4-6. Open and closed loop responses of the PLL

These values of the loop filter are then used to determine the open and closed loop response of the PLL and a plot of the responses is shown in Figure 4-6. The simulated phase margin is 68 degrees and the unity gain frequency is 40.6 MHz.

4-3. SUMMARY

This chapter presents an architecture for a fractional-N frequency synthesizer using a direct digital period synthesis circuit. The eight phases from the four stage differential ring oscillator are divided down in frequency and a transition selector picks one of these phases under the control of a phase accumulator and propagates it to the output. Thus a pulse train of the right frequency is generated at the output of the transition selector such that an overall fractional multiplication is achieved. The chapter also discusses the jitter issues arising out of this architecture followed by a brief description of the system design aspects.

Chapter 5: Circuit Design

A new fractional-N frequency synthesizer architecture is described in Chapter 4. System level design and simulations were done to find the optimum circuit component values for integrated circuit implementation. From a circuit design point of view, the most critical parameter to determine is the achievable VCO gain, K_{vco}. This parameter is dependent on the topology of the oscillator used, the CMOS process, supply voltage, operating temperature etc. Chapter 4 also discusses the effects of a large VCO gain on the jitter performance of the synthesizer. The circuit level design of the individual blocks of the frequency synthesizer is described in detail in the next several sections.

5-1. PROCESS SELECTION

Historically, very high speed communication links have been implemented in GaAs or bipolar technologies. The higher intrinsic device operating frequencies (f_T) of GaAs and bipolar technologies are the reasons for their choice until deep sub-micron CMOS processes have started coming close to matching the speeds of the other technologies. Figure 5-1 shows the scaling trend of commercially available CMOS technology versus GaAs and bipolar processes [2]. The slopes of the figure indicates the rapid increase in CMOS process speeds with decreasing feature size. Present day CMOS processes have minimum transistor lengths as small as 65 nm with fTs approaching the 75 GHz range. Due to the commercial popularity, feasibility and economy, the 0.13 μm process was chosen for this design. This process presents certain distinct advantages in terms of speed on one end and larger operating supply voltage at the other end. A 0.18 μm technology could have possibly been used. Indeed, exploratory simulations were conducted in a TSMC 0.18 μm process. Although this process provided the larger 1.8V

(almost 2.0V) supply operation, it was found that the transistor sizes used in the VCO delay cell were small enough that the 1/f noise or the flicker noise of the devices became an issue. On the other hand, the TSMC 0.13 μm process is quite a bit faster than the required 3.125 GHz operation. However, this allows the use of much larger devices to intentionally *slow* down the oscillator. Using the larger transistors increases the gate capacitance, which in turn slows down the oscillator. The use of larger devices results in a lower 1/f noise and better jitter performance of the oscillator.

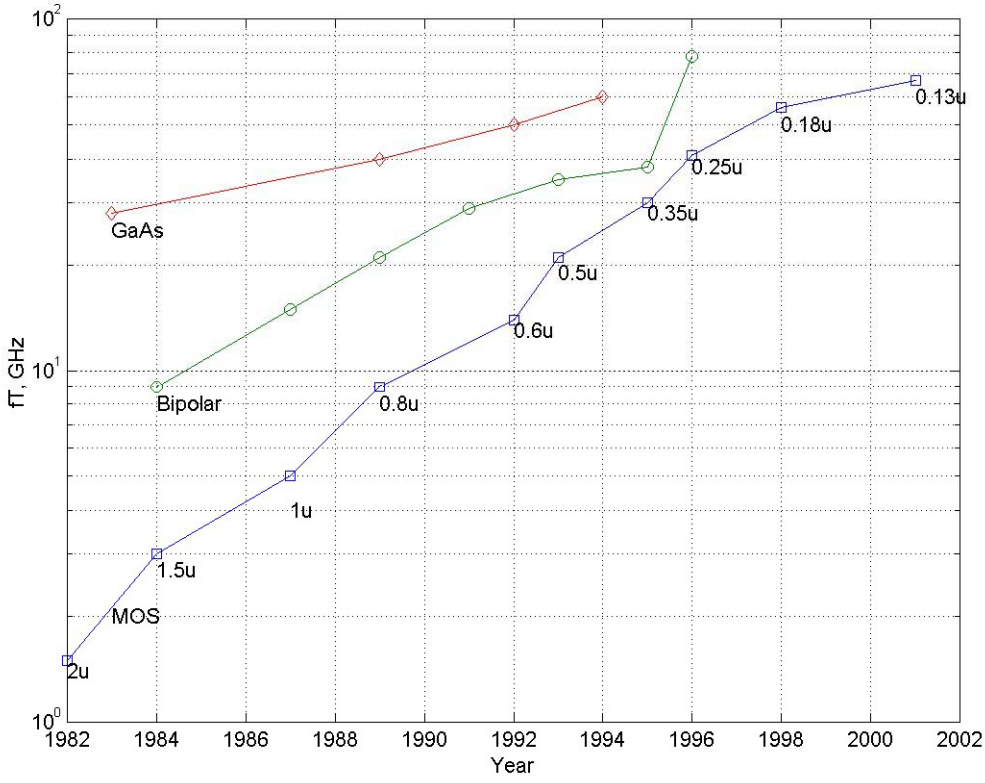


Figure 5-1. Technology fT scaling trends for CMOS, GaAs and bipolar processes. [2]

The process documentation [58] recommends a nominal power supply of 1.2V. But this design was done with a supply voltage of 1.5V after ensuring that the larger power supply would not pose a reliability issue with the process in causing transistor

breakdowns, etc. This process also allows for deep a N-well option to place the NMOS transistors inside of wells instead of placing them directly on the silicon substrate. The process provides specific transistor models for use in the design process for NMOS transistors that are intended to be inside the deep N-wells. This design was done without the use of the deep N-well option. The intention was to be able to avoid special processing steps allowing this design to be economically implemented.

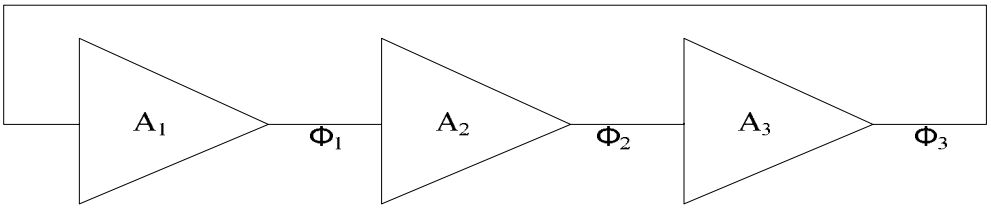
5-2. VOLTAGE CONTROLLED OSCILLATOR

The Voltage Controlled Oscillator (VCO) is the most critical building block in PLL based frequency synthesizers. The inherent jitter of the VCO is the fundamental limitation to the performance of a transceiver and its design is the most critical. The overall jitter of a PLL outside of its loop bandwidth is largely governed by the jitter produced by the VCO. Oscillators can be broadly classified into two groups: relaxation oscillators and harmonic or sinusoidal oscillators. While relaxation oscillators do not need resonating devices such as crystals, LC tank circuits and dielectric resonators to operate, they tend to have worse jitter performance for a given operating power. Harmonic oscillators can produce a near sinusoidal signal with high spectral purity with the help of the high Q resonating elements they employ.

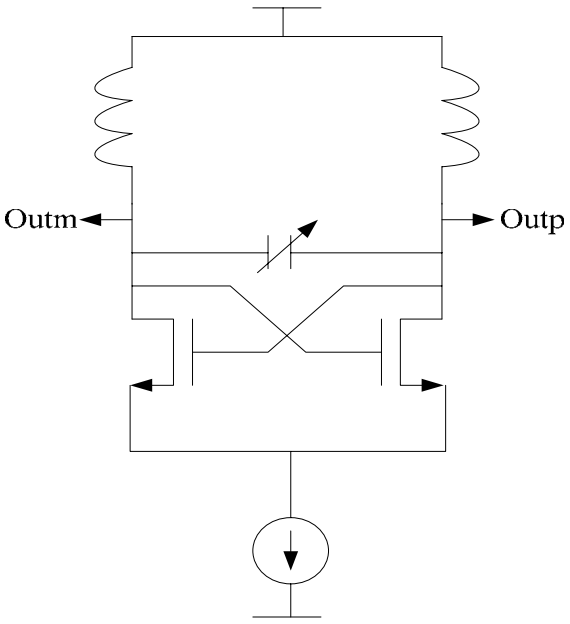
5-2-1. Oscillator Types

Integrating a high frequency, low jitter or a low phase noise VCO on a silicon die is a challenging task. The VCO is an extremely sensitive, high gain analog circuit and when operating in conjunction with other digital subsystems, it can pick up noise from different sources thereby degrading the quality of the VCO output. As a result, in silicon implementations, the most popular architectures have been ring oscillators and LC

oscillators. These two architectures are schematically depicted in Figure 5-2. The figure shows a 4 stage ring oscillator and an LC tank based oscillator. Ring oscillators are relaxation oscillators and do not employ a resonant element. They have several advantages such as wide tuning range, small area, ability to generate multiple phase outputs, etc. However, they tend to have relatively high phase noise or timing jitter. This is mainly due to the lack of a high gain tuned element like an LC tank. Still, the above mentioned advantages of ring oscillators make them a popular choice for wired serial communication systems. Wireless and RF communication systems, on the other hand, require that the local oscillator generate a pure sinusoid. LC oscillators perform better in such applications. Furthermore, in wired serial transceivers, the architectures for clock recovery/synchronization often depend on the availability of multiple phases of the serial clock. The received data is sampled using the multiple phases of the serial clock and the phase that best aligns with the bit center is chosen. For such architectures, ring oscillators are most certainly preferred. A four stage ring oscillator, for example, readily generates 8 phases of a high quality, high frequency clock. Generation of multiple phases is a much harder task in LC oscillators. For example, [5-2] describes an LC oscillator that can produce quadrature outputs. The fractional-N frequency synthesizer developed in this research relies on the availability of multiple phases of the high frequency clock and for that purpose, a ring oscillator is chosen. A more detailed look at the issues related to ring oscillator is presented further in this chapter.



(a) A 3 stage Ring Oscillator



(b). An LC based oscillator

Figure 5-2 Popular Oscillator types in MOS technology.

5-2-2. Oscillator Theory and Modeling

A simple electronic oscillator produces a periodic output, usually in the form of a voltage waveform. As such, the circuit has no input while sustaining output oscillations. This behavior of an oscillator follows from negative feedback system theory. Consider the unity gain negative feedback circuit shown in Figure 5-3. It can be derived that,

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{H(s)}{1+H(s)} \tag{5-1}$$

If the amplifier experiences so much phase shift at certain frequencies such that the

feedback becomes positive, then oscillation can occur. More accurately, if for $s = j\varpi$, $H(j\omega) = -1$, then the closed loop gain approaches infinity at the frequency ω. Under this condition, the circuit amplifies its own noise components at frequency ω, causing oscillations to occur [1].

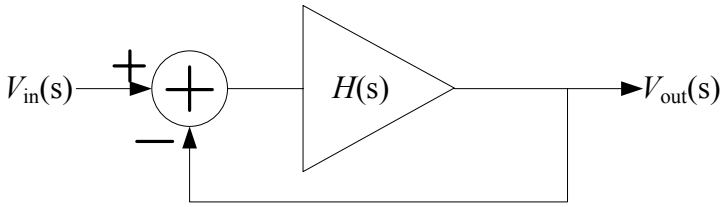


Figure 5-3. A negative feedback system exhibiting oscillatory behavior.

For oscillation to begin, a loop gain of unity or greater is necessary. This can be seen from following the signal around the loop over many cycles and expressing the amplitude of the subtractor's output in Figure 5-2 as a geometric series:

$$V_x = V_o + |H(j\varpi)|V_o + |H(j\varpi)|^2 V_o + |H(j\varpi)|^3 V_o + ... \tag{5-2}$$

If $|H(j\varpi)| > 1$, then the above summation diverges whereas if $|H(j\varpi)| < 1$,

$$V_x = \frac{V_o}{1 - |H(j\varpi)|} < \infty \tag{5-3}$$

In summary, if a negative feedback circuit has a loop gain that satisfies two conditions:

$$|H(j\varpi)| \geq 1 \tag{5-4}$$

and $$\angle H(j\varpi) = \pi \tag{5-5}$$

then the circuit can oscillate at frequency ω. These two conditions together are called the *Barkhausen criteria* and are necessary but not sufficient for a circuit to oscillate [1]. In order to ensure oscillation under all process (P), supply voltage (V) and temperature (T)

variations (commonly referred to as PVT corners), the loop gain is typically chosen to be two or three times the required value.

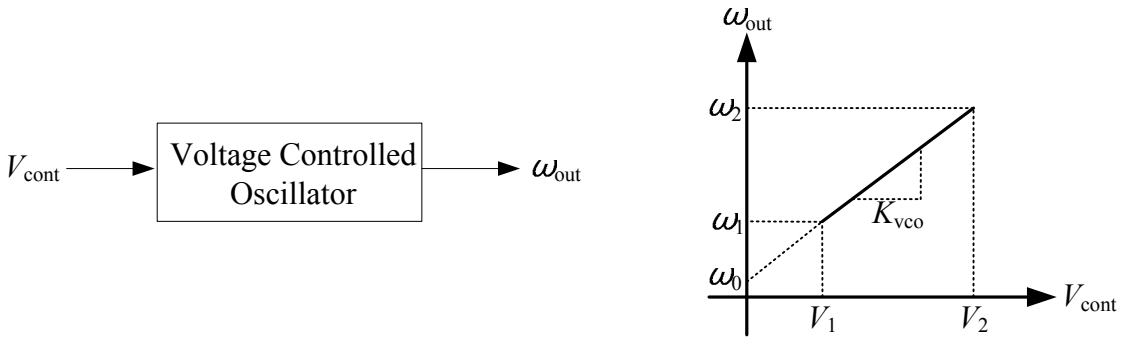


Figure 5-4. Definition of a VCO.

An ideal VCO is a system whose output frequency is a linear function of its control voltage as shown in Figure 5-4. Mathematically, it is modeled by the relation,

$$\varpi_{out} = \varpi_o + K_{vco} * V_{cont} \tag{5-6}$$

Where: ω_o represents the *free running frequency* of the oscillator and K_{vco} represents the *sensitivity* of the oscillator. From Equation (5-6) and Figure 5-4, it is evident why the K_{vco} needs to be as small as the loop dynamics will allow. Furthermore, the frequency of the output signal is expressed as the time derivative of its phase, as given in Equation (5-7).

$$\varpi = \frac{d\phi}{dt} \tag{5-7}$$

This equation indicates that, if the frequency of the waveform is known as a function of time, then its phase can be computed as

$$\phi = \int \varpi . dt + \phi_o \tag{5-8}$$

Since for a VCO, $\varpi_{out} = \varpi_o + K_{vco} * V_{cont}$, the output of the VCO can be calculated as

$$\begin{aligned} V_{out}(t) &= V_a \cos(\int \varpi_{out} . dt + \phi_o) \\ &= V_a \cos(\varpi_o t + K_{vco} \int V_{cont} . dt + \phi_o) \end{aligned} \tag{5-9}$$

This equation has been used in the analysis and modeling of the VCO and the PLL in the

earlier chapters. The initial phase Φ_o is not critical to the analysis and can be ignored. Once the VCO is placed inside a PLL, then only the integral term in Equation (5-9) becomes important for analysis. This term is referred to as the *excess phase* of the VCO [1]. In the Laplace domain, the VCO can be modeled by the control voltage as its input and the excess phase as the output, expressed as

$$\phi_{ex} = K_{vco} \int V_{cont} .dt \tag{5-10}$$

such that,

$$\frac{\phi_{ex}}{V_{cont}} = \frac{K_{vco}}{s} \tag{5-11}$$

One critical assumption that has been made in the above derivation is that the VCO output is sinusoidal. For systems that generate a clock signal, the output may be closer to a square wave than a sinusoid, thus containing significant higher harmonic content. Under such situations, the VCO output can be expressed as a Fourier series [1]. Also, for a square wave, if the fundamental frequency is changed by Δf, the frequency of the 2nd harmonic changes by 2Δf and so on. Thus if the control voltage changes by ΔV, then the frequency of the fundamental changes by $K_{vco}\Delta V$, that of the 2nd harmonic changes by 2 $K_{vco}\Delta V$ and so on. The VCO output can then be expressed as,

$$V_{out}(t) = V_1 \cos(\varpi_o t + K_{vco} \int V_{cont} .dt + \phi_1) + V_2 \cos(2\varpi_o t + 2K_{vco} \int V_{cont} .dt + \phi_2) + ... \tag{5-12}$$

The above equation suggests that the harmonics of the oscillator output can be readily modeled and for this reason, all system analysis is usually done with the assumption that the oscillator produces a single frequency at its output.

5-2-3. Ring Oscillator Based VCO

A symbolic representation of a ring oscillator was shown in Figure 5-1. A differential implementation of the same oscillator is shown in Figure 5-5. Almost all high

performance ring oscillator VCOs are implemented with fully differential circuitry to take advantage of their superior PSRR and CMRR performance and inherent noise immunity [45],[60]-[63]. The ring oscillator shown in Figure 5-5 is a cascade of 4 fully differential delay stages or trans-conductance stages. The delay time t_d of each stage can be controlled by varying the bias current of the delay cell because the delay is given by,

$$t_d = \frac{V_{sw}.C_l}{I_b} \tag{5-13}$$

Where: I_b is the bias current of each delay cell, C_l is the total load capacitance at the output of each delay cell and V_{sw} is the voltage swing.

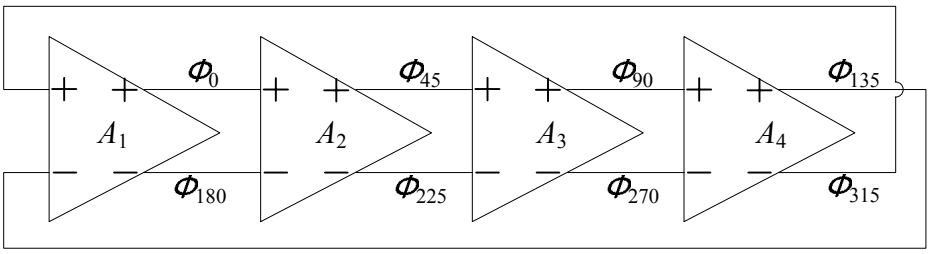


Figure 5-5. A 4 stage fully differential ring oscillator.

To calculate the design parameters and the oscillation frequency, the 4 stage ring oscillator is modeled in its single ended form as shown in Figure 5-6. In this figure, R and C represent the output resistance and the load capacitance of each delay cell, respectively. Gm represents the linear trans-conductance of the delay cell [25]. This circuit will oscillate at a frequency of ω_o if each stage has an open loop gain of *at least* 1 and a phase shift of exactly 45 degrees, taking phase inversions into account. With the help of the Barkhausen criteria established earlier and writing the loop gain equation, it can be shown that

$$\varpi_o = \frac{1}{4R.C} \text{ and } G_m R = \sqrt{2}$$

These two relations establish the design criterion for the delay cell and the VCO in order to ensure sustained steady state oscillations. In practice, the DC gain is designed to be slightly larger than the minimum required to ensure oscillations under all conditions. The delay cell designed for this research achieves a DC gain of 2 across the PVT corners.

