Design and Analysis of Agile Frequency Synthesizers for Mobile Communications

by

Amr Mohamed Fahim

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2000

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Acknowledgements

After thanking God Almighty for giving me the strength and ability to complete this work, I would like to express my sincere thanks to my parents who have instilled in me the drive and encouragement to complete this work.

I would also like to express my appreciation and gratitude to my thesis supervisor, Professor M. I. Elmasry for his encouragement, guidance and support in my research. He gave me valuable insight and motivation for this work.

I would also like to acknowledge and thank all my colleagues in the VLSI Research Group for their support and useful discussion.

This work is supported in part by research funding from NSERC, Canadian Microelectronics Corporation (CMC) and Micronet. Their contributions are greatly appreciated.

Abstract

The wireless market has experienced an exponential growth over the past few years. To sustain this growth along with the increasing demands of new wireless standards, the cost, battery lifetime, and performance of wireless devices must all be enhanced.

With the advancement of radio frequency (RF) technology and requirement for more integration, new RF wireless architectures are needed. One of the most critical components in a wireless transceiver is the frequency synthesizer. It largely affects all three dimensions of a wireless transceiver design: cost, battery lifetime, and performance. In this thesis, new generations of RF synthesizer and transmitter architectures allowing low-power, high-performance and lower cost are presented.

The common approach to frequency synthesis design for wireless communication is to design an analog-compensated fractional-N phase-locked loop (PLL). However, this technique suffers from lock time limitations and lack of adequate fractional spur suppression for third generation wireless standards. In this work, two new fast lock PLL architectures are reported to overcome the above mentioned limitations with the aid of digital signal processing. One such scheme makes use of modified digital sigma-delta modulator to completely randomize fractional spurs present in fractional-N PLLs as well as to reduce the level of phase noise produced by the sigma-delta modulator. This aids in fully integrating a high-performance PLL frequency synthesizer, and hence reducing cost. The use of this architecture for closed loop modulation is also examined.

In another approach, a high frequency digital comparator aids in quickly acquiring frequency lock. Very fast lock times are achievable using this architecture. This architecture removes the PLL's frequency resolution dependence on the loop filter

parameters. This helps to drastically reduce the size of the loop filter components, and enables them to be integrated on-chip. Although more suitable for low frequency resolution applications, such as wireless LAN and cordless, this architecture may be modified to obtain higher frequency resolutions. The major advantages of this architecture include low-cost, low-power, and a fully monolithic solution.

Throughout this work, low-power has been achieved by both architectural techniques as well as circuit techniques. Architectural techniques enable tighter integration of the PLL's loop components on a single chip as well as faster lock time. Since the proposed techniques rely heavily on digital signal processing, low-power, high-performance digital logic families are reported. It is demonstrated how these logic families may be used in the frequency synthesizer architectures detailed above. Although differential in nature, it is demonstrated that the use of these logic families also helps to reduce area.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The past few years have seen an explosive growth in the wireless telecommunications market. Signs of this explosive growth are everywhere. It is estimated that approximately 20,000 people join the cellular phone market daily in the United States alone. It is now estimated that more that half of homes in the United States now have cordless phones. More than 280 million people currently subscribe to cellular phones. Just ten years ago, there were only 2 million subscribers. Currently, the wireless communications industry generates $100 billion in revenue (in the year 2000). This revenue does not only come from cellular phones. It is estimated that 30% of the computer market revenue comes from portable computers (laptops and notebooks). These computers would require wireless access. Furthermore, dedicated e-mail wireless terminals are now available on the market. It is estimated that 10 million people are expected to use wireless networks to transmit data by the end of the year

2000.

The dream that anyone can communicate with anyone else "anytime, anywhere" and exchange "any kind" of information appears to be around the corner. Today, voice, data, and video can be transmitted over the wireless channel from one portable unit to another. The key to the success of the wireless market is affordability due to advances in semiconductor technology. As transistor dimensions shrink in size, more components are moved from the board to the chip. This not only helps to drastically reduce cost and area, but also helps in removing power consuming chip-to-chip communications. The present goal is to reduce both power consumption and the price of cellular phones by 30% each year. This cannot be achieved by relying on semiconductor technology alone. Circuit and architectural techniques must be investigated.