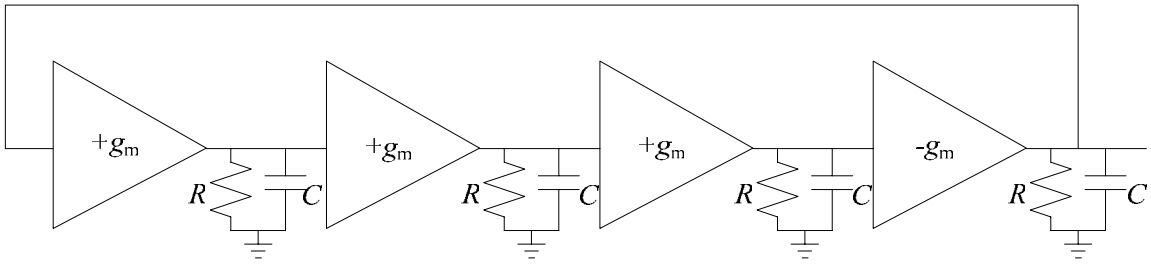


Figure 5-6. Linearized model of the 4 stage ring oscillator based VCO.

The schematic of the delay cell used in the ring oscillator is shown in Figure 5-7. It consists of a source coupled NMOS input pair MN0 and MN1 with their load resistors R0 and R1. PMOS transistors MP0 and MP1 operate in their linear region and provide continuous time common mode feedback to keep the DC level of the outputs constant. This is a unique feature in this circuit and addresses a major drawback of most delay cells published in the literature. Most designs vary the tail current of the delay cell as a function of the control voltage to achieve output frequency variation. The drawback of this method is that the DC level at the outputs of the delay cell also varies and can cause common mode range problems in the circuit. To avoid the common mode range issue, the circuits are either operated with very limited dynamic range (tail current variation) or a parallel control path is provided that adjusts for the DC level variation. In such designs,

extreme care must be paid to make sure the two paths don't fight with each other to control the delay element. The delay cell designed in this research eliminates this problem with the common mode feedback feature.

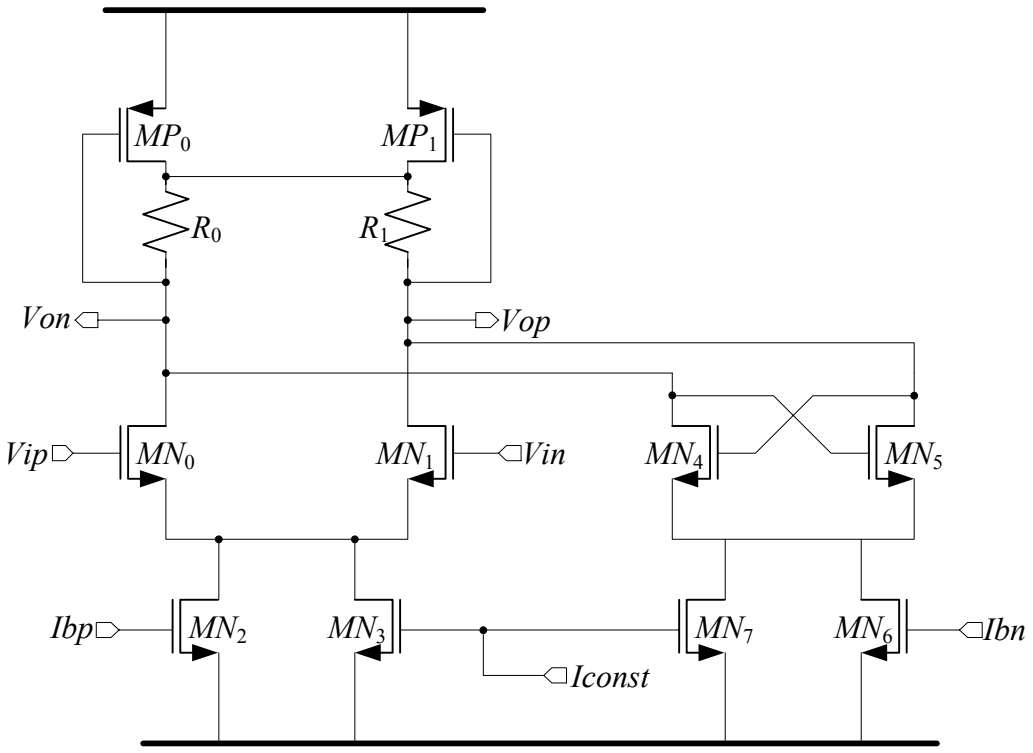


Figure 5-7. Schematic of the delay cell.

Another feature incorporated in this design is the delay variation with positive feedback [1]. The NMOS cross coupled pair MN_4 and MN_5 form a negative resistance pair that appear in parallel with the constant resistors R_0 and R_1. Suppose that this cross coupled pair presents a small signal resistance of $-R_n$ and let $R_0 = R_1 = R_p$. Then the equivalent resistance at the output nodes of the delay cell is given by $\frac{R_p R_n}{R_n - R_p}$, which is more positive if $|R_n| > |R_p|$. This idea has been incorporated into each delay cell. The

concept of a cross coupled MOSFET pair introducing negative resistance is derived using the conceptual diagram in Figure 5-8(a) and its small signal equivalent in Figure 5-8(b).

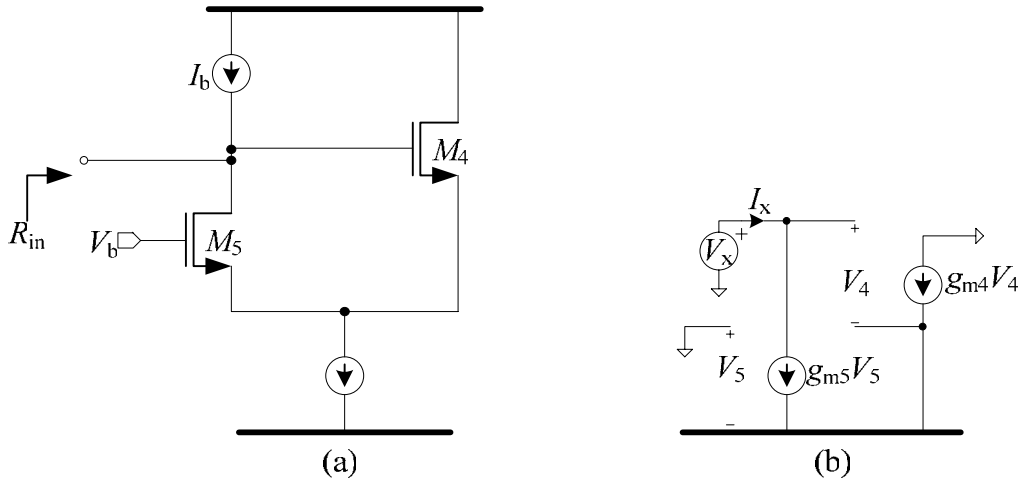


Figure 5-8 (a) and (b). A source coupled MOSFET pair to create negative resistance. Its small signal equivalent circuit.

From the equivalent circuit,

$$I_X = g_{m4}V_4 = -g_{m5}V_5 \tag{5-14}$$

and

$$V_X = V_5 - V_4 = -\frac{I_X}{g_{m4}} - \frac{I_X}{g_{m5}} \tag{5-15}$$

From which, the equivalent input resistance can be calculated to be

$$\frac{V_X}{I_X} = -\frac{2}{g_m} \tag{5-16}$$

Where: $g_{m4} = g_{m5} = g_m$ has been assumed. It must be observed that the negative resistance is an *incremental* quantity, indicating that if the applied voltage *increases*, the current drawn by the circuit *decreases* [1]. Referring to the delay cell schematic in Figure 5-7, as the total tail current in the cross coupled branch increases (this is the sum of the drain currents of MN_6 and MN_7), the small signal differential resistance $-2/g_m$ becomes

less negative and the equivalent resistance at the output nodes, given by $R_p / 1 - g_m R_p$ increases thereby lowering the output frequency. The delay cell incorporates *differential* current steering to maintain a fairly constant output swing. This is achieved by varying the gate voltage nodes *ibp* and *ibn* differentially. The total tail current in each of the source coupled pairs MN_0-MN_1 and MN_4-MN_5 is given by the sum of a constant currents produced by MN_3 and MN_7 respectively and the controlled currents produced by MN_2 and MN_6, respectively.

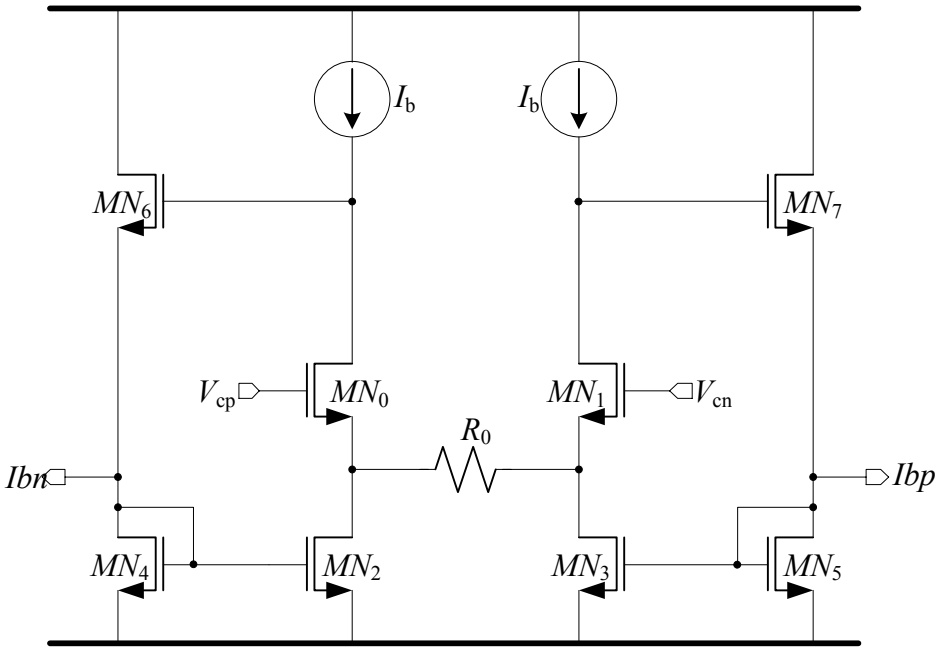


Figure 5-9. Differential control voltage generator for the VCO

The differential gate voltage control for the delay cell *ibp* and *ibn* is generated using the circuit shown in Figure 5-9. It consists of a differential NMOSFET pair MN_1 and MN_2 with their sources coupled through a constant resistor R_0. The inputs to this circuit are the nodes *vcp* and *vcn* which are the differential outputs of the charge pump and the loop filter. This is another unique feature of this design. Most of the PLL designs

reported in the literature use a single ended control for the VCO [63], [65], [66]. Even if the loop filter itself is fully differential, a differential to single ended converter is used to convert the differential voltage to an equivalent single ended current or voltage before it is used to control the VCO. This mechanism greatly reduces the noise immunity of the design, reduces the dynamic range of the VCO and defeats the purpose of having a fully differential loop filter.

By maintaining the complete signal path fully differential, as is the case in the present design, it can be ensured that most of the noise sources appear as common mode signals and do not introduce jitter at the VCO output. Referring to Figure 5-9, at quiescent condition where $vcp = vcn$, the current through the resistor R_0 is zero. Any Δv differential applied to the circuit will cause a corresponding Δi to flow in R_0. This differential current should flow through either MN_6 or MN_7 and the gate voltage, *ibn* or *ibp*, respectively will change to accommodate the current change, thus establishing a differential control voltage for the VCO.

5-2-4. Performance Results

Transistor level design and simulation was done using the Cadence Analog Artist™ environment and HSpice™ simulator. The HSpice model files also allow statistical parameter variations which can be used in running Monte Carlo simulations on the circuit. The Monte Carlo simulations were run for transistor mismatches and Vt mismatches. Figure 5-10 shows two of the output phases of the ring oscillator under steady state oscillations. The oscillations are purely sinusoidal and this is desirable to minimize noise injection into the power supplies. Simulations were run across process and temperature corners and the oscillation behavior stayed consistent. The circuit exhibited steady state oscillations in all corners indicating sufficient loop gain.

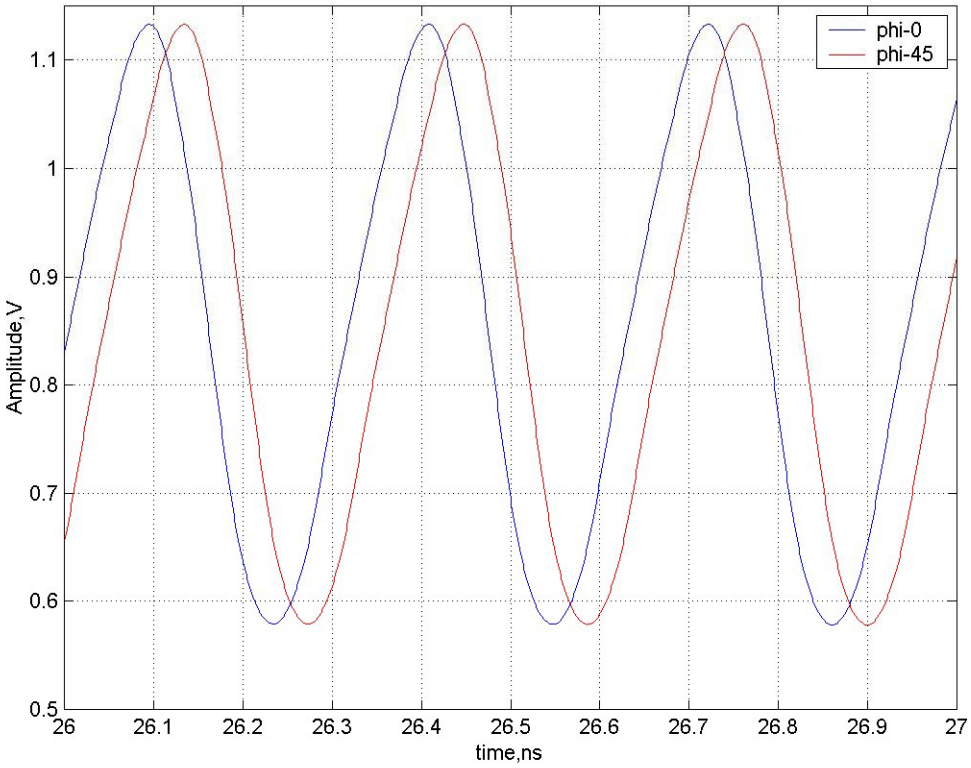


Figure 5-10. Steady state oscillation outputs of the VCO showing two adjacent phases

Figure 5-11 shows the variation of the output frequency versus the differential control voltage across PVT corners. The circuit exhibits a very linear relation with respect to the control voltage. This indicates that the differential voltage to current converter is also very linear over the process and temperature variations. The fully differential nature of the control enables this circuit to have a K_{vco} *half* as much as its single ended counterparts. This is a significant advantage that this oscillator and design has achieved. A smaller oscillator gain for the same control voltage range means less jitter sensitivity for the overall system. The fully differential signal path also ensures better power supply and common mode noise immunity. It can also be observed that the

overall slope of the curves stays relatively consistent across process and temperature variations resulting in relatively small K_{vco} variation.

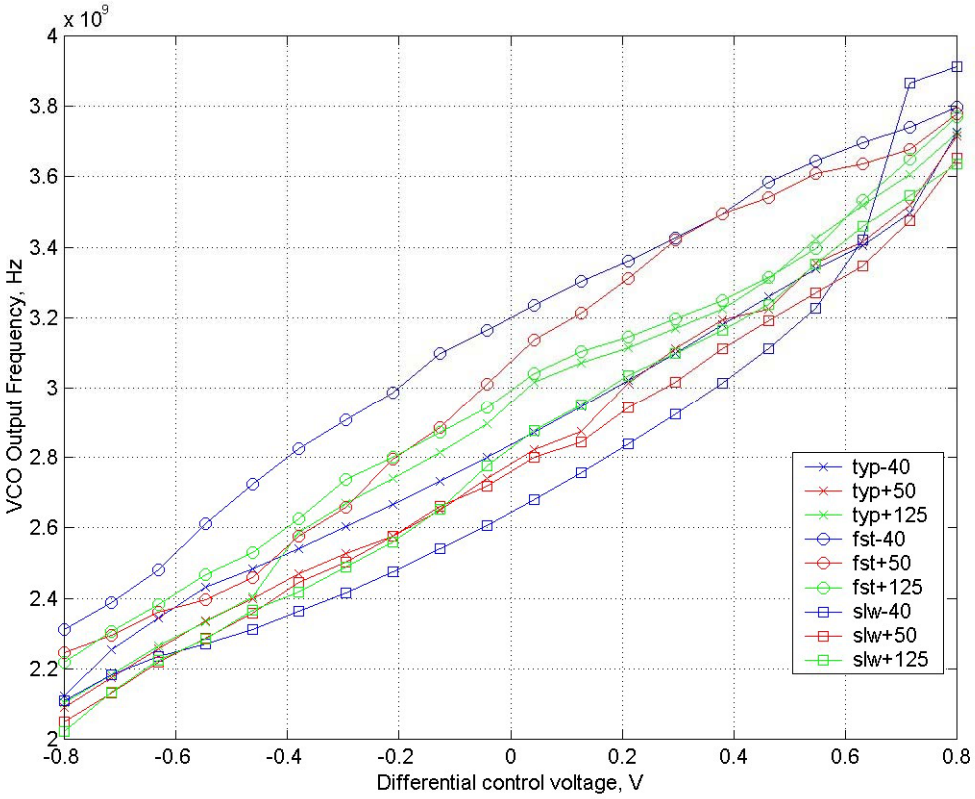


Figure 5-11. VCO output frequency vs control voltage across PVT corners

Figure 5-12 shows the output common mode voltage across PVT corners. The overall common mode variation is within 100mV for the most part as indicated in the figure. The results indicate that the common mode feedback circuit is robust across process and temperature variations and maintains optimal common mode levels at the VCO outputs. Any adjustment in the output common mode can be done by changing the sizes of the PMOSFETs in the VCO delay cell. A larger W/L ratio moves the common mode up towards power supply and vice versa.