One pivotal block used in the wireless radio terminal is the frequency synthesizer. The frequency synthesizer is responsible for generating the carrier frequencies for both the transmitter and receiver. A wireless radio may have a shared frequency synthesizer for both transmit and receive, or two dedicated frequency synthesizers. The performance of the frequency synthesizer directly affects the performance of transceiver. The receiver sensitivity relies on how accurately the frequency synthesizer produces its output frequency.

The amount of time required for the synthesizer to generate a certain reference frequency (called lock time) also affects performance and power consumption. In many wireless standards, time division multiple access (TDMA) is used. In such standards, each user is given a time slice to transmit the data. After this time slice, the frequency synthesizer (of the base station) must tune to the transmission frequency of another user. If the frequency synthesizer lock time is reduced, the bandwidth allocated to each user can be increased. The frequency synthesizer is also the most active component in the wireless radio. Since the radio must periodically check for any incoming data, the synthesizer must be turned on for that amount of time. The

transceiver must also be on while the synthesizer is locking to the desired channel. In this time period, the transceiver is wasting power waiting for the synthesizer to lock.

Integrating the entire frequency synthesizer on a single chip is a challenging task. This is related to the stringent performance requirements put on the synthesizer. On-chip passive components used in the frequency synthesizer usually have much less quality than discrete components. Also, interference from one of the frequency synthesizer's components on another may cause performance degradation. This leads to extra board space, which entails higher cost, size, and power dissipation.

One approach to integrate the entire frequency synthesizer on a single chip is to use a specialized technology. This is regarded as an expensive solution and has not received widespread industrial acceptance. An alternative solution is to use a cheap technology, such as CMOS, and try to achieve integration through architectural innovations. This technique requires the redesign of the frequency synthesizer architecture to suit the current capabilities of the technology. Frequency synthesizers for wireless communications are usually implemented as phase locked loops (PLL). A PLL has both digital and analog components. In this study, digital solutions for improving the performance, integration, and lock time of PLLs are examined. In chapter 2, the basic operation of the PLL as well as a survey of current fast lock PLL architectures are given. In chapter 3, analytical solutions illustrating the performance of the most commonly used PLL architectures are given. In chapter 4, an implementation of a MASH sigma-delta PLL is given. This architecture attempts to provide a digital solution to eliminating the spurious noise produced by PLLs. It also offers high frequency resolution. In chapter 5, a novel wideband PLL architecture based on sigma-delta modulation is given. It is demonstrated that this architecture can also be used for closed loop modulation to directly transfer digital baseband data to radio frequency signals. In chapter 6, a new low-power fast lock PLL architecture is given. It is demonstrated that lock times much less than current PLL architectures may be achieved, while still meeting the performance requirements of wireless systems.

Furthermore, it is demonstrated that the loop filter components in this architecture can be easily integrated on-chip. Finally, in chapter 7, digital logic families are detailed which may be used to further reduce the power consumption of the digital sections of the PLLs implemented. Such reductions may help to integrate both digital and analog components of the transceiver onto a single chip.

Chapter 2

Background

2.1 Frequency Synthesizer Requirements in Wireless Systems

Perhaps the most critical component in a wireless transceiver is the frequency synthesizer. The frequency synthesizer is used to generate an accurate local oscillator (LO) output frequency. The main frequency synthesizer parameters include frequency range, frequency resolution, switching speed, spurious output, phase noise, and power consumption. These parameters are specified by the specific standard being implemented. Table 2.1 summarizes some of the 2^{nd} generation wireless communications standards that are currently used [27].

Table 2.1. Summary of 2G wireless communications standards [27]

	IS-95	IS-54 / IS-136	GSM	GPRS	DCS-1800	CT2	DECT	IEEE 802.11
F in MHz (Rx/Tx)	869-894 824-849	869-894 824-849	925-960 880-915	925-960 880-915	1805-1880 1710-1785	864/868	1880-1900	2400-2483
Access Method	CDMA/ FDM	TDMA/ FDM	TDMA/ FDM	Packet	TDMA/ FDM	TDMA/ FDM	TDMA/FDM	CSMA
Duplex Method	FDD	FDD	FDD	FDD	FDD	TDD	TDD	TDD
# chan.	798	832	124	374	1600	40	10	FHSS:79 DSSS:7
Chan. spacing	1250 KHz	30KHz	200KHz	200KHz	200KHz	100KHz	1.728 MHz	FHSS:1 DSSS:11
Modulation	QPSK/ OQPSK	$\pi/4$-DQPSK	GMSK (0.3 GF)	GSMK (0.3 GF)	GMSK (0.3 GF)	GFSK (0.5GF)	GFSK 0.5GF	GFSK, DQPSK
FS switching time	Slow	Slow	577us	200us	577us	1ms	30usBS 450usHS	300us
Freq. Accuracy		200Hz (0.2ppm)	50Hz (0.1ppm)	50Hz (0.1ppm)	100Hz (0.1ppm)	10KHz 10ppm	50KHz 25ppm	75KHz 30ppm

As shown in Table 2.1, all frequency standards demand switching speeds of at most one millisecond and frequency accuracies less than 30ppm. Note that all the cellular standards require the most frequency accuracy of less than 0.2ppm. Also note that the GSM (Global System for Mobile Communications) standard is the most widely used digital standard worldwide and it demands the most from frequency synthesizers. Both the frequency accuracy and switching speed of GSM are very aggressive.

In an effort to extend the use of wireless transmission from merely voice, new third generation standards are beginning to emerge. The objective of 3rd generation wireless standards can be summarized as:

- Full coverage and mobility for 144Kbps, preferable 384Kbps;
- Limited coverage and mobility for 2Mbps;
- High spectrum efficiency compared to existing systems;
- High flexibility to introduce new services.

The bit rates of 384Kbps and 2Mbps were set in order to meet current video transmission standards of H.263 [28] and MPEG-2 [29], respectively. There are two 3rd

generation standards that have emerged. One is called wideband CDMA [30] (or WCDMA). WCDMA standard has received wide acceptance in Europe, Japan, and North America. This standard was made backwards compatible with the GSM standard. Another version of WCDMA, called cdma2000 [31], was created to be backward compatible with the IS-95 standard.

Wideband CDMA has a nominal bandwidth of 5MHz. This has been done in order to incorporate data rates of 144Kbps, 384Kbps, and 2Mbps. In order to be compatible with GSM, different grades of data transmission have been created. The lowest grade being a 200KHz channel (which is the same as the GSM). Higher data rates can be achieved by combining several GSM channels.

Current TDMA-based technologies [32] (GSM, IS-136, IS-54) use a single voice channel for data per user, which delivers data at a rate of 9.6Kbps and 14.4Kbps. Today, new standards enabling high-speed wireless data transmission are emerging. These standards, which exhibit some features of 3rd generation standards, are extensions of current 2nd generation standards. Such standards are referred to as 2.5 generation or 2+ generation standards. The two most popular are the GPRS [33] (general-packet-radio service) and HSCSD [33] (high-speed circuit switched data) standards. HSCSD is a circuit switched data transmission that can offer as much as 76.8Kbps. This is done by dedicating more than one channel to a user. Circuit switched data transmission entails that a dedicated connection must be established and this connection cannot be broken until all transmission is complete. GPRS, on the other hand, is a packet switched data transmission that can offer data rates over 100Kbps. In packet switched transmission, a physical connection only exists during the transmission burst. This means that the channel is used more efficiently. This also means that in GPRS, users could be "virtually" connected for hours at a time and only incur connection charges during transmission bursts. GPRS is expected to be the main enabling technology for high-speed wireless internet access in the near future.

As shown in Table 2.1, GPRS requires fast switching from one channel to another. This is due to the fact that GPRS uses a combination of TDMA and FDMA. In this type of wireless transmission, each user is reserved a certain time slice (called a frame) to transmit the data. When the frame transmission is over, the next user transmits his own frame, which can be transmitted at a different frequency. In this case, a certain time period must be allocated for the frequency synthesizer to complete switching from one frequency to another. If less time is allocated for this frequency switching, more time can be allocated for data transmission, and hence data transmission rates can be increased. In the GPRS standard, the frequency synthesizer is given less than 200μs to switch from one frequency to another. Frequency synthesizer switching time is expected to be constrained even further as data rates increase.

Another important requirement in wireless standards that directly affect frequency synthesizers is spectral purity. Table 2.2 shows the spectral purity requirements of some of the 2nd generation wireless standards. The spectral purity requirements depend on several parameters, which include the data signal's dynamic range, channel spacing, and the modulation technique.