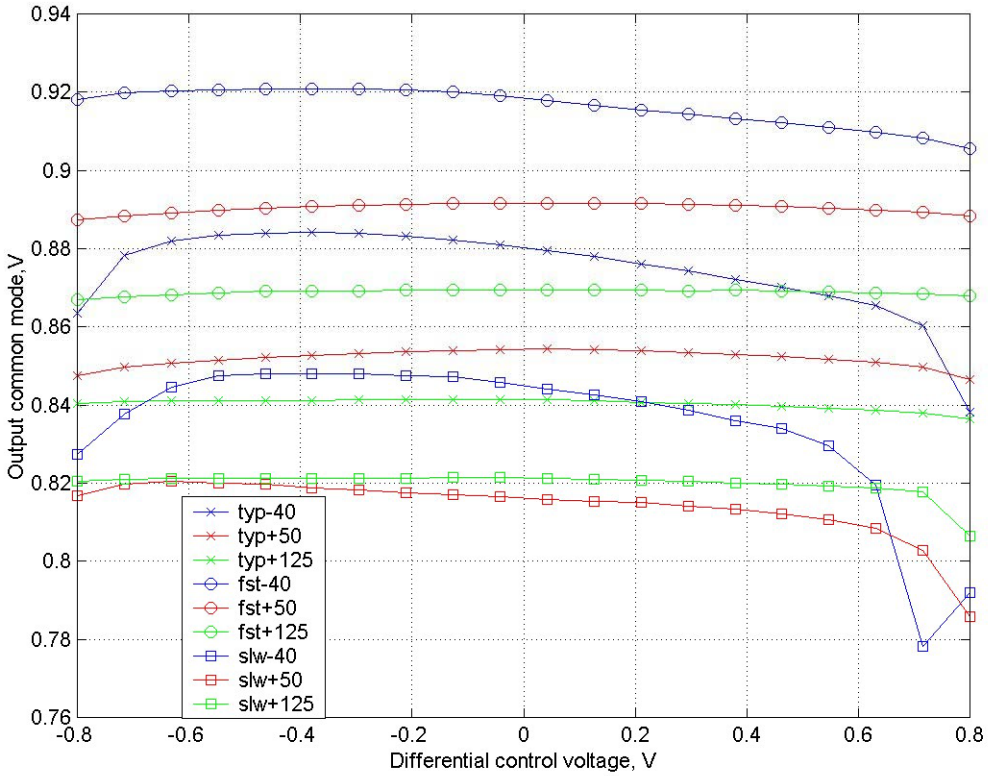


Figure 5-12. VCO output DC level across PVT corners

Figure 5-13 shows the output peak to peak voltage variations in response to PVT variations. The differential control ensures that the peak to peak outputs are fairly stable across process corners. It is important to ensure this behavior in oscillators. A relatively constant common mode level and peak to peak amplitude results in a consistent region of operation for the active elements in the circuit, thereby ensuring a uniform performance from the oscillator. On the other hand, large variations in output common mode and amplitude can potentially move the transistors from one region of operation into another (saturation to cut off, for example) and cause significant variations in the performance parameters.

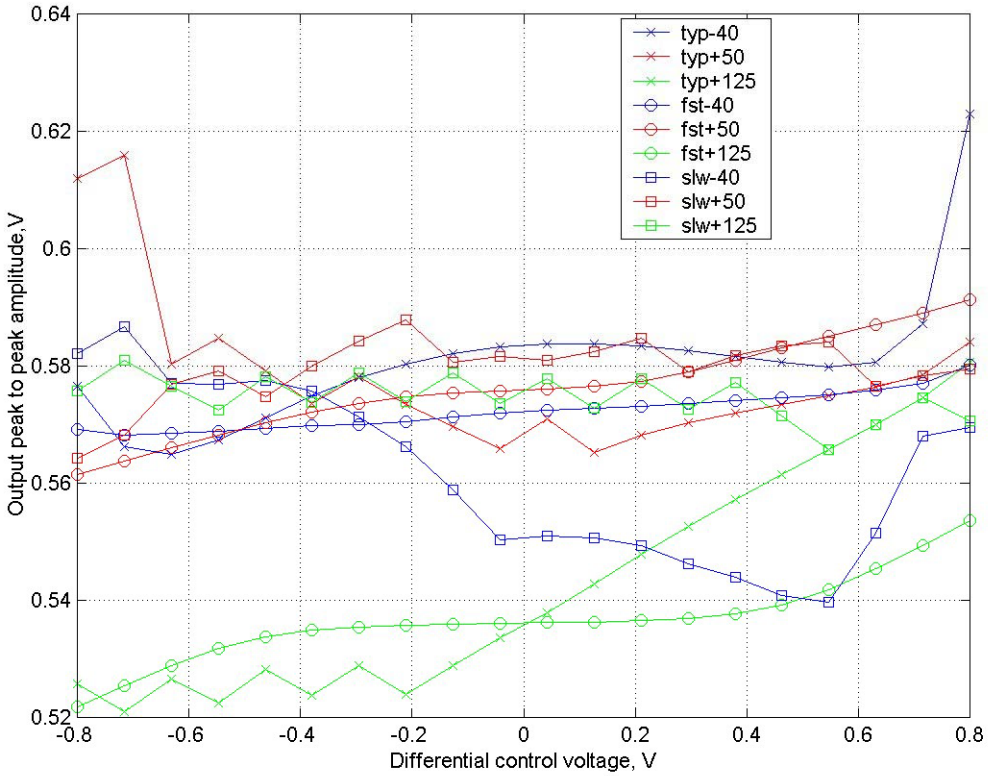


Figure 5-13. VCO output swing across PVT corners

The oscillator gain Kvco is calculated from Figure 5-11 and the minimum and the maximum values are obtained to be 750 MHz/V and 917 MHz/V respectively. The maximum K_{vco} variation is about 167 MHz/V across all variations. Most designs, in response to large variations in K_{vco}, provide for loop filter parameter adjustments using digital programmability. Common practice is to have multiple capacitors that are switched in or out as needed. In such cases, the switch resistance needs to be taken into consideration while simulating the system performance and the switch resistance variation across PVT corners can lead to additional sources of error. The variation of the

K_{vco} achieved in this design is much smaller compared to others reported in literature and additional loop filter elements are not needed.

Figures 5-14 shows the DC power supply sensitivity and noise rejection of the VCO across process and temperature variations. The graph plots output frequency of the VCO measured under nominal power supply, a +10% step in the power supply, a -10% step in the power supply and back to nominal power supply. This graph gives a good indication of the expected peak to peak jitter of the VCO. The power supply noise is likely to be largest contributor of the cycle to cycle jitter of the VCO. This number can be estimated from Figure 5-14. For example, in the typical process corner (labeled TT) and 50 degC, the maximum variation in the period of the output is 1/3.199624e9 – 1/3.234370e9 = 3.357 ps. At 125 degC, the same number is 0.609 ps. These numbers indicate very good power supply noise immunity. With very careful chip and board layout, actual silicon performance close to these numbers can be achieved. The maximum variation is observed in the fast, fast, -45 degC corner, which exhibits a peak to peak variation of 5.71 ps.

Figure 5-15 shows the AC power supply rejection of the VCO. In any PLL design, the major source of high frequency power supply noise is likely to be the feedback divider. In this design, the Fvco is divided by 8 and the major source of noise is likely to be at a frequency of $F_{vco}/8$. Accordingly, Figure 5-16 plots the change in F_{vco} due to a noise source on the power supply at a frequency of $F_{vco}/8$. The maximum frequency deviation occurs at Fast/Fast process corner at 50 degC and the resulting peak to peak timing variation or timing jitter is 6.3 ps. It can be observed that for a typical process corner, both the DC and the AC power supply rejection performance of the VCO is excellent and exhibits very little timing jitter.

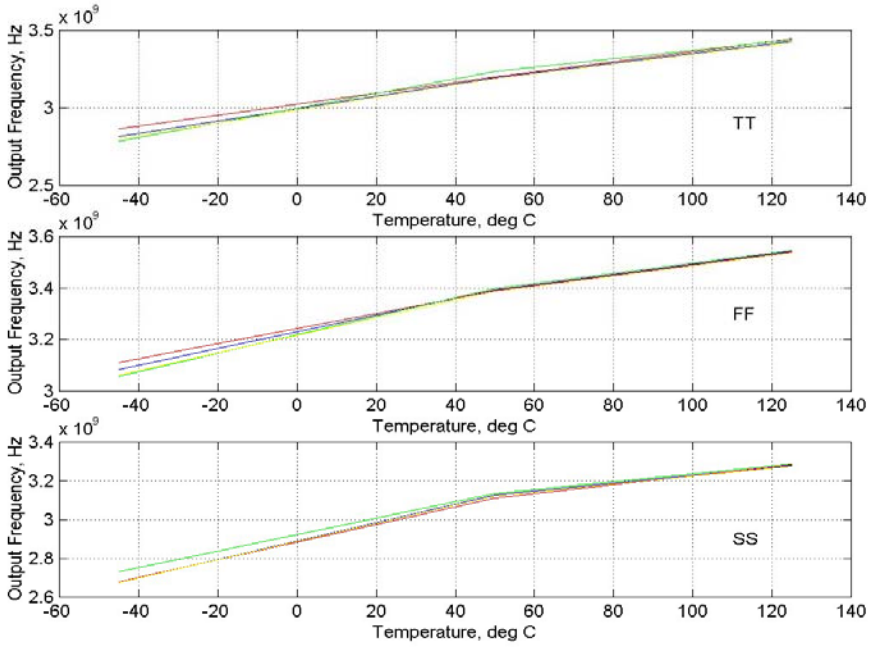


Figure 5-14. Output frequency variation for a 10% step jump in power supply

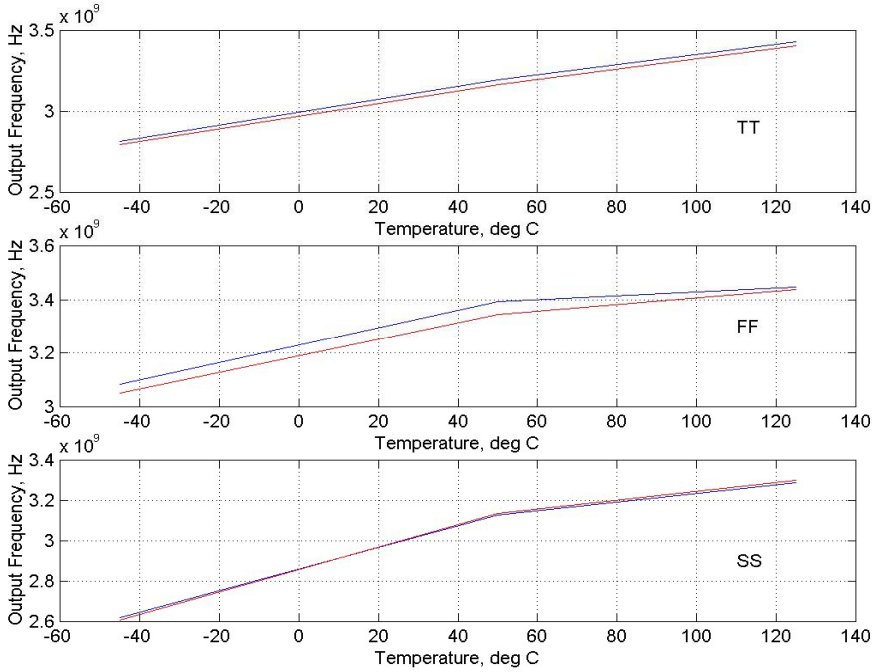


Figure 5-15. Output frequency variation for a 10% AC noise at $F_{vco}/8$

Table 5-1 below summarizes the performance and the estimated silicon area for the VCO. The estimated area includes the VCO delay cells, the VCO buffers and the differential voltage to current converter.

Table 5-1. VCO performance data and silicon area estimates

Nominal Frequency	3.125 GHz
Tuning Range	2.1 – 3.8 GHz
Control Voltage Range	1.6 V differential
Maximum Kvco variation	165 MHz/V
Maximum power supply induced p-p jitter	5.71 ps (DC)/6.3 ps (AC)
Power dissipation	12 mW (VCO+VtoI), 24 mW (VCO+VCO Buffer+VtoI)
Die area estimate	10,000 μm^2

5-2-5. Discussion on VCO jitter

In an ideal oscillator, the spacing between adjacent transitions is constant [72]. In practice, however, the transition spacing will vary. The variation is called timing jitter and increases with measurement interval ΔT (i.e., the time delay between the reference and the observed transitions). This increase is due to the fact that in ring oscillators, the jitter *accumulates* because any uncertainty in one transition affects all the following transitions and effect persists indefinitely [72]. The classification of timing jitter was explained in Chapter 2. This section discusses the effect of device noise sources on the random jitter in an oscillator. For a detailed work on this topic, refer to [72]. Generally, the jitter introduced in a ring oscillator by individual delay stages is assumed to be

uncorrelated with respect to each other and the total jitter is the rms sum of individual jitter sources. For ring oscillators with identical delay stages, the variance of the jitter is given by $m\sigma_s^2$, where m is the number of transitions during ΔT and σ_s^2 is the variance of the jitter introduced by one stage during one transition. Since m is proportional to ΔT, the standard deviation of the jitter after ΔT seconds is

$$\sigma_{\Delta T} = \kappa\sqrt{\Delta T} \tag{5-17}$$

Where: κ is a proportionality constant determined by circuit parameters [72]. When considering correlated noise sources such as substrate, supply noise and 1/*f* noise, the standard deviations add directly as opposed to an rms sum. Therefore the standard deviation of the jitter after ΔT seconds is given by,

$$\sigma_{\Delta T} = \zeta\Delta T \tag{5-18}$$

Where: ζ is another proportionality constant. Device thermal noise sources are uncorrelated noise sources that follow Equation (5-17) whereas correlated noise sources mentioned above follow Equation (5-18). In practice, both noise sources exist in a circuit and a log-log plot of the timing jitter $\sigma_{\Delta T}$ versus the measurement delay ΔT for an oscillator in open loop will show regions with slopes of ½ and 1, as shown in Figure 5-16. A study of timing jitter for single ended and differential ring oscillators is presented in detail in [72]. In summary, for a single ended ring oscillator, the jitter proportionality constant κ is given by the relation,

$$\kappa \approx \sqrt{\frac{8}{3\eta}}\cdot\sqrt{\frac{kT}{P}\cdot\frac{V_{DD}}{V_{char}}} \tag{5-19}$$

Where: η is a proportionality constant typically close to one, k is the Boltzmann's constant, P is the total power dissipation, V_{DD} is the power supply voltage and V_{char} is the characteristic voltage which is a function of V_{gs}-V_t of the device. For a fully differential ring oscillator, the expression for κ is given by,

$$\kappa \approx \sqrt{\frac{8}{3\eta}} \cdot \sqrt{N \cdot \frac{kT}{P} \cdot \left(\frac{V_{DD}}{V_{char}} + \frac{V_{DD}}{R_L I_{tail}} \right)} \tag{5-20}$$

Where: N is the number of stages in the oscillator, R_L is the load resistance and I_{tail} is the tail current of the differential pair used in the oscillator delay cell.

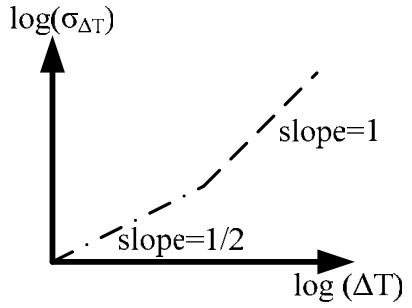


Figure 5-16. RMS jitter versus measurement time on a log-log plot

From a design consideration point of view, Equations (5-19) and (5-20) reveal that for a single ended ring oscillator, the jitter is expected to be *independent* of the number of stages in the ring. For a N stage ring oscillator, the jitter for a differential ring oscillator is likely to be *larger* than that of a single ended equivalent for same N, P and F_{osc}. For both types, increasing the operating power *decreases* the jitter. Also, operating the devices with larger *on* voltages also achieves lower jitter.

In integrated implementations, fully differential oscillators are still the choice due to lower noise sensitivity as well as lower noise injection into other circuits. For the best jitter performance, a single ended design with its own quiet power supply derived from an on-chip voltage regulator is preferred. If an on-chip regulator is not an option, a fully differential design is preferred. These issues influenced the use of a four stage ring oscillator in this design.

5-3. PHASE FREQUENCY DETECTOR

A sequential phase frequency detector is most commonly used in frequency synthesizer applications and one has been designed in this research. The PFD has the ability to detect both phase and frequency differences between two signals, thus increasing the acquisition range and the locking speed of PLLs [1]. Another advantage of the synchronous PFD is that since the D flip flops are rising edge triggered, it is not necessary to have a 50% duty cycle on the two clock inputs to the PFD.

Top level implementation of the PFD is shown in Figure 5-17. It consists of two D Flip Flops, implemented using CMOS Current Mode Logic (CML), followed by CML to CMOS converters, a delayed reset generator and up/down signal generators. The schematic of the CML flip flop used in this design is shown in Figure 5-18. The use of CML based circuit design ensures fully differential circuit operation and reduced noise injection into the power and the ground lines. The update rate of the PFD is equal to the reference frequency and a fully CMOS implementation will inject a lot of noise at the reference frequency into the power supply and ground lines. A CML implementation, on the other hand, operates fully differentially and to a first order, the power and ground lines stay quiet. The penalty for using CML flip flop is increased power consumption. For example, each CML flip flop used in the PFD design consumes 400 μA of operating current which is about an order of magnitude larger than a typical CMOS counterpart.

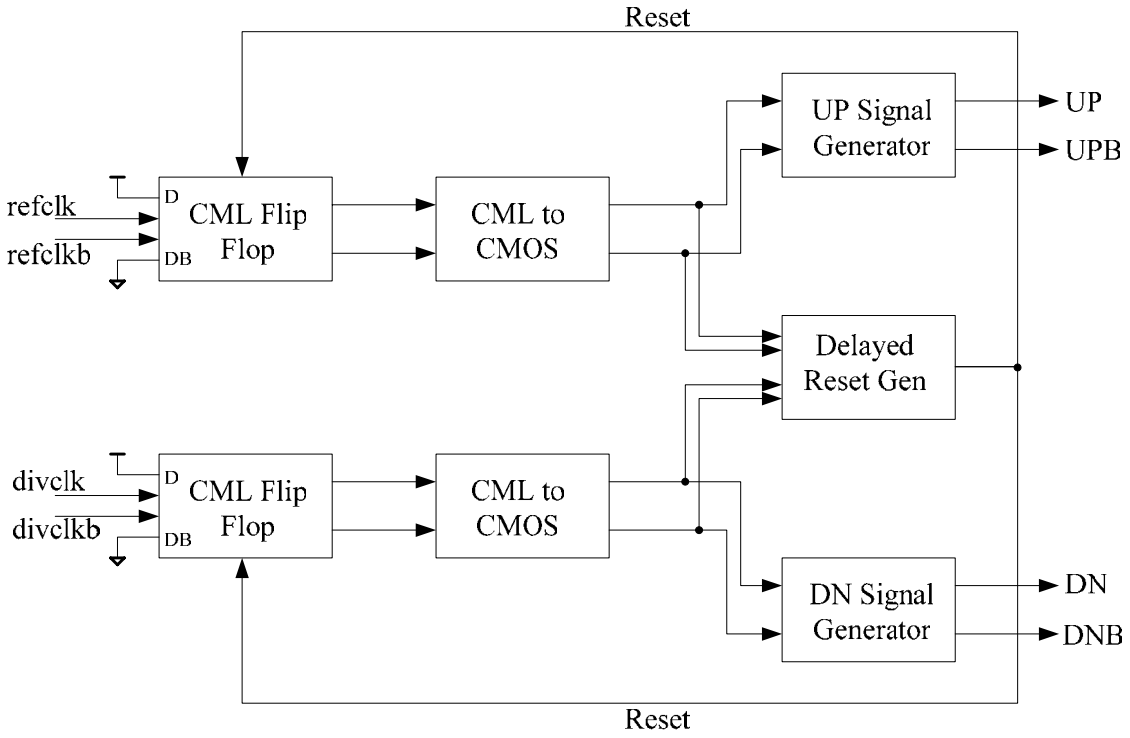


Figure 5-17. Top level schematic of the Phase Frequency Detector

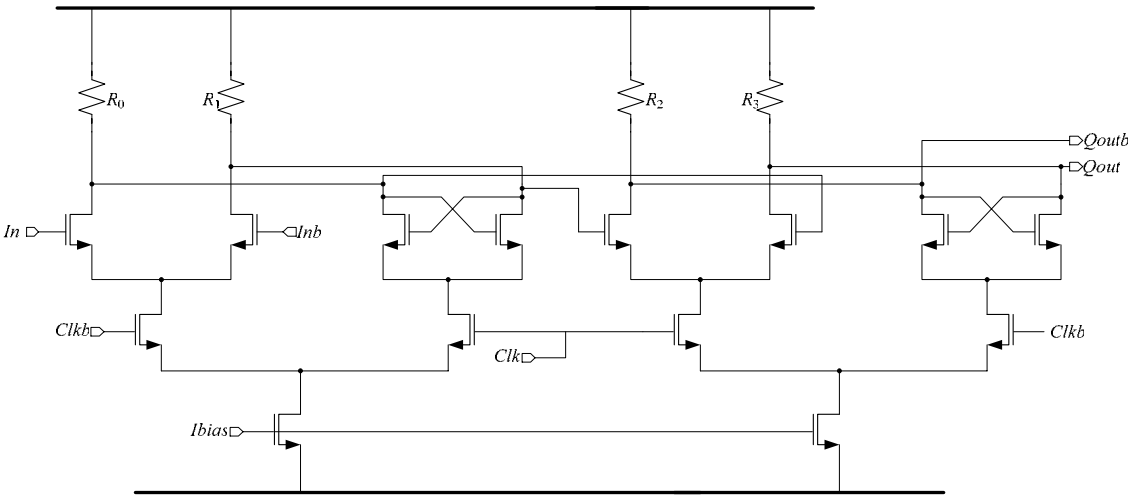


Figure 5-18. Schematic of the CML Flip Flop used in the PFD

The operation of the PFD can be explained with the help of Figures 5-19(a) and (b). The rising edge of the refclk sets the UP signal. The rising edge of the divclk sets the DOWN signal. As soon as both the UP and the DOWN signals go high, the AND gate generates the reset signal to reset both the Flip Flops, thus resetting the UP and DOWN signals. The duration for which the UP and the DOWN signals will be high depends on the delay associated with the AND gate and reset path of the Flip Flop.