Table 2.2. Spectral purity requirements of some 2nd generation wireless standards

Standard	**Phase noise requirement**
GSM	-141dBc/Hz @ 3MHz
DECT	-131dBc/Hz @ 4.7MHz
PHS	-96dBc/Hz @ 100KHz
DCS-1800	-123dBc/Hz @ 600KHz
PDC	-120dBc/Hz @ 50KHz

2.2 Wireless Transceiver Architectures

There are three main types of wireless receiver architectures, as shown in Fig. 2.1(a)-(c). The first one is the double-IF receive architecture in such an architecture, the input RF frequency is down mixed to two intermediate frequencies (IF) before it is converted to a baseband signal. Although this yields high spectral purity, this architecture

requires the most off-chip components. It also has the most demand from the frequency synthesizer, since three different frequencies must be generated. Another architecture, single-IF receiver, attempts to reduce the number of off-chip SAW filters by reducing the number of intermediate frequencies to one. The third architecture, the direct conversion architecture, no intermediate frequencies are used, which means that no SAW filters are used. This is the most suitable architecture for full integration, and it has the least demand from the frequency synthesizer.

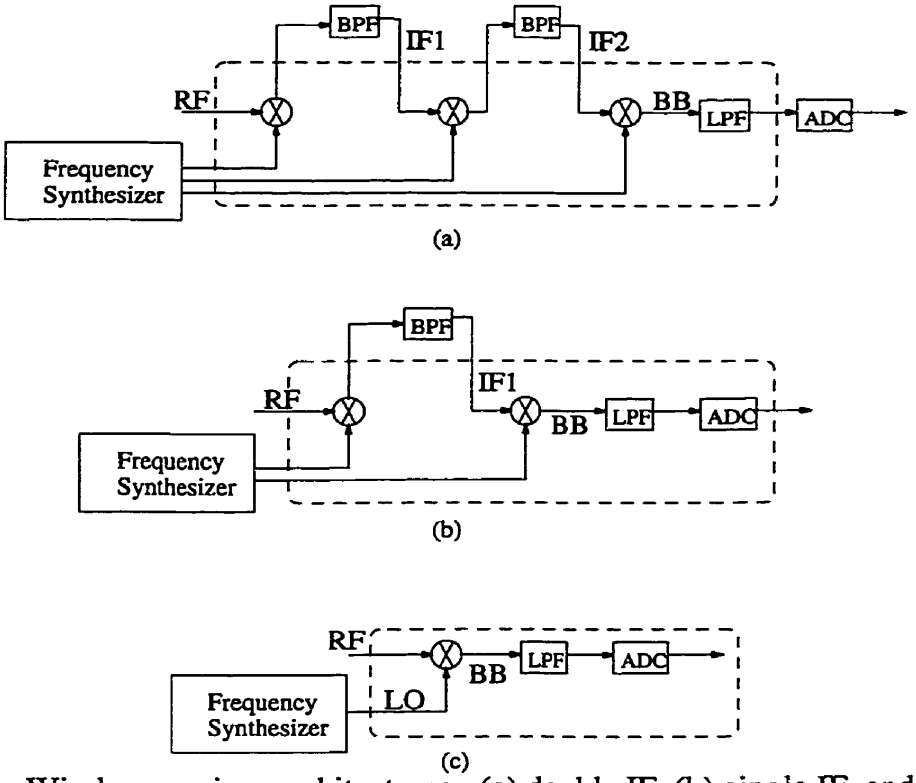


Figure 2.1. Wireless receiver architectures: (a) double IF, (b) single IF, and (c) direct conversion architecture

2.3 Classical Phase-Locked Loop

2.3.1 Basic Operation

The Phase-Locked Loop (PLL) is the most commonly used form of frequency

synthesizer used in wireless systems. The classical PLL consists of a voltage-controlled oscillator (VCO), a frequency divider, a phase frequency detector (PFD), and a loop filter (LF) as shown in Fig. 2.2. Now the operation of each component is described. The VCO produces a frequency proportional to its input voltage. More specifically,

$$\omega_2 = \omega_0 + K_0 u_f(t) \tag{2.1}$$

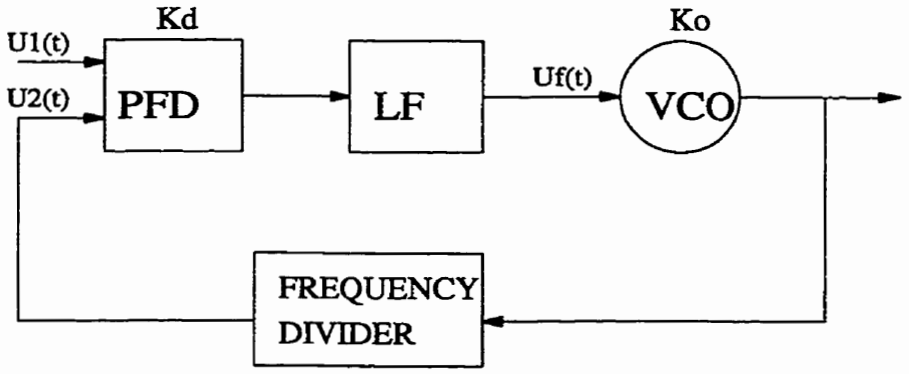


Figure 2.2. Classical PLL

where ω_0 is the free oscillation frequency (frequency at which the VCO oscillates at when no input frequency is applied) measured in rad/s, and K_0 is the VCO gain (measured in $s^{-1}V^{-1}$). The PFD compares the input signal, $u_1(t)$, and the feedback signal, $u_2(t)$, and produces an output which is proportional to the phase difference between the two signals. In terms of a mathematical description, the operation of the PFD may be expressed as

$$u_d(t) = K_d \theta_e \tag{2.2}$$

where K_d is the gain of the PFD and θ_e is the difference in phase between $u_1(t)$ and $u_2(t)$ (measured in rads). The units of K_d is $V \cdot rad^{-1}$. The purpose of the loop filter is to control the bandwidth of the closed loop system. The loop bandwidth, in turn, controls the *settling time*, or *lock-in time* of the PLL. The loop filter, in most cases, is a second or third order low-pass filter. The frequency divider allows the input frequency to be lower than the output frequency by a factor of N. Note that if the value of N is

variable, then the PLL-based frequency synthesizer can be used to synthesize different frequencies.

The operation of the PLL is actually quite simple. If the input signal's phase is lagging behind the phase produced by the VCO, this means that the VCO frequency is too high and hence must be lowered. The phase detector handles this by producing an inhibiting signal to the loop filter, which in turn lowers the voltage input to the VCO, $u_f(t)$. As can be shown by Equation (2.1), lowering $u_f(t)$ amounts to lowering $\omega_2(t)$, the output frequency of the VCO, which is what is desired. On the other hand, if the input signal's phase is leading that of the VCO, the PFD produces an excitatory signal to the loop filter, which increases the output voltage of the loop filter, and in turn, increases the output frequency of the VCO, as desired.

Frequency accuracy is ensured by the (1/s) term in the VCO, which acts as an integrator. Typically, another pole is contributed by the loop filter in order to ensure that the phase is also equalized to the input reference signal. This configuration would ensure both frequency and phase stability of the PLL's output.

2.3.2 PLL Performance Measures

There are several figures of merit which objectively determine the PLL's performance. These figures of merit may be broken down into three categories[1]. One category of such figures describe the *acquisition characteristics* of the PLL. The four acquisition characteristics measures are:

- *Pull-in range* $\Delta\omega_P$: This is the maximum frequency deviation range which the PLL can tolerate without becoming unstable. If the frequency change in the input is smaller than this, then the PLL will eventually relock into the new frequency.
- *Pull-in time* T_P: Time required for the pull-in process.
- *Lock range* $\Delta\omega_L$: This is the maximum frequency step that can be applied to the input of the PLL while still maintaining lock to within a single beat note (phase

difference of 2π) between the reference frequency and the output frequency.

- *Lock-in time* T_L: Time required for the lock-in process.

Another category of performance measure determines the *tracking capability* of the PLL. The only performance measure belonging to this category, which is considered in this study, is:

- *Pull-out range* $\Delta\omega_{PO}$: This is the maximum frequency step that can be applied to the input of the PLL while still having the PLL to re-lock quickly. If the frequency step is greater than this quantity but less than the pull-in range, relocking will be a slow process.

Fig. 2.3 shows the relationship between the different frequency ranges of the acquisition characteristics and tracking capability characteristics of the PLLs.