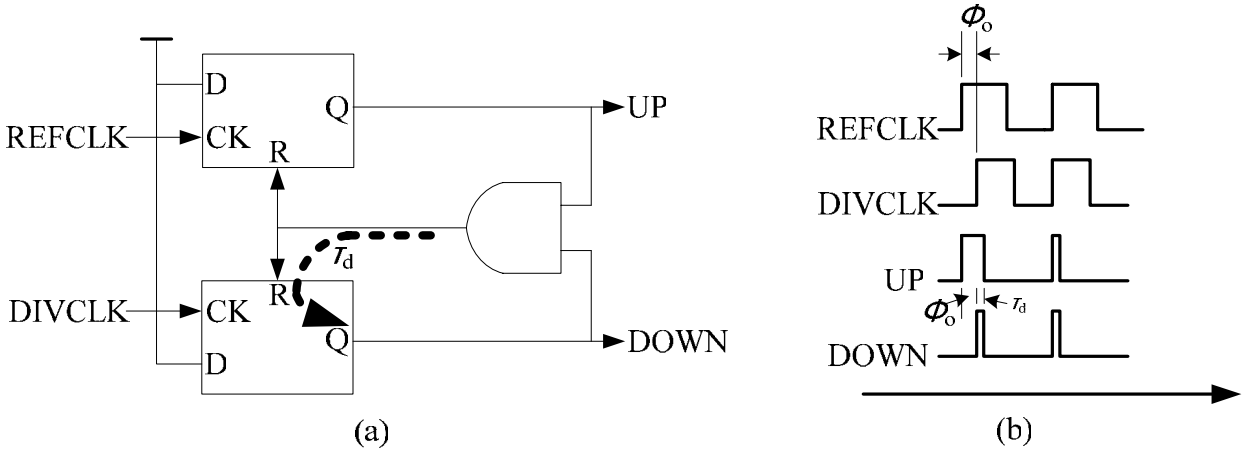


Figure 5-19(a) A sequential PFD without delayed reset. (b) PFD output waveforms.

5-3-1. PFD Dead Zone Problem.

When the phase and the frequency of the refclk and the divclk are very close, the sequential PFD can exhibit a *dead zone* problem [1]. Suppose that the two clock signals have a very small phase difference $\Delta\phi$. In that situation, the PFD can generate extremely small pulses on UP or DOWN outputs. Due to finite rise and fall times associated with the circuit design, if the narrow pulse fails to turn on the charge pump switches one way or the other, the control voltage to the VCO will not get updated. This situation is depicted in Figure 5-20. As shown in the figure, for a phase difference $|\phi| < \phi_o$, the

charge pump outputs no current. In the small signal sense, the loop gain goes to zero causing the loop to go out of lock.

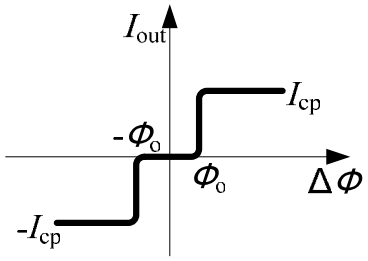


Figure 5-20. Charge pump output illustrating dead zone problem in sequential PFDs.

The dead zone is a serious problem in high performance, high frequency systems as it allows the VCO to accumulate as much phase error as ϕ_o while receiving no corrective feedback. This causes jitter in the VCO output. Although the dead zone problem is a serious one, the solution is fairly simple. Referring back to Figure 5-17, the delayed reset generator block ensures that the dead zone problem is eliminated.

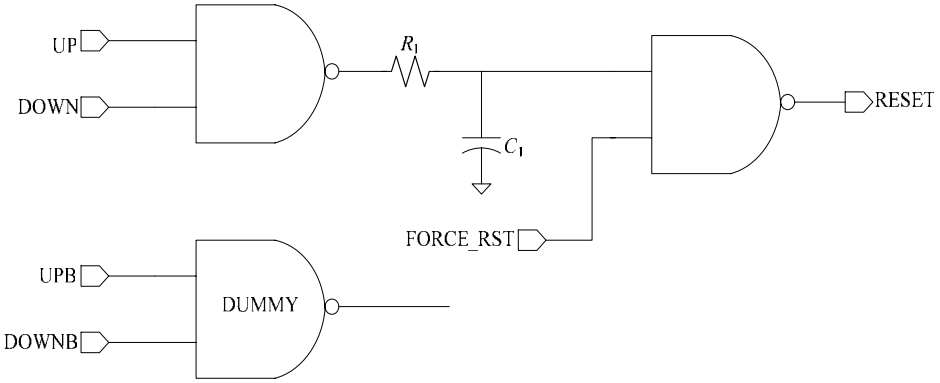


Figure 5-21. Delayed reset generator in the PFD to eliminate dead-zone problem.

Refer to Figure 5-21 which shows the schematic of the delayed reset generator. An additional NAND gate is used for the UPB and DOWNB signals in order match the capacitive loading to be the same as UP and DOWN signals. The R-C combination

introduces the necessary delay time. The *reset* input is used as an external independent signal to force a reset condition. The use of an R-C combination to establish the delay as opposed to CMOS gate based delay (for example, an inverter chain) ensures that the delay is fairly constant over supply voltage and temperature variations. Process variation of the R-C delay can be expected to be roughly the same as that experienced by a CMOS inverter. Figure 5-22 shows the timing diagram of the PFD for three consecutive cases of phase difference between the reference clock and the divider clock signals. In the first comparison, the reference clock leads the divided clock, in the second case the two signals are in phase and third case has the divider clock ahead of the reference clock. The UP and the DOWN signals shown are the outputs of the UP/DOWN signal generator that converts a CMOS level signal to a small differential signal to drive the charge pump.

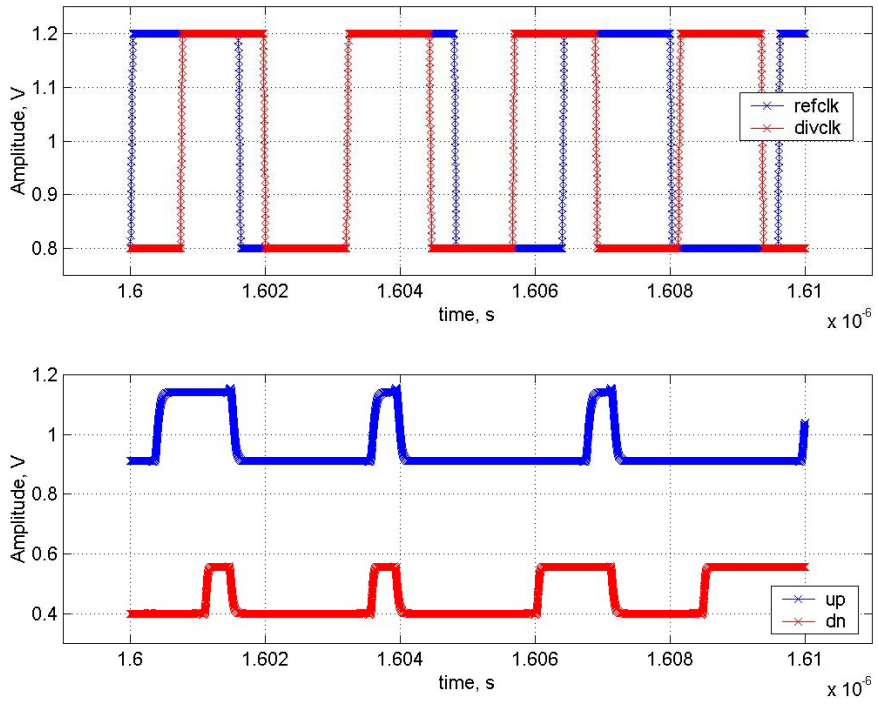


Figure 5-22. Timing diagram of the PFD.

5-4. CHARGE PUMP AND LOOP FILTER.

5-4-1. Charge Pump

The charge pump converts the phase difference information from the PFD into an equivalent current. The output current of the charge pump is converted into a control voltage for the VCO by the impedance of the loop filter. A simple conceptual model of a charge pump with the ideal output current waveforms is shown in Figure 5-23. The following observations can be made about the ideal operation of the charge pump:

a. I_{up} and I_{down} must be as close to each other as possible.

b. The turn on and turn off times of the UP and DOWN switches must match closely.

c. The output current must be as close to zero as possible when both UP and DOWN signals are on or off simultaneously.

d. The recovery time of the current sources must be negligible. It is desirable to design the charge pump such that the current sources do not have to turn off.

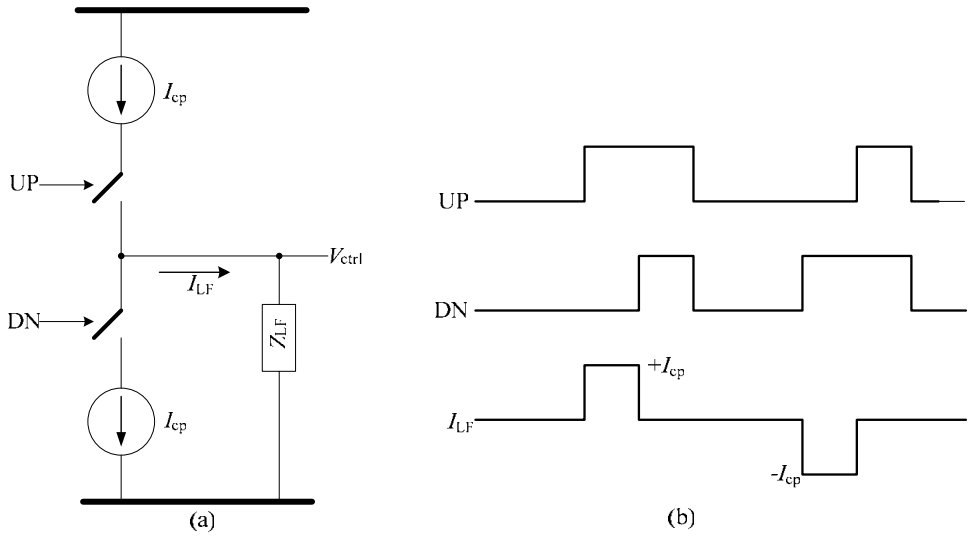


Figure 5-23. An ideal Charge pump and its output current waveforms.

Of the above observations, ‘a’ results in a *static* phase offset between the input reference and the VCO, to a first order. The current mismatch between I_{up} and I_{down} due to PVT influences will cause the static offset to vary as well. Although undesirable, this issue is less of a problem. Items ‘b’ and ‘c’, on the other hand, can have serious impact on the jitter performance of the system. Non-uniform rise and fall times of the UP and DOWN switches can cause glitches at the loop filter output causing the VCO to experience jitter [71]. Also, if there is a non-zero current going to the output when either both UP and DOWN are on or when both are off, it can cause serious jitter issues. This condition is often the most serious problem in a charge pump design because the zero output current condition usually occurs when the PLL is in lock and any leakage current can cause the VCO output to jitter. Additionally, the leakage current is highly dependent on process parameters, device sizes and temperature and can be hard to predict and accurately simulate [72]-[75]. The recovery time of the current sources can be an issue if they are allowed to turn off when the UP and DOWN switches are off. Notice that when the UP switch is off, the current I_{up} has no place to go and the current source will turn off. This issue can easily addressed by providing a parallel current path such that when the UP switch is off, another switch comes on to steer the current away into a low impedance node. This is a commonly used circuit technique which allows the current sources to stay on all the time.

The schematic of the charge pump implemented in this design is shown in Figure 5-24. This architecture was chosen for its simplicity and the high frequencies involved in this design. Transistors MP_4/MP_5 and MN_4/MN_5 generate the up and the down currents, respectively, required for the differential operation of this circuit. Transistors MP_0/MP_1 and MP_2/MP_3 form PMOS source coupled differential pairs and similarly, MN_0/MN_1 and MN_2/MN_3 form NMOS source coupled differential pairs.

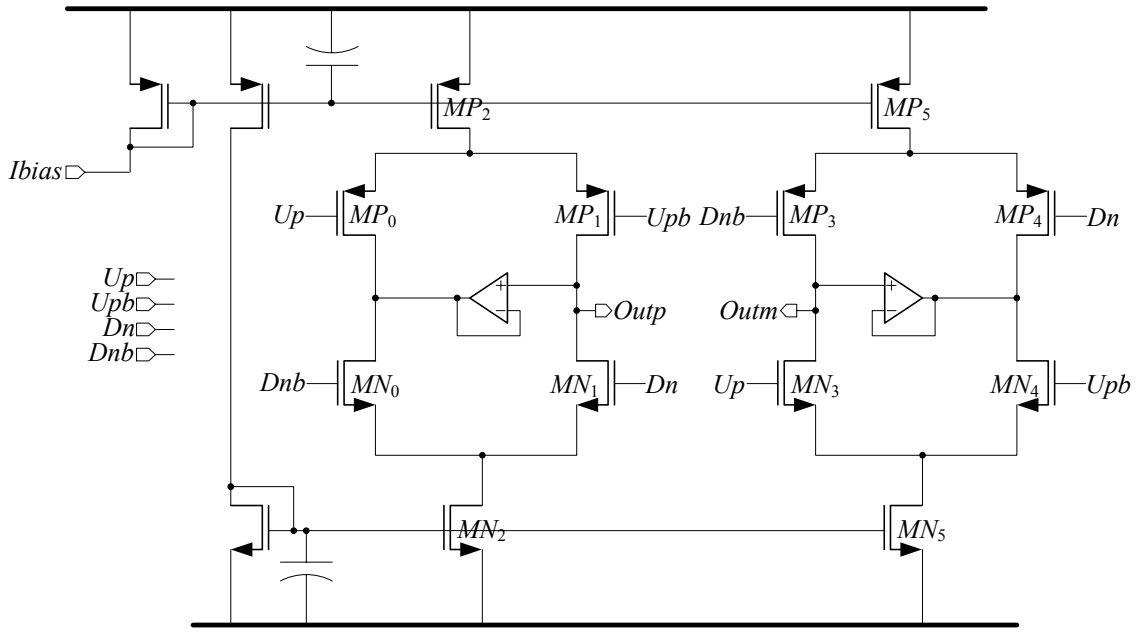


Figure 5-24. Schematic diagram of the differential charge pump.

5-4-2. Differential Current Steering.

In most charge pump designs, the transistors that steer the up and down currents to and from the output are operated as switches with full CMOS voltage levels controlling their gates [63],[66],[67]. Many other designs switch the current sources on and off [60],[70]. Both these architectures introduce severe glitches at high update rates due to parasitic capacitor coupling, transistor charge injection, finite turn on and turn off times of transistors etc. In this design, the charge pump is viewed more as a *differential amplifier* than as a current steering switch. If the PMOSFET source coupled pair of MP_0 and MP_1 are sized such that their small signal trans-conductance is able to steer the current completely from one side to the other, then full CMOS level input gate drives will

not be necessary. Also, the common source node will not experience any voltage glitches that can modulate the currents. The NMOSFETs are designed in a similar manner.

The two unknowns in the design are the gate voltage swings and the differential pair sizes. One of the two parameters can be fixed to calculate the other. Based on the $K_p^{'}$ and the $K_n^{'}$ values in this process, the PMOSFETs MP_0 and MP_1 are sized with a $W = 7\mu m$ and $L = 0.28\mu m$ for a small signal g_m of 360 $\mu A/V$. The Δv_{gs} needed to steer $60\mu A$ is calculated to be $\frac{60\mu A}{360\mu A/V} = 166mV$. On the NMOSFET side, MN_0 and MN_1 were sized with a $W = 2\mu m$ and $L = 0.28\mu m$ for a small signal g_m of $396\,\mu A/V$. The Δv_{gs} needed in this case is found to be $\frac{60\mu A}{396\,\mu A/V} = 151mV$. Minimum length of 130 nm was not used for these devices to reduce the output leakage current. When both UP and DOWN signals are off, ideally zero current should flow into the loop filter. However, any leakage current due to short channel effects will alter the loop filter voltage causing jitter. Larger than minimum device lengths are used to mitigate this problem. Another advantage of not operating the charge pump in a switch mode is the reduction in the charge injection to the loop filter. When a MOSFET is on (in its linear or saturation region), electrical charge is present in the channel (region under the gate). When the transistor is turned off, the channel charge is *squirted* to the source and the drain terminals. How the charge splits towards the source and the drain is a complicated physical phenomenon and is a function of the relative impedances seen at the source and drain terminals. By operating the charge pump more like a differential amplifier, the transistors are not completely turned off thus greatly reducing the charge injection effects. Figure 5-25 shows the schematic of the UP and DOWN signal generators that interface between the PFD and the Charge Pump to generate signals of proper voltage swings. The back connected inverters act as DC bias voltage generators to ensure proper

voltage swings on the respective voltages. The back connected inverters were chosen for simplicity in implementation. A better way to generate the UP and DOWN signals would be to use source follower based level shifters at the outputs of the CML flip flops in the PFD and resistor dividers to get the right voltage swings. This approach can be investigated for potential improvements to the design.