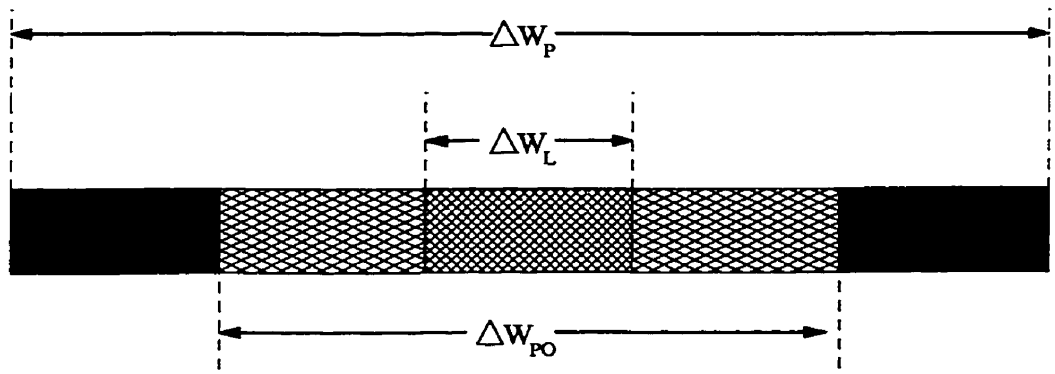


Figure 2.3. Relationship between various frequency ranges

The third category is sometimes referred to in some texts [1,2] as the *noise immunity*, or the *noise rejection capability*, of the PLL. The total RMS phase error in a wireless transceiver can be contributed from several factors. The frequency synthesizer is one of them. When designing a PLL frequency synthesizer for wireless applications, the RMS phase error contribution of the frequency synthesizer should be no more than 2 degrees [33]. The PLL's RMS phase error, is also referred to as close-in phase noise, which is the phase noise within the closed loop bandwidth of the PLL. It should be noted that the close-in phase noise determines the bit error rate (BER) of the entire

system. The larger the bandwidth, the more phase noise is integrated. Noise within the closed loop bandwidth of the PLL can cause modulation on the VCO control voltage, which in turn causes phase excursions at the VCO's output. The peak excursion $\Delta\varphi$ can be described as

$$\begin{aligned} s_{VCO}(t) &= A\cdot\cos(\omega_c t+\Delta\varphi\sin\omega_m t) \\ &= A[\cos\omega_c t\cdot\cos(\Delta\varphi\sin\omega_m t)-\sin\omega_c t\sin(\Delta\varphi\sin\omega_m t)] \end{aligned} \tag{2.3}$$

If $\Delta\varphi$ is small (which is the case for 2 degrees), then Equation 2.3 simplifies to

$$s_{VCO}(t) = A\left\{\cos\omega_c t-\frac{\Delta\varphi}{2}[\cos(\omega_c+\omega_m)t-\cos(\omega_c-\omega_m)t]\right\} \tag{2.4}$$

This equation shows that the noise spectrum is a sum of a large number of offset frequencies around the carrier due to phase modulation of the VCO. The level of noise sidebands on each side of the carrier is

$$L(\omega)=20\cdot\log_{10}\left(\frac{\Delta\varphi}{2}\right) \tag{2.5}$$

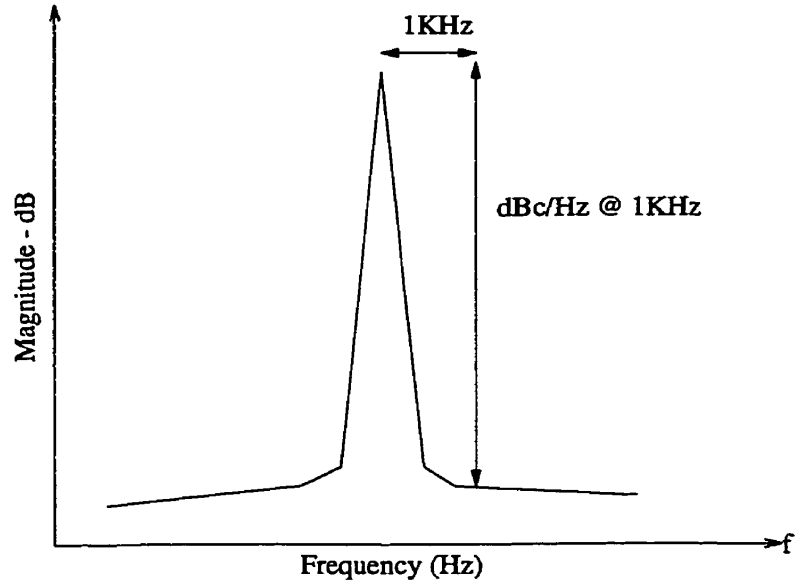


Figure 2.4. Measure of SSB phase noise in frequency synthesizers

Integrating over the frequency range of interest (closed loop PLL bandwidth, BW) gives the final equation for the close-in phase noise as