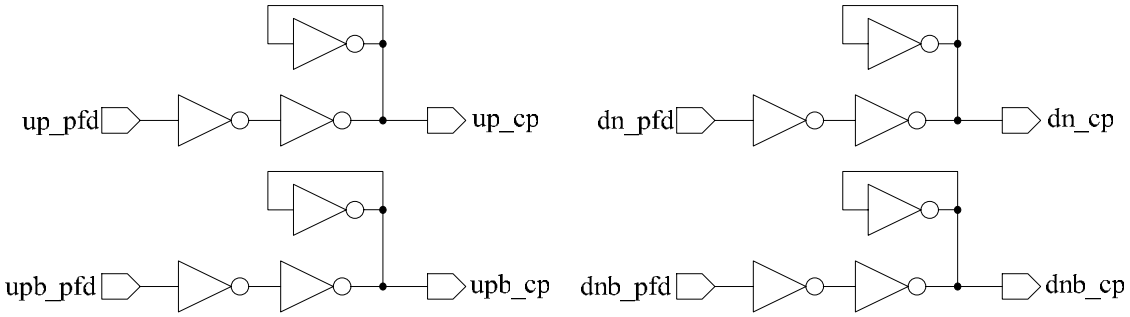


Figure 5-25. UP and DOWN signal generators for the charge pump

Figure 5-26 shows the behavior of the charge pump under three phase different conditions between the reference clock and the divider clock. The full signal path from the PFD to the charge pump is included in this simulation. Small amounts of overshoots are noticed in the output current. The most critical aspect of the charge pump is the output current when the loop is in perfect lock and when both the UP and the DOWN signals are off. The waveforms in the figure show that the charge pump behaves very well in these situations. The maximum leakage current occurs in the fast, fast, cold temperature, high power supply corner and the leakage current is observed to be 181 nA. The peak to peak jitter induced from the PFD and the charge pump can be calculated from this leakage number. The cycle to cycle voltage variation on the loop filter is influenced mostly by the smaller ripple attenuating capacitor C_2 [29]. The update rate of

the charge pump is 3.2 ns (1/312.5MHz). The differential ΔV change in the loop filter voltage due to the leakage current is

$$\Delta V_{diff} = 2 * \frac{181e-9 * 3.2e-9}{1e-12} = 1.1584mV$$

and the maximum VCO frequency variation due to this voltage is

$$\Delta F = 915e6 * 1.1584e-3 = 1.059936MHz$$

which corresponds to a maximum timing jitter of 0.108 ps.

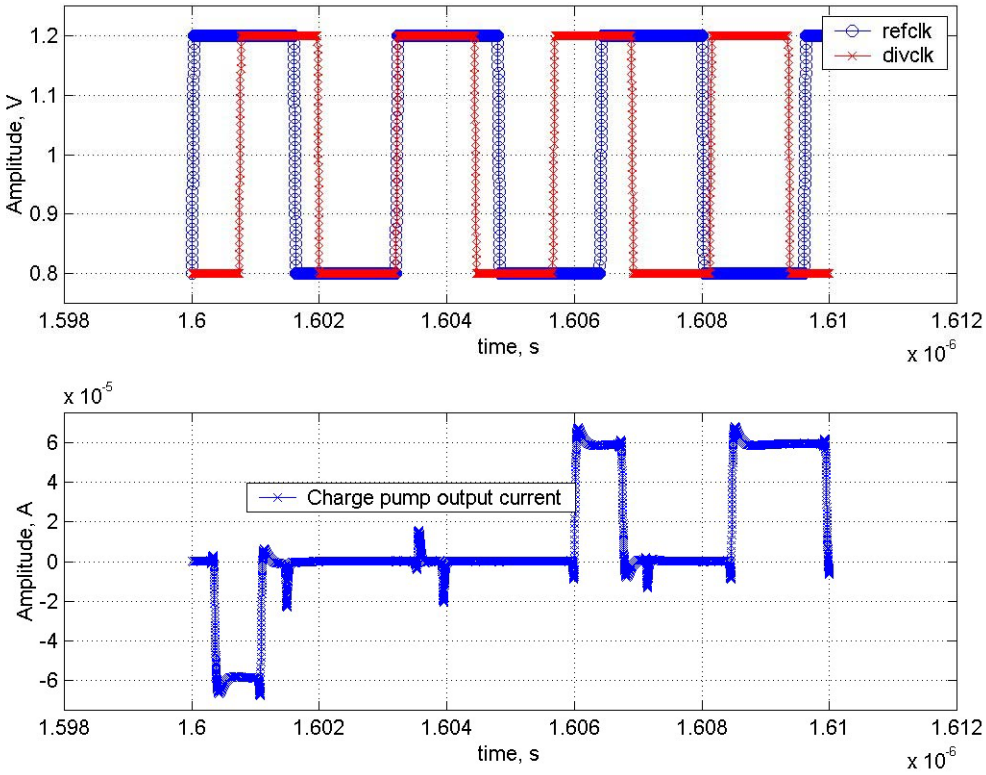


Figure 5-26. Transient behavior of the charge pump

5-4-2. Loop Filter

The schematic of the passive loop filter used in this design is shown in Figure 5-27. The transfer function of this filter was derived in Chapter 3. The maximum K_{vco} value is obtained from the Table 5-1 to be $915\,MHz/V$ and the values of the loop filter elements used are $R_1 = 3.7k\Omega$, $C_1 = 27\,pF$ and $C_2 = 1pF$. These values were obtained from the plot of the loop filter elements versus the VCO gain from Figure 4-5 in Chapter 4. The differential implementation of the loop filter requires two sets of the above resistor and capacitors. The resistor is designed in the N+ Polysilicon layer whose process parameters are given in [58]. The capacitors are implemented as metal-insulator-metal (MIM) capacitors. These structures have very little process variation and their parasitic capacitance to substrate is negligible. The total silicon area estimate for the loop filter is dominated by the capacitor C_1. The capacitance per unit area for MIM capacitors is 1.2 fF/μm^2 and the total estimated silicon area for the loop filter is 48,000 μm^2.

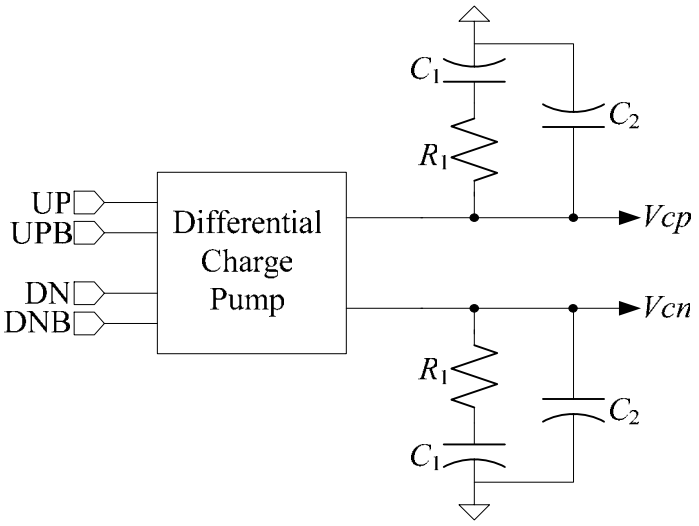


Figure 5-27. Schematic of the loop filter

5-5. DIVIDER

The most challenging aspect of the divider design is the high frequency operation required of the Flip Flops. Major issues for designing the divider are high operating frequency and low timing jitter while minimizing power consumption. All these are fairly challenging design issues [25],[76]. Several divider/prescaler designs have been reported in the literature in GaAs [77]-[79], bipolar [80], [81] and CMOS [69], [82]-[87] technologies. All these designs address the issue of doing multi-modulus division. The architecture developed in this design requires only a divide by 8 circuit which significantly reduces design complexity.

The schematic of the CML based divide by 8 circuit is shown in Figure 5-28. A CML based fast flip flop has been designed for this design. Although schematically the same as the one used in the PFD, this flip flop operates at much higher current to achieve the high frequency operation. The operating current of the fast flip flop is 3.6 mA. Four divide by 8 blocks are needed to interface with the VCO. Each circuit consists of three flip flops and if the fast flip flops are used for all the three, the total current consumed would be 12x3.6 mA = 43.3 mA which is prohibitively large. Instead, the divide by 8 is actually implemented as a divide by 4 and a divide by 2 such that the final divide by 2 does not need to have a large operating current. In the implementation the flip flop used for the final divide by 2 runs at 1 mA. This mechanism reduces the total power consumption of the divider to 31.8 mA, which is almost a 30% reduction. Further investigation in this area to reduce power in the divider is necessary for a more commercially viable solution. Table 5-2 below summarizes the simulated maximum operating frequency, power consumption and power estimates for the divider.

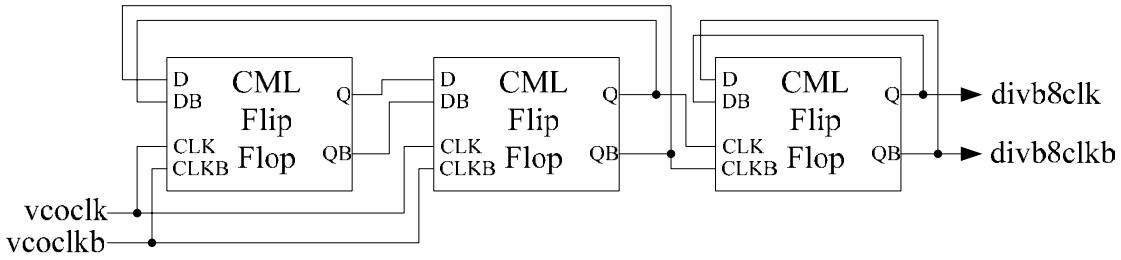


Figure 5-28. Schematic of the CML flip flop based divide by 8 circuit.

Table 5-2. Summary of simulated results for the divider and area estimates

Maximum Operating Frequency	4.4 GHz
Power dissipation	48 mW
Estimated die area	48000 μm^2

5-6. TRANSITION SELECTOR AND PHASE ACCUMULATOR

Recall from Chapter 4 that the fractional division in this architecture is achieved using phase manipulation using the transition selector and the phase accumulator. Based on the phase accumulator output, the transition selector picks the proper phases of the divider output to generate the pulse train with the necessary frequency. Note that the output frequency of the transition selector will always be equal to that of the input reference frequency, which in this case is 312.5 MHz. The loop adjusts the VCO frequency appropriately to account for the fractional divide ratio. The output frequency of the PLL, in a fractional sense, is given by $F_{vco} = F_{ref}.N(1+Ph)$.

5-6-1. Transition Selector

The schematic implementation of the transition selector is shown in the Figure 5-29. In order to simplify the description of the operation of the TSel block, a four phase

version is shown in the figure. The transition selector is comprised of a 2/4 logic decoder, four resettable D flip-flops and an OR gate.

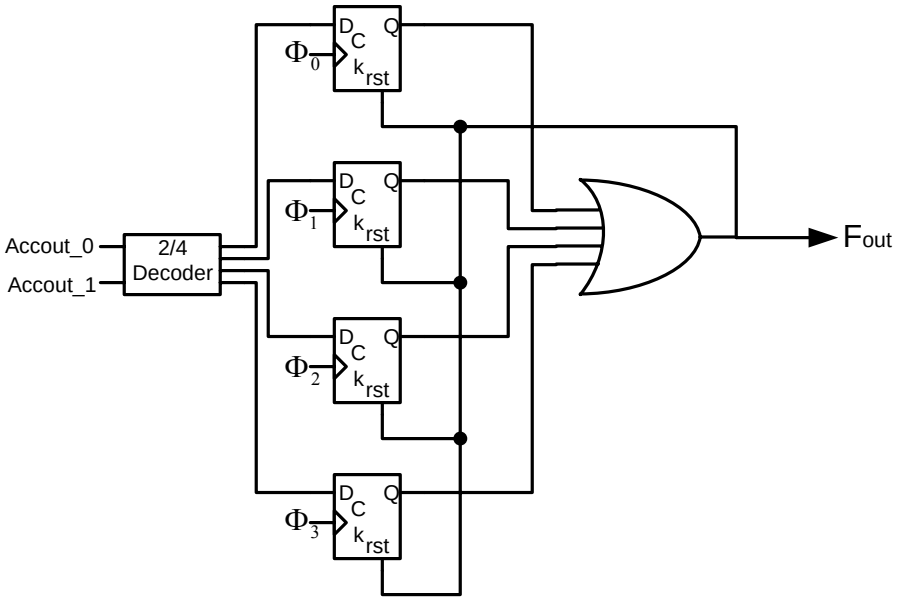


Figure 5-29. A simplified four phase version of the transition selector.

The 2/4 decoder receives the 2-bit output of the phase accumulator and decodes it into four possible values which form the inputs to the four D flip-flops as shown in the figure. At the respective rising edges of the phases $\Phi_0 - \Phi_3$, the respective outputs of the flip-flops become '1'. Every time the OR gate receives a '1', it outputs a '1' which resets all the flip flops. Thus the output of the OR gate is a pulse which will update the phase accumulator. It is very important to equalize the propagation delays from the selected clock phase to the output of the OR gate. Any variation in the delay causes variation in the reset timing and introduces jitter at the OR gate output.

The output of the divider produces 64 phases separated in time by 40 ps with a nominal frequency of 390.625 MHz (3125/8 MHz). As a result, the full implementation of the transition selector consists of 64 D flip flops. The outputs of these 64 flip flops

have to be condensed by a 64 input OR gate. Using a standard OR gate is impractical as the propagation delay through the gate is too large. Moreover, the rise and fall times of such a large input gate would be highly asymmetric and process dependent. To solve this problem, the 64 input OR gate is broken down into 8+1 NMOS only type OR gates, each having 8 inputs [56]. The schematic of the 64 input OR gate is shown in Figure 5-29. The rise and fall time of each OR gate must be closely matched to minimize timing jitter. This was achieved by extensive simulation of the OR gate over PVT corners and the final design matched the rise and fall times closely.

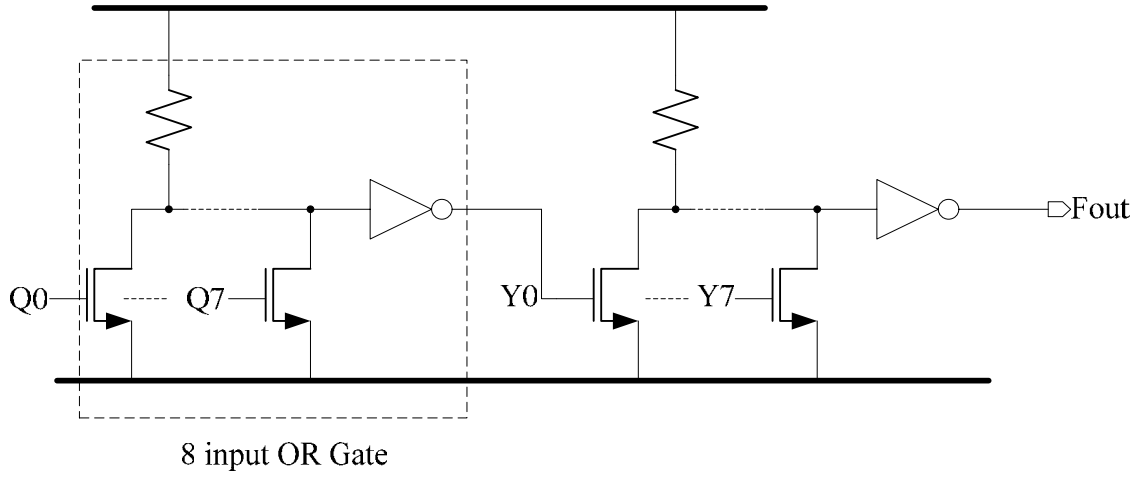


Figure 5-30. A NMOS only 64 input OR gate implemented as 8+1 topology.

5-6-2. Phase Accumulator

An m-bit accumulator consists of an m-bit adder and m-bit latches. The three targeted divide ratios were tabulated on Table 2-2 in Section 2-4. The corresponding phase accumulator input words are summarized in Table 5-3. The binary numbers required are 0.01, 0.0001, 0.00001 and a 6-bit accumulator is implemented to handle the

different frequencies. It is observed that the fixed divider provides 64 phases and a 64-bit accumulator can produce output frequencies with extremely fine resolution.

Table 5-3. Phase Accumulator input word for the expected output frequencies

VCO Output Freq	Fixed Divider Ratio	Frac. Divide Ratio	Ph. Acc Input #
3.125 GHz	8	1.25	0.25
2.65625 GHz	8	1.0625	0.0625
2.578125 GHz	8	1.03125	0.03125

A simple 6 bit Manchester adder with a conflict free bypass circuit is implemented in this design [25]. The carry of the i^{th} stage, C_i, may be expressed as

$$C_i = G_i P_i^{'} + P_i C_{i-1} \tag{5-17}$$

Where:

$G_i = A_i \bullet B_i$ and $P_i = A_i \oplus B_i$. In these equations, G_i indicates the generate signal at stage i and P_i is the propagate signal stage i of the adder. The sum output is generated by,

$$S_i = P_i \oplus C_{i-1}. \tag{5-18}$$

The 6-bit Manchester adder consists of a PG generator to generate the propagate and the generate signals, the carry generator and the sum generator, as shown in Figure 5-31.

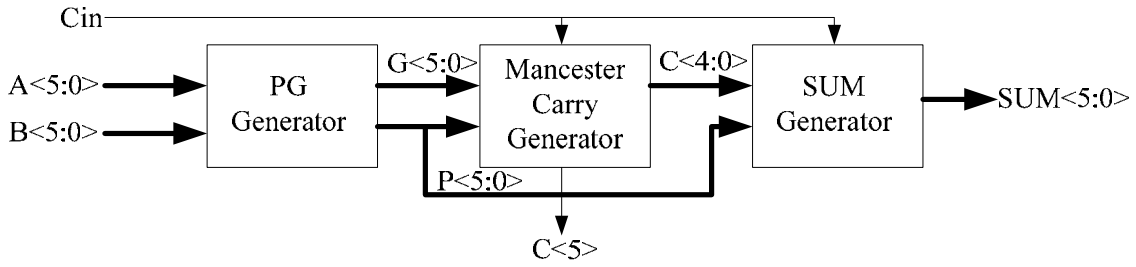


Figure 5-31. Block diagram of the 6-bit Manchester adder.

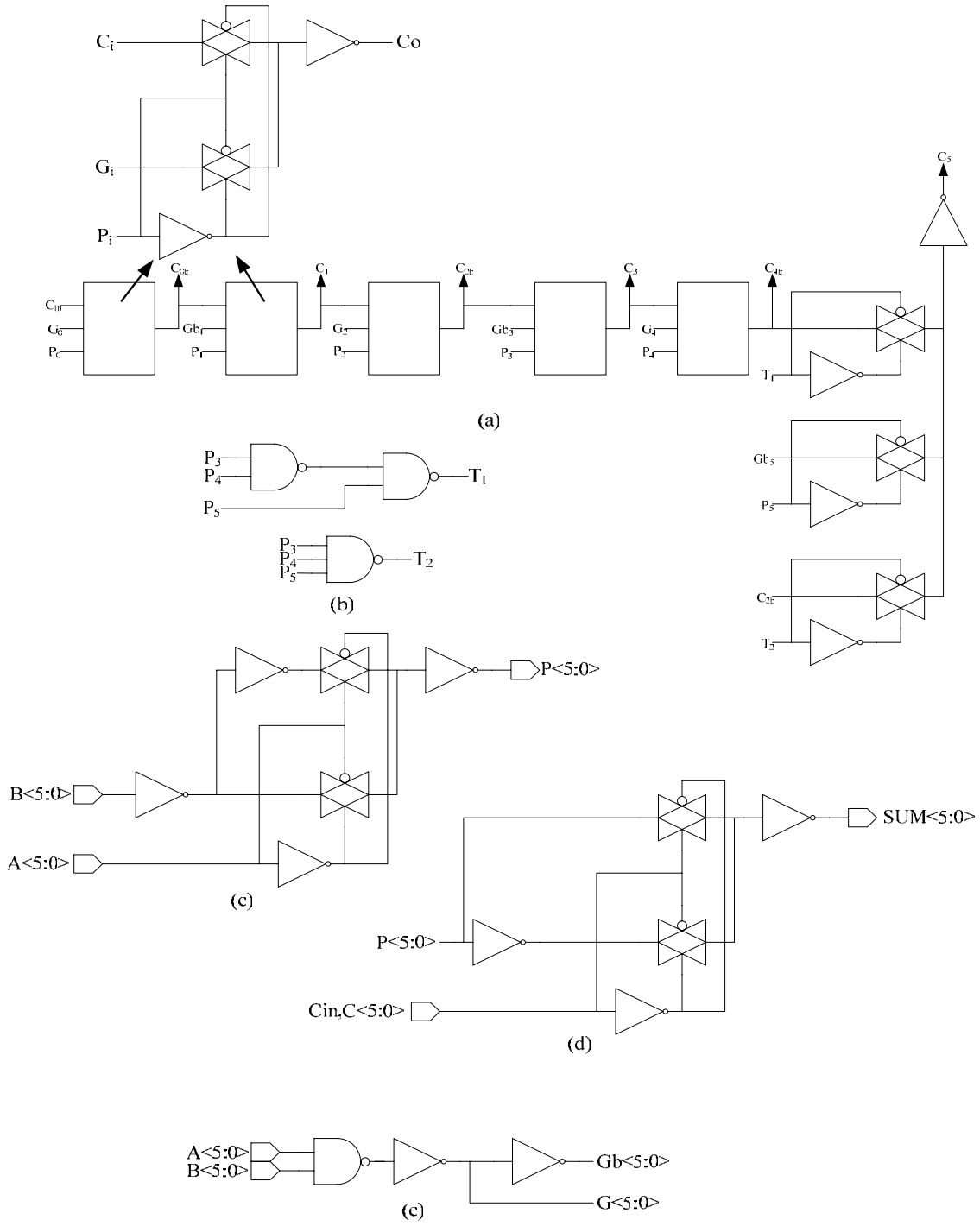


Figure 5-32. Logic implementation of the 6-bit adder: (a) Manchester carry generator, (b) Bypass Control Logic, (c) Propagate Logic, (d) Sum Logic and (e) Generate Logic.

Logic level implementation of the adder is shown in Figure 5-32. The worst case propagation time of the carry can be improved by bypassing three stages if the propagate signals P_3, P_4 and P_5 are true [25].

5-7. BIAS GENERATOR

The bias generator for this design is shown in Figure 5-33. It consists of an op-amp based V-I generator with an external resistor to generate an accurate temperature and process independent current. The input to the op-amp and the resistor are intended to be external so that the internally generated current can be accurate. A fully integrated implementation would include an on-chip bandgap or similar reference circuit with either an on-chip or an external resistor to do the V-I conversion. Designing a bandgap reference at a low supply voltage 1.5 V is a non-trivial exercise and is beyond the scope of this work. Hence, this design is implemented with an external voltage and resistor.

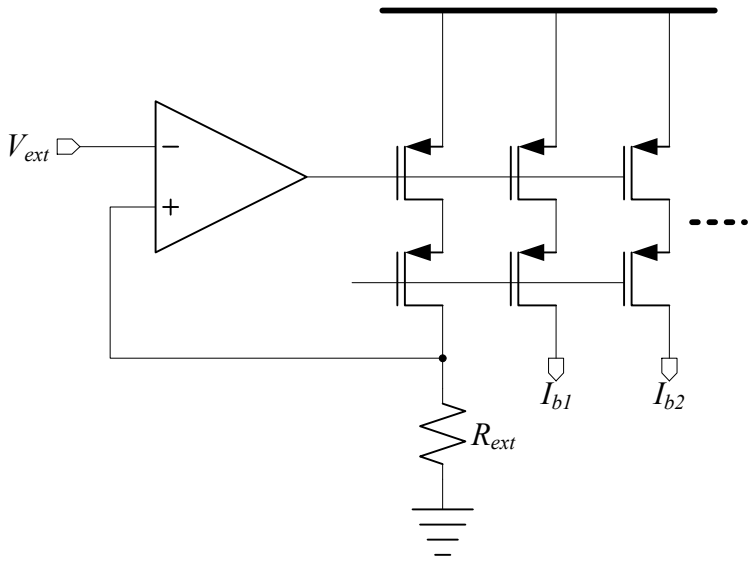


Figure 5-33. Bias Current Generator.

5-8. PUTTING IT ALL TOGETHER

Each of the building blocks of the fractional-N frequency synthesizer have been described in detail. However, the task of putting all these blocks together is a non-trivial one. This is particularly the case because of the use of CML based voltage levels in most blocks and CMOS level voltage levels in a few. The CML voltage levels in this design are set at approximately 500-650 mV peak to peak single ended. That means that for a power supply of 1.5 V with a +/-5% variation, the CML logic swing will be from 700 mV to Vdd. At the PFD to Charge Pump interface, however, the voltage swings are only about 170 mV as explained earlier.

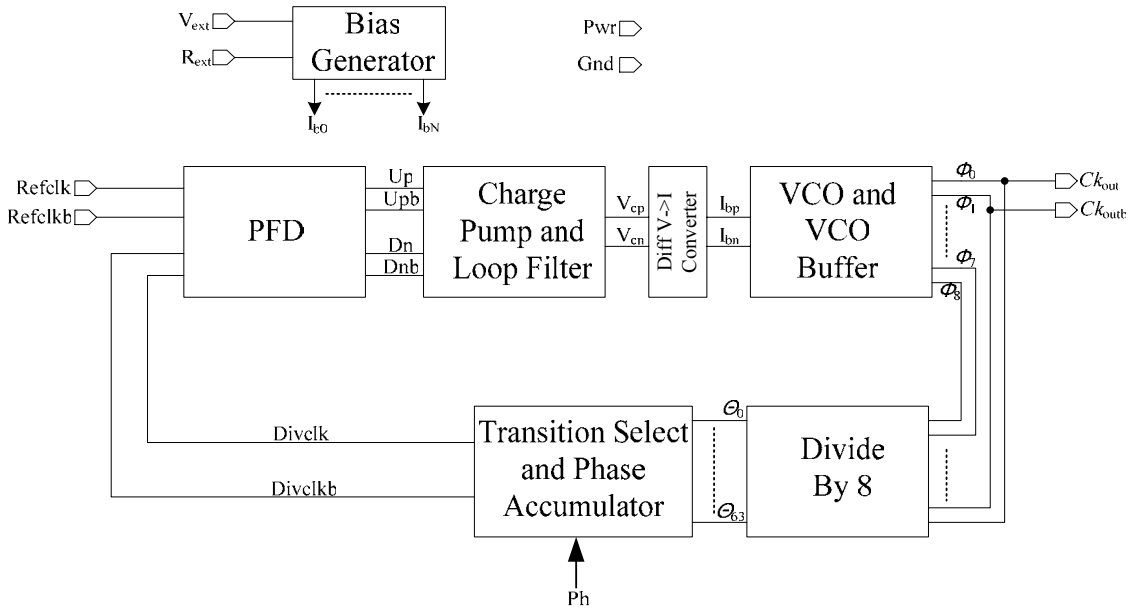


Figure 5-34. Top level schematic of the complete frequency synthesizer.

The top level schematic of the complete frequency synthesizer is shown in Figure 5-34. The performance summary of the synthesizer is given in Table 5-6. The large portion of the power supply current is consumed by the feedback divider (31.8 mA) and the transition selector (25.6 mA). Architectural and circuit level optimization may be done to address this issue. Chapter 7 discusses some of the possible changes to reduce the power consumption.

Table 5-6. Simulated performance summary of the synthesizer

Technology	0.13 μm CMOS
Power Supply	1.5 V
Max Power Consumption (@Fout = 3.125GHz)	112 mW (75 mA)
Est. Peak to Peak Jitter (@Fout=3.125 GHz)	49 ps
Est. RMS Jitter (@Fout = 3.125 GHz)	3.20 ps For a BER=10^{-14}
Est. Die Size	196,000 μm^2

The peak to peak jitter number is the sum of the jitter introduced by the VCO, the PFD and the charge pump and the feedback path of the divider and the transition selector. In the number shown in the table, the transition selector is estimated to introduce a peak to peak jitter of about 3 ps to the system. The silicon implementation may result in a slightly higher number to due to substrate noise, layout mismatches and transmission line effects due to long metal traces, etc. The RMS jitter number does not take into account the contribution due to the device noise such as thermal and 1/f noise. The major source

of device noise induced jitter is in the VCO input stages. Further investigation in this area needs to be done to arrive at a more accurate number. The overall RMS jitter number is unlikely to increase by a large amount when the device noise contribution is included.

5-9. SUMMARY

This chapter described the circuit design aspects of each of the building blocks of the fractional-N frequency synthesizer. The theory and the design of the new VCO is explained and its performance parameters are presented. Similarly, the chapter presents the design of the VCO, PFD, charge pump, loop filter, divider, the transition selector and the phase accumulator. While running at the maximum frequency of 3.125 GHz, this architecture is estimated to consume 75 mA of supply current while running off of a 1.5 V nominal power supply. This power consumption is fairly large when compared to published literature. However, when integrated in a network backplane environment, the PLL will be used to supply the clocks for multiple ports (typically four) and the power measurement metric is calculated on a per port basis and under that scenario, the power consumption of the PLL will be 18.5 mA/port.

When compared to stand-alone PLL designs, this architecture definitely consumes more power. Most of the power is consumed in the feedback divider and the transition selector and this is not a huge surprise.

Chapter 6: Layout Design

This chapter briefly discusses the layout issues concerning the synthesizer circuit developed so far. Although the full layout and fabrication of this design has not been done, a discussion of the various issues involved in the layout of such a high frequency, high performance system is relevant. A possible floor plan of the synthesizer is shown in Figure 6-1. The VCO and the VCO buffers are placed close together and separated from the rest of the circuitry with guard rings for noise immunity. The VCO is also placed close to the pads of the package such that the power supply and ground routing is minimized. The VCO and the VCO buffers are placed as close to each other as possible to avoid excessive routing parasitics on the VCO outputs

The loop filter, charge pump and the PFD can be laid out together as shown in Figure 6-1. These three blocks operate at the rate of the input reference clock and must be shielded from the VCO. The most critical signal nodes in the system are the control voltage to the VCO, that is the loop filter outputs. Any noise coupled to these nodes influence the jitter performance of the VCO. Placing the loop filter close to the VCO allows these signals to be routed in the quietest possible manner and also in the shortest possible distance to the VCO.

The divider, the transition selector and the phase accumulator can be placed to the side as shown in the figure. Again, this type of floor plan allows the output of the transition selector to be routed to the input of the PFD very conveniently. For issues related to the pad parasitics, instead of bringing out the high frequency clock, a divided version of that can be brought outside of the chip.

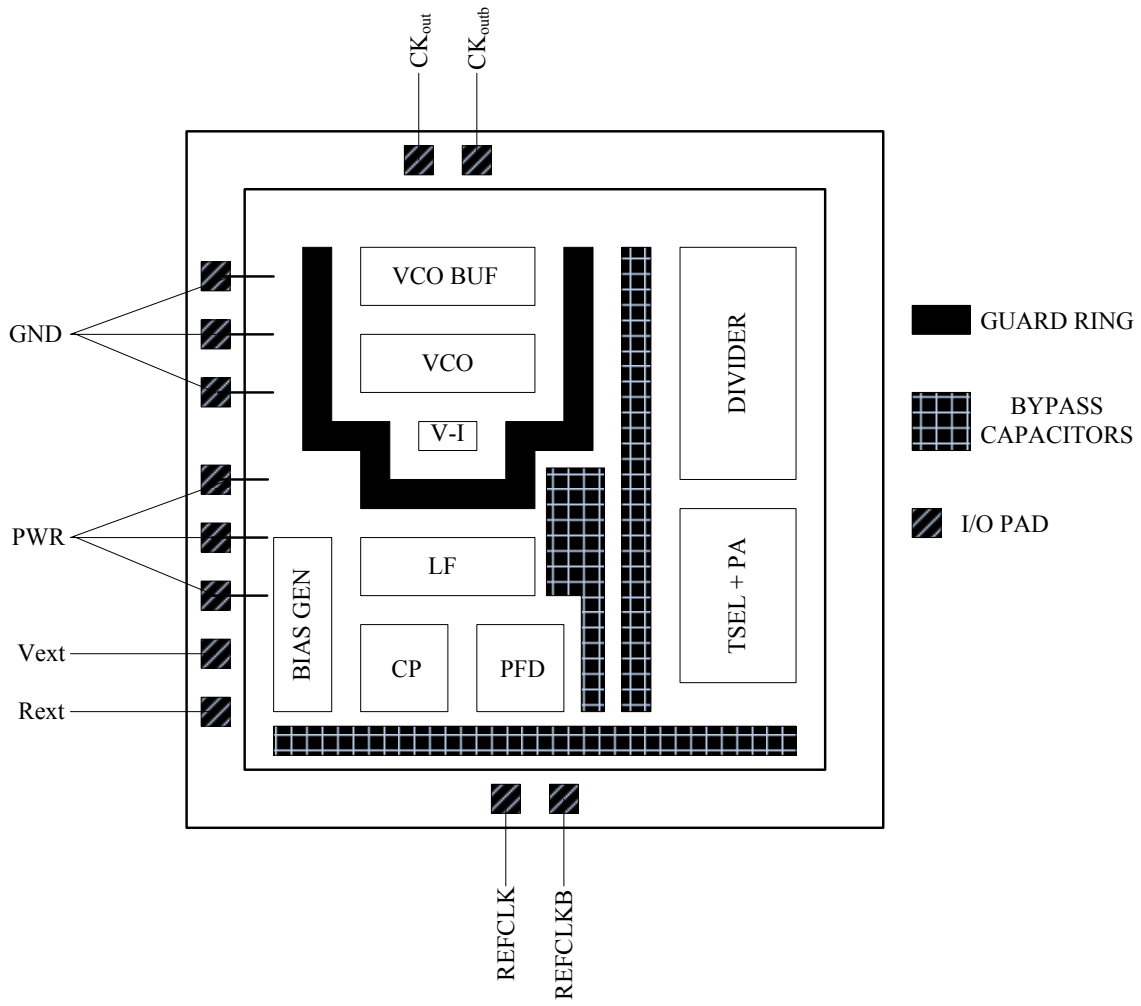


Figure 6-1. Possible layout floor plan for the frequency synthesizer.

If possible, the VCO and the VCO buffer should be operated from a separate pair of power supply and ground pins. This allows for the maximum noise immunity for the VCO. However, in a real world implementation, if separate pins are not available, then either multiple power supply and ground pads must be placed inside the chip which are bonded to a single pins outside the package or an on-chip voltage regulator can be included to generate a voltage dedicated for the VCO. An on-chip voltage regulator may

also be more desirable so that the VCO can be operated at a supply voltage different than that of the rest of the circuitry.

6-1. VCO LAYOUT

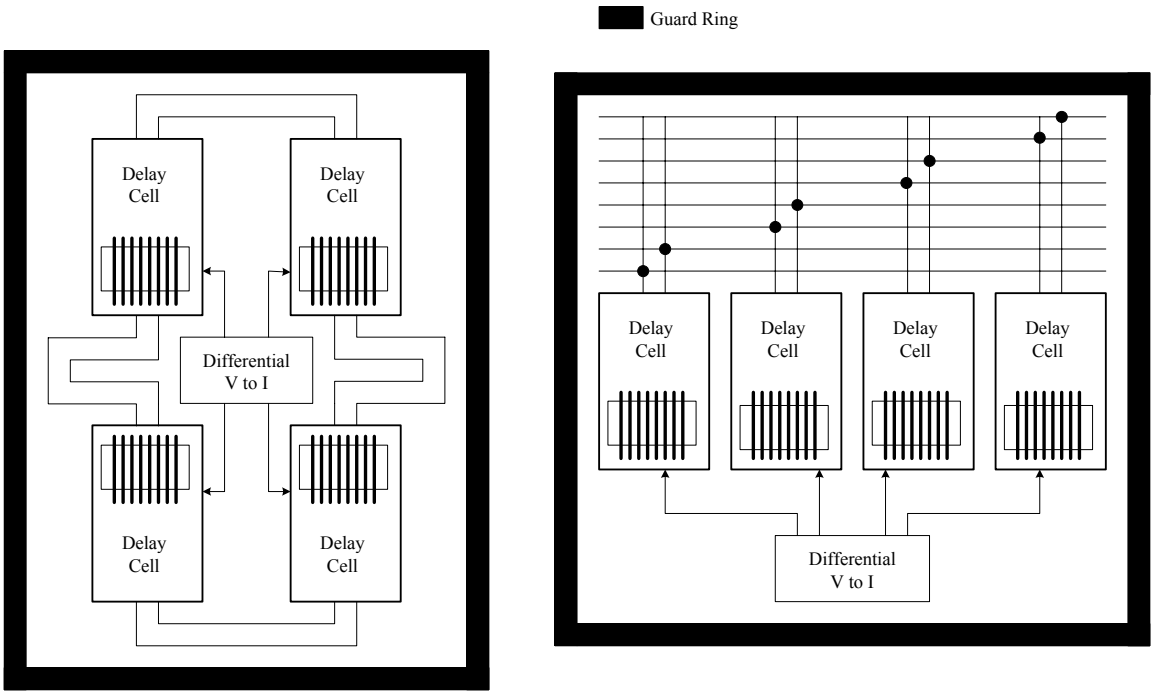


Figure 6-2. Candidate floor plans for the VCO showing the delay cell placements.

The layout of the VCO and the delay cell is very crucial to the overall jitter performance of the VCO as well as the PLL. Extreme care is taken to ensure that differential signals are matched well in all respects. Two candidate delay cell placements for the VCO are shown in Figure 6-2. In the first floor plan, the four delay cells are placed in a common centroid symmetry enclosed inside a guard ring. The V to I converter is placed in the center such that the routing to the individual delay cells is identical. One drawback of this plan is that the routing between the delay cells will not be perfectly

matched. The floor plan on the right addresses this problem where the delay cells are placed next to each other. This plan also has the advantage that the loop filter outputs can be brought to the voltage to current converter without crossing any of the VCO signals. Just from that point of view, the second floor plan may be preferred. The active and the passive elements inside the delay cell must also be laid out with common centroid symmetry and with dummy cells such that process and planarization variations introduce very minimal performance degradations.

The above two aspects are just examples of good layout and floor planning practices that are needed to ensure the functionality and the performance of the design. Overall performance will be decided by the layout of the rest of the functional blocks. Good layout practices, solid power supply and ground routing, minimization of parasitics and in cases where longer routing is needed, matching of the parasitics, keeping clocks of different frequencies far from each other, simulating the circuits with the electrical model of the pads etc can significantly increase the probability of getting a fully functional and fully performing silicon.

6-3. SUMMARY

Although the layout of the synthesizer has not been completed, this chapter describes the layout strategies to be used that ensure a successful silicon implementation. As examples, candidate floor plans of the synthesizer and the VCO are shown. In high performance and high frequency mixed mode systems, the block level and the top level layout of the system is as critical to the success of the design as any other aspect.

Chapter 7: Conclusions

This dissertation presents the design and development of a new architecture for a fractional-N frequency synthesizer for a high speed serial communications link such as a network backplane. As part of the development, an adaptive line rate network backplane architecture is also proposed. The new network backplane architecture allows for on the fly change of the line coding schemes used at either ends of the serial link. The choice of the line codes dictates the bandwidth overhead needed to achieve a particular data rate. The smaller the overhead, the slower the serial link needs to operate to achieve the data rate. A slower operating serial link is more robust, experiences less channel influenced signal distortion and reduces the probability of making a bit error at the receiver (lowers bit error rate).

The adaptive nature of network backplanes requires that the serial clock generator at the transmitter be flexible enough to generate any one of several frequencies. Such a problem can be solved by changing the reference clock source to the transmit clock synthesizer, having a set of clock synthesizers one of which is operational at any time, having a single PLL with multiple VCOs and several other options. These approaches to solve the clock generation problem were studied along with their relative merits and demerits.

A new fractional N frequency synthesizer architecture is developed and designed to address this problem. This architecture recognizes the fact that in a sequential phase frequency detector, the two inputs need not have a 50% duty cycle. The PFD is active on the rising edges of the two signals. Thus if a phase manipulation technique is embedded in the feedback loop of the PLL such that a pulse train with right frequency is generated, then the loop dynamics will ensure that the VCO frequency locks to a value which can be

a fractional multiple of the input reference. In this architecture, a *direct digital period synthesis* scheme is embedded along with a fixed modulus divider in the feedback of the PLL. With the help of simple digital circuitry and a phase accumulator, the pulse train is generated to be compared with the input reference. The frequency of the VCO output is related to that of the input reference through the relation, $F_{out} = F_{ref} \bullet N(1 + Ph)$, where N is the ratio of the fixed divider and Ph is the input control word to the phase accumulator. It was demonstrated that a simple 6-bit accumulator is sufficient to generate the three target frequencies addressed in this work.

The output jitter performance of this architecture is governed primarily by the VCO. This is because the design does not place low bandwidth requirements on the loop, as opposed to other fractional-N architectures. For example, in a fractional-N synthesizer using a multi-modulus divider, the pattern in which the divide ratio is modulated appears on the VCO as *spurs* in the frequency domain around the fundamental or pattern jitter in the time domain. In circuits using ΔΣ modulators, the loop bandwidth must be kept low in order to filter out the shaped quantization noise of the ΔΣ.

The transition selector in the feedback relies on close spacing of the clock phases to be able to jump from one to the other. The time spacing between phases essentially sets a lower bound for the peak to peak jitter of the system. In this case, a four stage ring oscillator was implemented to operate at 3.125 GHz with a phase spacing of 40 ps. Thus the peak to peak jitter estimated from simulation outputs was as much as 55 ps. This number is very close to the maximum allowable jitter in a network backplane system running at a line rate of 6.25 GBPS.

A fully differential implementation of the PFD/Charge Pump/Loop Filter/VCO signal path ensures good power supply and substrate noise immunity. Unlike most designs, the control path to the VCO is also kept fully differential which helps to double

the dynamic range of the loop filter/VCO combination. The single ended voltage needs to swing only half as much to get the same frequency range out of the VCO. In low voltage designs, this feature can be very attractive. Also, the charge pump, instead of being designed as a *semi-digital* circuit is now operated as a fully analog circuit. A differential amplifier based structure is used to steer the UP and DOWN current, which greatly reduces switching glitches keeping the current sources as quiet as possible.

7-1. PERFORMANCE COMPARISON

Table 7-1 shows a performance comparison of several published works with this design. A MATLAB program was used to convert the phase noise spectrum published in each of the references to an equivalent RMS jitter number and in turn, a peak to peak jitter value at the same BER. This design performs better than two other designs using ring oscillators [23],[46] and compares closely with that using an LC oscillator [66].

Table 7-1. Performance comparison of this work

Designs	Fout GHz	Process μm	VCO Type	Area sq. mm.	Pdiss, Vsup mW, V	Peak to peak jitter, ps
This Work*	3.125	0.13	Ring	0.2	112,1.5	49
[23]*	1.8	0.6	Ring	10.7	52, 3.3	164
[46]*	2.4	0.35	Ring	3.7	50,3.3	76.5
[52]	0.6	0.25	DLL	0.13	60, 2.5	54
[66]	6.3-9	0.18	LC	0.77	58, 1.8	42.6
[67]	2.4	0.12	LC	0.7	32, 1.2	11.3

* Fractional N Synthesizers.

The peak to peak jitter of this design can be significantly reduced by decreasing the time separation between adjacent phases. For example, by doubling the number of phases from 8 to 16, the time step between each phase at 3.125 GHz will be 20 ps and the overall jitter would reduce to 29 ps. The important observation is that the contribution of the circuit induced jitter to the overall performance is small. Doubling the number of phases may be done by going to an 8 stage ring oscillator. The drawback of doing so would be increased device noise contribution. Another method to increase the number of phases is to perform one stage of phase interpolation. Such a scheme would be sensitive to process variations and device mismatches. Careful design, layout and trimming options will be needed for any delay/phase mismatch calibration.

Addressing the power consumption of this design, there are both architectural and circuit design aspects that have contributed to the high power consumption:

a. The two blocks that consume the most power are the divider and the transition selector. The large power consumption is due to the large number of flip flops in these blocks. The fast flip flops used in the divider were not optimized for power consumption during the design phase. Additional power savings can be obtained by thoroughly optimizing the flip flop design. The same argument can be made for the flip flops used in the transition selector. Quite possibly, a full CMOS implementation of the flip flop can be used in the transition selector and the current reduced. However, a fully CMOS version would introduce a lot of power supply and ground noise which is undesirable.
b. If the transition selector and the fixed divider are interchanged, then only eight fast flip flops are needed for the transition selector as it operates on only the VCO output. Instead of four dividers, only one divider is needed which greatly reduces power consumption. The drawback, however, is that the

frequency resolution will be reduced by a factor of 8 as the transition selector now has only 8 phases to manipulate as opposed to 64. For limited number of output frequencies, this set up will be more power efficient.

c. Use of CML based design in most of the blocks also increases power consumption. However, in a power consumption versus noise injection tradeoff and in most cases, the lower noise performance is preferred.

7-2. FUTURE WORK.

The most critical validation of any new architecture and design is in the form of a working silicon. Hence, a silicon implementation of this work would be of great interest. Opportunities to commit this design to silicon implementation will be explored. The relatively large power consumption of this design is another area that needs investigation. Both architectural and circuit level alternatives were proposed to address this issue in the previous section. Each of these can be investigated with respect to the overall performance of the system.

This research work attempts to address a very real problem existing in the industry even today. Personal communications with large networking companies have shaped the scope of the problem. The applicability of the adaptive line rate network backplane in a networking company's real world product installation would be a nice validation of this concept. Any possibility in this direction will be explored.

Recent product announcements and research papers have described serial communications devices at 6.25 GBPS and beyond. These, however, achieve the performance with the help of faster silicon processes rather than attempting to solve the speed problem at an architectural level. As bandwidth and throughput requirements rise with newer applications involving video information, the line rate bottleneck will again

be faced. Constant chasing of smaller and faster processes will not be able to solve the problem as operating voltages reduce. This research has addressed the problem at an architectural level.

Appendices

A-1. HSPICE STIMULUS FILES

A-1-1. VCO simulation

```
* simulation stimulus file to run the VCO

.options method=gear brief captab nomod probe measdgt=7 ingold=2

.include 'hspiceFinal'
.param cnt = 0
.param bias=100uA
.param cbias=200uA
.param vco_sup='1.5*1.0'
.temp=-40 50 125

* Model file inclusion
.lib 'c:\wrk\spice\models\include.txt' TYPICAL

* power supply connections with possible metal parasitics
rsup1 t1 vco_pwr 1
vsup1 t1 0 pwl 0ns 0 10ns vco_sup
rgnd vco_gnd tgnd 1
vgnd tgnd 0 0

* Bias currents for the VCO and the V to I
ib0 ibias vco_gnd pwl 0ns 0 10ns bias
ib1 vco_pwr ib_const pwl 0ns 0 10ns cbias

* Control voltage definition
vc1 vcn vcom pwl 0ns 0 10ns cnt
vc2 vcom vcp pwl 0ns 0 10ns cnt
vc3 vcom 0 pwl 0ns 0 10ns 'vco_sup/2'

* simulation cards
* First run a single simulation
*.tran 1ps 40ns
* Run a temperature sweep
*.tran 1ps 40ns sweep temp poi 3 -45 50 125
* Run a control voltage sweep
.tran 1ps 40ns sweep cnt lin 20 -0.4 0.4
* Run a bias current sweep
*.tran 1ps 60ns sweep cbias lin 10 50u 500u
*.tran 1ps 60ns sweep bias lin 10 50u 500u

* Output plotting and probing
*.probe tran v(o0n) v(o0nb) v(p1) v(p1b) v(p2) v(p2b) v(p3) v(p3b)
```

```
*.probe tran v(gnda) i(rsup1) i(vsub1)
*.probe tran i(xi4.mn6) i(xi4.mn7) i(xi0.mn6) i(xi0.mn11) i(xi0.mn10)
*+ i(xi0.mn12) i(xi0.r0) i(xi0.r3)
*.probe tran v(v0p) v(v0m) v(v45p) v(v45m) v(v90p) v(90m) v(v135p)
v(v135m)
*.probe tran v(v0m) v(v0p)
*.probe tran v(xi4.net29) v(xi0.net15) v(xi0.net039) v(vcp) v(vcn)
*.probe tran v(ibn) v(ibp) v(ib_const)

* Some parameter measurements
* First measure the output common mode voltage
.measure tran vdiff param 'v(vcp)-v(vcn)'
.measure tran vmax max v(v0p) from=34ns to=35ns
.measure tran vmin min v(v0p) from=34ns to=35ns
.measure tran vavg param '(vmax+vmin)/2'
* Next measure the VCO frequency
.measure tran ft1 when v(v0p)='vavg' td=34ns rise=1
.measure tran ft2 when v(v0p)='vavg' td=34ns rise=2
.measure tran fosc param '1/(ft2-ft1)'
* Now measure the output peak to peak amplitude
.measure tran phpp pp v(v0m) from=34ns to=35ns
.measure tran phbpp pp v(v0p) from=34ns to=35ns

* Process, variation cards.
.alter
.prot
.lib 'c:\wrk\spice\models\include.txt' FAST
.unprot
.alter
.prot
.lib 'c:\wrk\spice\models\include.txt' SLOW
.unprot
*.alter
*.prot
*.lib 'c:\wrk\spice\models\include.txt' FSSL
*.temp -10
*.unprot
*.alter
*.prot
*.lib 'c:\wrk\spice\models\include.txt' SLFS
*.temp 120
*.unprot

.end
```

A1-1-2. PFD and Charge Pump simulation

```
* simulation stimulus file to run the PFD and the charge pump

* Spice option decks
.options method=gear brief captab nomod post
.options off ingold=2 probe numdgt=7
```

```
* Include the netlist
.include 'hspiceFinal'

* Some temperature cards
.temp -40 50 125

* Parameter definitions
.param cbias=60uA
.param pbias=900uA
.param cp_sup='1.5*1.0'
.param pd_sup='1.5*1.0'
.param filt=0.75

*up and down signal timings
.param fcp=312.5Meg
.param refper='1/fcp'
.param tr='refper*0.01'
.param tf='refper*0.01'
.param refpul='0.5*(refper-tf-tr)'
.param td='500/fcp'

* Run a worst case 30% faster divide clock
.param fdiv='fcp*1.3'
.param divper='1/fdiv'
.param divpul='0.5*(divper-tr-tf)'

*up and down signal levels
.param com=1.0
.param amp=0.2

* Model cards
.lib 'c:\wrk\spice\models\include.txt' TYPICAL

* power supply
rsup1 t1 cppwr 0.1
vsup1 t1 0 pwl 0ns 0 'td/20' cp_sup
rsup2 t2 pdpwr 0.1
vsup2 t2 0 pwl 0ns 0 'td/20' pd_sup
rgnd gnda tgnd 0.1
vgnd tgnd 0 0

* Bias currents to the charge pump and the PFD
iv0 cpbias gnda pwl 0ns 0 'td/20' cbias
iv1 pdpwr pdbias pwl 0ns 0 'td/20' pbias

vout cppwr cpout pwl 0ns 0 'td/20' filt
vfr  force_rst 0 pwl 0ns 0 'td/20' 0 '(td/20)+10ps' pd_sup
+ 'td-10ns' pd_sup 'td-10ns+10ps' 0

* clock inputs
vrf refclk 0 pulse 'com-amp' 'com+amp' td tr tf refpul refper
vrfb refclkb 0 pulse 'com+amp' 'com-amp' td tr tf refpul refper
```

```
vdv divclk 0 pulse 'com-amp' 'com+amp' 'td+(refper-divper)' tr tf
+divpul divper
vdvb divclkb 0 pulse 'com+amp' 'com-amp' 'td+(refper-divper)' tr tf
+divpul divper

* simulation and option cards
*.tran 1ps 100ns sweep filt poi 6 1.0 1.25 1.5 1.75 2.0 2.25
*.tran 1ns 1.6us 1ps 1.61us sweep filt lin 11 0.4 1.1
.tran 1ns 1.6us 1ps 1.61us

* Output plotting and probing
*.probe tran i(xi3.m23) i(xi3.m21) i(vout) i(xi3.m22) i(xi3.m20)
*+ i(xi3.m25) i(xi3.m24) i(xi3.m18)
.probe v(refclk) v(divclk) v(dn) v(up)
.print v(refclk) v(divclk) v(dn) v(up)
*.probe i(vout)
*.print i(vout)

*.alter
*.prot
*.lib 'c:\wrk\spice\models\include.txt' FAST
*.unprot
*.alter
*.prot
*.lib 'c:\wrk\spice\models\include.txt' SLOW
*.unprot
*.alter
*.prot
*.lib 'c:\wrk\spice\models\include.txt' SLFS
*.unprot
*.alter
*.prot
*.lib 'c:\wrk\spice\models\include.txt' FSSL
*.unprot

.end
```

A-2. MATLAB CODE

```
% Program to compute the PLL open loop and closed loop responses given
% the bandwidth and phase margin and the Kvco from the VCO design

clear all;
close all;

%constants
two_pi = 2.0*pi;
fref = 312.5e6;
```

```
%set desired corner frequency and phase margin
fc=fref/20;
pm=75;

%nominal divide ratio
Ndiv = 10;

%pll parameters
wc=two_pi*fc;
rm = pm*pi/180; %convert phase margin to radians
fstart=1000;fstop=1e10;nstep=100;

fratio = (fstop/fstart)^(1/nstep); % frequency step ratio
for i = 1:nstep+1
      f(i) = fstart*fratio^(i-1);
end % geometric progression of frequencies for log plot
w=2*pi*f; % frequency array in Hz converted to radian frequency

tau_z = 1/((sec(rm)-tan(rm))*wc);
tau_p = 1/(wc*wc*tau_z);
kv=915e6; %Maximum kvco number obtained from VCO design

kvco=two_pi*kv;
icp=60e-6; %Design trade off

R = ((Ndiv*wc)/(icp*kvco))*(tau_z/(tau_z-tau_p));
C1 = tau_z/R;
C2 = C1*tau_p/(tau_z-tau_p);

s=tf('s');
t_z=R*C1;t_p=R*C1*C2/(C1+C2);
zero=1/t_z;
pole=1/t_p;
funity=(icp*kvco*R)/Ndiv;
g_s = ((icp*kvco)/Ndiv)*(1/(C1+C2))*(1+s*t_z)/(s^2*(1+s*t_p));
k_c = ((icp*kvco)/Ndiv)*(1/(C1+C2));
h_s = k_c*(1+s*t_z)/(s^3*t_p+ s^2+s*t_z*k_c+k_c);
[mg,pg]=bode(g_s,w);mg_1=reshape(mg,1,nstep+1);pg_1=reshape(pg,1,nstep+
1);
[mh,ph]=bode(h_s,w);mh_1=reshape(mh,1,nstep+1);ph_1=reshape(ph,1,nstep+
1);
subplot(2,1,1);semilogx((w/(2*pi)),20*log10(mg_1),'bx-');hold on;
semilogx((w/(2*pi)),20*log10(mh_1),'rx-');grid;xlim([1000 1e10]);
xlabel('Frequency, Hz'); ylabel('Magnitude, dB');
subplot(2,1,2);semilogx((w/(2*pi)),pg_1,'bx-');hold on;grid;
semilogx((w/(2*pi)),ph_1,'rx-');xlim([1000 1e10]);
xlabel('Frequency, Hz'); ylabel('Phase, deb');
```

```
% Program to compute the PLL paramters given the bandwidth and
% phase margin

clear all;
```

```
close all;

%constants
two_pi = 2.0*pi;
fref = 312.5e6;

%set desired corner frequency and phase margin
fc=fref/20;
pm=70;

%nominal divide ratio
Ndiv = 10;

%pll parameters
%kv=625e6; %625 MHz/V. desired kvco
wc=two_pi*fc;
rm = pm*pi/180; %convert phase margin to radians

tau_z = 1/((sec(rm)-tan(rm))*wc);
tau_p = 1/(wc*wc*tau_z);

kv=[500:10:1200];
for i=1:length(kv)
      kv_rad = two_pi.*kv.*1e6;
   Ip = 60e-6; % charge pump current

   R1 = ((Ndiv.*wc)./(Ip.*kv_rad)).*(tau_z/(tau_z-tau_p));
   C1 = tau_z./R1;
   C2 = C1.*(tau_p/(tau_z-tau_p));
end
subplot(3,1,1); plot(kv.*1e6,C1.*1e12,'bx-');
xlabel('vco gain,Hz/V'); ylabel('loop filter cap C1, pF');
grid; subplot(3,1,2);plot(kv.*1e6,C2.*1e12,'bx-');grid;
xlabel('vco gain,Hz/V'); ylabel('loop filter cap C2, pF');
subplot(3,1,3); plot(kv.*1e6,R1,'bx-');grid;
xlabel('vco gain,Hz/V'); ylabel('loop filter resistor');

fid = fopen('lpf.txt','w');
for i=1:length(kv)
   fprintf(fid,'%6.4g\t%6.4g\t%6.4g\t%6.4g\n',kv(i),C1(i),C2(i),R1(i));
end
fclose(fid);
```

Bibliography

1. B. Razavi, *Design of Analog CMOS Integrated Circuits,* New York: McGraw Hill Publications, 2001.

2. C. K. K. Yang, *Design of High Speed Serial Links in CMOS*, Ph.D. Dissertation, Stanford University, 1998.

3. V. Balan, J.Caroselli, J.Chern, C.Chow, R.Dadi, C.Desai, L.Fang, D.Hsu, P.Joshi, H.Kimura, C.Liu, T.Pan, R.Park, C. You, Y.Zeng, E.Zhang and F.Zhong, "A 4.8-6.4 GBPS Serial Link for Backplane Applications Using Decision Feedback Equalization," *IEEE Journal of Solid State Circuits,* 2005, vol. 40, pp. 1957-1967.

4. S. B. Anand and B. Razavi, ""A CMOS Clock Recovery Circuit for 2.5 Gbps NRZ data," *IEEE Journal of Solid State Circuits,* 2001, vol. 36, pp. 432 – 439.

5. R. Farhad-Rad, *A CMOS 4-PAM Multi-Gbps Serial Link Transceiver,* Ph.D. Dissertation, Stanford University, 2000.

6. M. Horowitz, C. K. K. Yang and S. Sidiropoulos, "High Speed Electrical Signaling: Overview and Limitations," *IEEE Micro,* Jan/Feb 1998, pp. 12-24.

7. J. Redd, "A second look at jitter: Calculating bit error rate," *Electronic Engineering Design,* August 2002.

8. Maxim Integrated Products, *Jitter Specification Made Easy: A heuristic discussion of Fibre Channel and Gigabit Ethernet Methods,* Application Note HFAN-4.3.0, February 2001.

9. J. Redd, "Synchronization and clock recovery – an analog guru looks at jitter," *Planet Analog,* Aug. 2001.

10. K. Kundert, "Modeling Jitter in PLL based Frequency Synthesizers," *The Designer's Guide,* Apr. 2003.

11. K. Kundert, "Predicting the Phase Noise and Jitter of PLL based Frequency Synthesizers," *The Designer's Guide*, Nov. 2003.

12. A. Schultz, "Random Jitter – What is really going on?," *Electronic Engineering Design,* Feb. 2002.

13. G. Wei, J. Stonick, D. Weinlader, J. Sonntag and S. Searles, "A 500 MHz MP/DLL Clock Generator for 5 GBPS Backplane Transceiver in 0.25 μm CMOS," 2003 *IEEE International Solid State Circuits Conference, Digest of Technical Papers,* Feb. 2003, pp. 464-465.

14. J. Zerbe, C. Werner, V. Stojanovic, F. Chen, J. Wei, G. Tsang, D. Kim, W. Stonecypher, A. Ho, T. Thrush, R. Kollipara, G-J. Yeh, M. Horowitz and K. Donnelly, "Equalization and Clock Recovery for a 2.5-10 GBPS 2-PAM/4-PAM Backplane Transceiver Cell," 2003 *IEEE International Solid State Circuits Conference, Digest of Technical Papers,* Feb. 2003, pp. 80-81.

15. Vitesse Semiconductor Corporation, *VSC 870 High Performance Serial Backplane Transceiver,* Data Sheet, June 2001.

16. J. M. Khoury and K. R. Lakshmikumar, "High Speed Serial Transceivers for Data Communication Systems," *IEEE Communications Magazine,* July 2001, pp. 160-165.

17. C. McKelvey, "Real Data cuts timing mumbo-jumbo for cascaded PLL designs," *Planet Analog*, Sept. 2002.

18. J. H. Day, "Analog World puts a new spin on PLLs, frequency conversion," *Planet Analog,* June 2002.

19. Q. Du, "Delta-Sigma Fractional N Synthesizers Enable Low-Cost, Low-Power, Frequency Agile Software Radio," *Planet Analog,* Sep. 2000.

20. N. Filiol, T. Riley, C. Plett and M. A. Copeland, "An Agile ISM Band Frequency Synthesizer with Built-in GMSK Data Modulation," *IEEE Journal of Solid State Circuits,* 1998, vol. 33, pp. 998-1008.

21. I. Galton, "Delta-Sigma Fractional N Phase Locked Loops," *University of California, San Diego*, pp. 23-33.

22. I. Galton, W. Huff, P. Carbone and E. Siragusa, "A Delta-Sigma PLL for 14-b, 50 kS/s Frequency to Digital Conversion of a 10MHz FM Signal," *IEEE Journal of Solid State Circuits,*1998, vol. 33, pp. 2042-2053.

23. C. Heng and B. S. Song, "A 1.8 GHz CMOS Fractional N Frequency Synthesizer with Randomized Multiphase VCO," *IEEE Journal of Solid State Circuits,* 2003, vol. 38, pp. 848-854.

24. M. Kozak and I. Kale, "A Pipelined Noise Shaping Coder for Fractional N Frequency Synthesis," *IEEE Transactions on Instrumentation and Measurement,* 2001, vol. 50, pp. 1154-1161.

25. B. H. Park, *A 1-GHz Low Power, Low Phase Noise, CMOS Fractional-N Frequency Synthesizer with on-chop LC VCO for Wireless Communications,* Ph.D. Dissertation, Georgia Institute of Technology, Oct. 1997.

26. W. Egan, *Frequency Synthesis by Phase Lock*, 2nd Edition, New York: John Wiley & Sons, 2000.

27. T. Weigandt, *Low Phase Nois, Low Timing Jitter Design Techniques for Delay Cell Based VCOs and Frequency Synthesizers,* Ph.D. Dissertation, University of California, Berkeley, 1998.

28. C. S. Vaucher, *Architectures for RF Frequency Synthesizers,* Boston: Kluwer Academic Publishers, 2002.

29. F. M. Gardner, "Charge Pump Phase Locked Loops," *IEEE Transactions on Communications,* 1980, vol.com-28, pp. 1849-1858.

30. Analog Devices, *A Technical Tutorial on Digital Signal Synthesis*, Analog Devices Inc., 1999.

31. Osicom Technologies, *Direct Digital Frequency Synthesis, A Tutorial*, Osicom Technologies, 2002.

32. B. Goldberg, *Digital Frequency Synthesis Demystified,* Eagle Rock, VA: LLH Technology Publishing, 1999.

33. B. Razavi, "Design of Monolithic PLLs and Clock Recovery Circuits – A Tutorial," *Monolithic Phase Locked Loops and Clock Recovery Circuits: Theory and Design,* New York: IEEE Press, 1996.

34. J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers,* 1960, vol. EC-8, pp. 330-334.

35. L. Weaver and R. Kerr, *High resolution phase to sine conversion,* U.S. Patent 4905177.

36. S. Norsworthy, R. Schreier and G.C. Temes, Editors, *Delta Sigma Data Converters, Theory, Design and Simulation,* New York: IEEE Press, 1997.

37. S. Mortezapour and E. K. F. Lee, "Design of a low-power ROM-less DDFS using Nonlinear Digital-to-Analog Converter," *IEEE Journal of Solid State Circuits,* 1999, vol. 34, pp. 1350 – 1359.

38. G. Chien and P. R. Gray, "A 900 MHz Local Oscillator Using a DLL based Frequency Multiplier Technique for PCS Applications," *IEEE Journal of Solid State Circuits,* 2000, vol. 35, pp. 1996-2000.

39. T. D. Riley, M. A. Copeland and T.A. Kwasniewski, "Delta-Sigma modulation in Fractional-N frequency synthesis," *IEEE Journal of Solid State Circuits,* 1993, vol. 28, pp. 553-559.

40. W. Rhee, B. S. Song and A. Ali, "A 1.1 GHz CMOS Fractional-N Frequency Synthesizer with a 3-b Third-Order Delta-Sigma Modulator," *IEEE Journal of Solid State Circuits,* 2000, vol. 35, pp. 1453 – 1460.

41. M. H. Perrott, T. Tewksbury III and C. Sodini, "A 27 mW CMOS Fractional-N synthesizer using digital compensation for a 2.5 Mb/s GFSK modulation," *IEEE Journal of Solid State Circuits,* 1997*,* vol. 32, pp. 2048 – 2059.

42. M. H. Perrott, M. Trott and C. Sodini, "A Modeling Approach for Sigma-Delta Fractional-N Frequency Synthesizers Allowing Straightforward Noise Analysis," *IEEE Journal of Solid State Circuits,* 2002*,* vol. 37, pp. 1028 – 1038.

43. C. Park, O. Kim and B. Kim, "A 1.8 GHz Self Calibrated PLL with Precise I/Q Matching," *IEEE Journal of Solid State Circuits,* 2001, vol. 36, pp. 777-783.

44. H. Mair and L. Xiu, "An Architecture of High Performance Frequency and Phase Synthesis," *IEEE Journal of Solid State Circuits,* 2000, vol. 35, pp. 835-846.

45. S. Lee, B. Kim and K. Lee, "A Fully Integrated Low Noise 1 GHz Frequency Synthesizer Design for Mobile Communication Application," *IEEE Journal of Solid State Circuits,* 1997, vol. 32, pp. 760-765.

46. Z. Shu, K. L. Lee and B. H. Leung, "A 2.4 GHz Ring Oscillator based CMOS Frequency Synthesizer with a Fractional Divider Dual PLL Architecture," *IEEE Journal of Solid State Circuits,* 2004, vol. 29, pp. 452-461.

47. Y. Song and B. Kim, "A 250 MHz Direct Digital Frequency Synthesizer with ΔΣ Noise Shaping," *2003 IEEE International Solid State Circuits Conference, Digest of Technical Papers,* Feb. 2003, pp. 472-473.

48. S. Sidiropoulos and M. Horowitz, "A Semidigital Dual Delay Locked Loop," *IEEE Journal of Solid State Circuits,* 1997, vol. 32, pp. 1683 – 1692.

49. D. J. Foley and M. P. Flynn, "CMOS DLL based 2V 3.2ps jitter 1 GHz Clock Synthesizer and Temperature compensated tunable oscillator," *IEEE Journal of Solid State Circuits,* 2001, vol. 36, pp. 417 – 423.

50. H. Chang, C. H. Sun and S. Liu, "A Low Jitter and Precise Multiphase Delay Locked Loop Using Shifted Averaging VCDL," *2003 International Solid State Circuits Conference, Digest of Technical Papers,* Feb. 2003, pp. 434-435.

51. A. Coban, M. Koroglu and K. A. Ahmed, "A 2.5-3.125 GBPS Quad Transceiver with Second Order Analog DLL based CDRs," *IEEE Journal of Solid State Circuits,* 2005, vol. 40, pp. 1940-1947.

52. Y. J. Jung, S. W. Lee, D. Shim, W. Kim, C. Kim and S. I. Cho, "A Dual Loop Delay Locked Loop Using Multiple Voltage Controlled Delay Lines," *IEEE Journal of Solid State Circuits,* 2001, vol. 36, pp. 784-791.

53. M. H. Perrott, R. T. Baird and Y. Huang, *Digitally Synthesized Loop Filter Circuit Particularly useful for a Phase Locked Loop,* US Patent Pub. No.:US 2002/0089356.

54. J. Kim, S. B. Lee, T. S. Jung, C. H. Kim, S. I. Cho and B. Kim, "A Low Jitter Mixed-mode DLL for High Speed DRAM Applications," *IEEE Journal of Solid State Circuits,* 2000, vol. 35, pp. 1430-1436.

55. Y. Moon, J. Choi, K. Lee, D. K. Jeong and M. K. Kim, "An All-Analog Multiphase Delay Locked Loop Using a Replica Delay Line for Wide Range Operation and Low Jitter Performance," *IEEE Journal of Solid State Circuits,* 2000, vol. 35, pp. 377-384.

56. D. E. Calbaza and Y. Savaria, "A Direct Digital Period Synthesis Circuit," *IEEE Journal of Solid State Circuits,* 2002, vol. 37, pp. 1039-1045.

57. E. Roth, M. Thalmann, N. Felber and W. Fichtner, "A Delay line based DCO for Multimedia Applications Using Digital Standard Cells only," *2003 IEEE International Solid State Circuits Conference, Digest of Technical Papers,* Feb. 2003, pp. 432-433.

58. TSMC: "TSMC 0.13µ Logic 1P8M Salicide 1.2V/2.5V Low-K IMD Spice Model", *Taiwan Semiconductor Manufacturing Company Ltd.,* Nov 2000.

59. A. Rofougaran, M. Rofougaran, J. Rael and A. Abidi, "A 900 MHz CMOS LC oscillator with quadrature outputs," 1996 *IEEE International Solid State Circuits Conference Digest of Technical Papers,* Feb. 1996, pp. 316-317.

60. H. Ahn and D. Allstot, "A Low Jitter 1.9V CMOS PLL for UltraSPARC Microprocessor Applications," *IEEE Journal of Solid State Circuits,* 2000, vol. 35, pp 450-454.

61. V. V. Kaenel, D. Aebischer, C. Piguet and E. Dijkstra, "A 320 MHz, 1.5mW @ 1.35V CMOS PLL for Microprocessor Clock Generation," *IEEE Journal of Solid State Circuits,* 1996, vol. 31, pp. 1715-1722.

62. K. Lee, J. Park, J. W. Lee, S. W. Lee, H. K. Huh, D. K. Jeong and W. Kim, "A Single chip 2.4 GHz Direct Conversion CMOS Receiver for Wireless Local Loop

using Multiphase Reduced Frequency Conversion Technique," *IEEE Journal of Solid State Circuits,* 2001, vol. 36, pp. 800-809.

63. J. G. Maneatis, J. Kim, I. McClatchie, J. Maxey and M. Shankaradas, "Self Biased High Bandwidth, Low Jitter 1-to-4096 Multiplier Clock Generator PLL," *2003 International Solid State Circuits Conference, Digest of Technical Papers,* Feb. 2003, pp. 424-425.

64. J. G. Maneatis, "Low Jitter Process Independent DLL and PLL based on Self biased Techniques," *IEEE Journal of Solid State Circuits,* 1996, vol. 31, pp. 1723-1732.

65. M. Mansuri and C. K. K. Yang, "A Low Power Low Jitter Adaptive Bandwidth PLL and Clock Buffer," *2003 IEEE International Solid State Circuits Conference, Digest of Technical Papers,* Feb. 2003, pp. 430-431.

66. G. Y. Tak, S. B. Hyun, T. Y. Kang, B. G. Choi and S. S. Park, "A 6.3-9 GHz CMOS Fast Settling PLL for MB-OFDM UWB Applications," *IEEE Journal of Solid State Circuits,* 2005, vol. 40, pp. 1671-1679.

67. R. Nonis, N. Dalt, P. Palestri and L. Selmi, "Modeling, Design and Characterization of a New Low Jitter Analog Dual Tuning LC-VCO PLL Architecture," *IEEE Journal of Solid State Circuits,* 2005, vol. 40, pp. 1303-1309.

68. B. Razavi, "A 2 GHz 1.6mW PLL," *IEEE Journal of Solid State Circuits,* 1997, vol.32, pp. 730-735.

69. U. Singh and M. M. Green, "High Frequency CML Clock Dividers in 0.13 μm CMOS operating up to 38 GHz," *IEEE Journal of Solid State Circuits,* 2005, vol. 40, pp. 1658-1661.

70. W. Wilson, U. K. Moon, K. R. Lakshmikumar and L. Dai, "A CMOS Self Calibrating Frequency Synthesizer," *IEEE Journal of Solid State Circuits,* 2000, vol. 35, pp. 1437-1444.

71. S. Ye, L. Jansson and I. Galton, "A Multiple Crystal Interface PLL with VCO Realignment to reduce Phase Noise," *IEEE Journal of Solid State Circuits,* 2002, vol. 37, pp. 1795-1803.

72. A. Hajimiri, S. Limotyrakis and T. H. Lee, "Jitter and Phase Noise in Ring Oscillators," *IEEE Journal of Solid State Circuits,* 1999, vol. 34, pp. 790-804.

73. D. Ham and A. Hajimiri, "Concepts and Methods in Optimization of Integrated LC VCOs," *IEEE Journal of Solid State Circuits,* 2001, vol. 36, pp. 896-909.

74. P. Andreani, X. Wang, L. Vandi and A. Fard, "A Study of Phase Noise in Colpitts and LC-Tank CMOS Oscillators," *IEEE Journal of Solid State Circuits,* 2005, vol. 40, pp. 1107-1118.

75. T. H. Lee and A. Hajimiri, "Oscillator Phase Noise: A Tutorial," *IEEE Journal of Solid State Circuits,* 2000, vol. 35, pp. 326-336.

76. B. Razavi, et al., "Design of High-Speed, Low Power Frequency Dividers and Phase Locked Loops in Deep Submicron CMOS," *IEEE Journal of Solid State Circuits,* 1995, vol. 30, pp. 101-109.

77. Y. Nakasha, et al., "A 0.7mW 2-GHz Dual Modulus Prescaler IC," *IEEE Custom Integrated Circuits Conference, Digest of Technical Papers,* 1994, pp. 16.4.1-16.4.4.

78. K. Maemura, et al., "A 1 GHz, 2mA, 5 V GaAs 128/129 Prescaler IC," *IEEE GaAs IC Symposium,* 1987, pp. 273-276.

79. S. Kataoka, et al., "A 0.8 V Prescaler IC," *IEEE Custom Integrated Circuits Conference, Digest of Technical Papers,* 1992, pp. 10.5.1-10.5.4.

80. M. Mizuno, et al., "A 3mW 1.0 GHz Silicon-ECL Dual Modulus Prescaler IC," *IEEE Journal of Solid State Circuits,* 1992, vol. 27, pp. 1794-1797.

81. R. Rogenmoser, Q. Huang and F. Piazza, "1.57 GHz Asynchronous and 1.4 GHz Dual Modulus Prescalers in 1.2 μm CMOS," *IEEE Custom Integrated Circuits Conference, Digest of Technical Papers,* 1994, pp. 16.3.1-16.3.4.

82. N. Foroudi and T. A. Kwasniewski, "CMOS High Speed Dual Modulus Frequency Divider for RF Frequency Synthesis," *IEEE Journal of Solid State Circuits,* 1995, vol. 30, pp. 93-100.

83. J. Craninckx and M. Steyaert, "A 1.75 GHz, 3 V Dual Modulus Divide by 128/129 Prescaler in 0.7 μm CMOS," *21st ESSCIRC*, 1995, pp. 19-22.

84. R. Rogenmoser, et al., "A 1.16 GHz Dual Modulus 1.2 μm CMOS Prescaler," *IEEE Custom Integrated Circuits Conference, Digest of Technical Papers,* 1993, pp. 27.6.1-27.6.4.

85. H. I. Cong, et al., "A Multigigahertz CMOS Dual Modulus Prescaler IC," *IEEE Journal of Solid State Circuits,* 1988, vol. SC-23, pp. 1189-1193.

86. B. Chang, J. Park and W. Kim, "A 1.2 GHz CMOS Dual Modulus Prescaler Using New Dynamic D-Type Flip Flops," *IEEE Journal of Solid State Circuits,* 1996, vol. 31, pp. 749-752.

87. P. Larsson, “High Speed Architecture for a Programmable Frequency Divider and a Dual Modulus Prescaler,” *IEEE Journal of Solid State Circuits,* 1996, vol. 31, pp. 744-748.

Vita

Giri NK Rangan was born in Bangalore, India, in 1968. He was born to Mr. N. Kasturi Rangan and Mrs. Godamani. He earned his Bachelor of Engineering degree in 1990 from the University College of Engineering, Bangalore University. After a brief stint in industry in Bangalore, he joined the Department of Electrical and Computer Engineering, Oregon State University in 1991 and earned his MSEE degree in 1993. Since 1993, he has been in the semiconductor industry in the San Francisco Bay Area and Austin holding design engineering positions at various levels. He entered the Department of Electrical and Computer Engineering at the University of Texas at Austin in 1997 for his doctoral work. His professional interests are in high precision data converters, analog and digital signal processing, system modeling and simulation, digital communication and control systems. His other interests include basketball, chess, astronomy and Indian classical music.

Permanent address: 29, 4th Temple Street, 15th Cross, Malleswaram, Bangalore 560003 India.

This dissertation was typeset in Microsoft Word™ by the author.