$$\phi_{PLLRMS} = \sqrt{\int 2 \cdot L(\omega) d\omega} \tag{2.6}$$

Single sideband (SSB) phase noise outside the loop bandwidth is also important. It is defined as the distance (in dB) from the amplitude of the center frequency to the amplitude of the sidelobe at a specified frequency distance (as shown in Fig. 2.4). The unit of measure is dBc/Hz @ carrier offset frequency (in Hz).

It is important to note that phase noise is related to the noise of the semiconductor devices being used[73,74]. For metal oxide semiconductor (MOS) devices, the two main contributors of noise are flicker (or 1/f) noise and thermal noise. When implementing a VCO in MOS technology, the transfer function of the VCO becomes $H(s)=s^{-2}$, and the flicker noise is translated to f^{-3} noise. Thermal noise is a form of amplitude noise. When applied to VCOs, the thermal noise modifies the zero crossings of the output of the VCO. This random modification of the zero crossings results in phase errors, or phase noise.

2.3.3 Charge-pump PLL

In order to have an infinite capture range, an active filter must be used for the loop filter. Unfortunately, this is not a suitable structure for wireless applications since an active loop filter introduces a significant amount of noise (since active components are on whenever the PLL is turned on). Alternatively, wireless systems usually employ a charge-pump based PLL [37]. Fig. 2.5 shows a block level diagram of a charge-pump based PLL. The reasons for their popularity include extended lock range and low cost [37]. As its name implies, a charge pump is responsible for injecting a fixed amount of current to the loop filter. The charge pump is controlled by the PFD. If UP=1, then current is injected into the loop filter, and hence the voltage across the loop filter is increased. If DN=1, on the other hand, an equal amount of current is drawn out of the

loop filter, and thereby decreasing the voltage across the loop filter. Note that during locked condition the current sources of the charge pumps are on for a small fraction of the time, as opposed to being on all the time as in PLLs utilizing active filters. It is for this reason that charge pumps exhibit less in-band noise and are preferred when implementing PLLs fore wireless systems. Unless otherwise stated, all PLLs from this point forward are assumed to employ charge pumps.

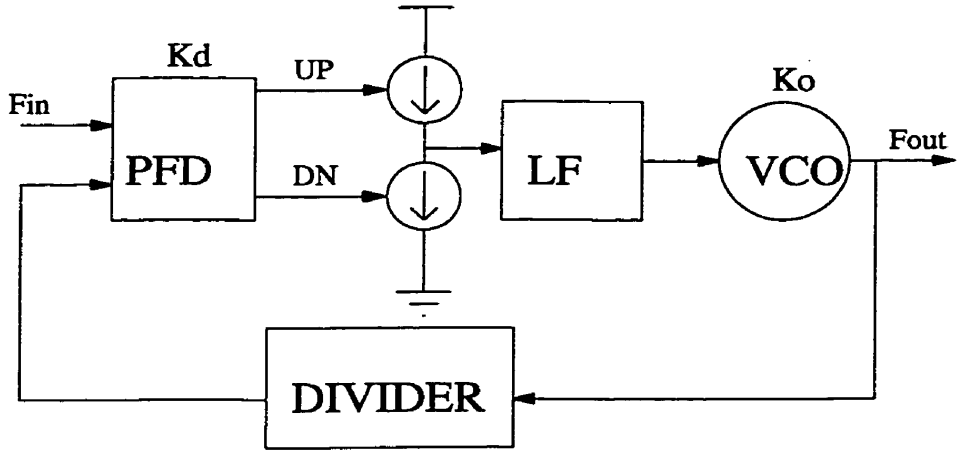


Figure 2.5. Charge pump based PLL

2.3.4 Closed Loop PLL Analysis

For the classical PLL, shown in Fig. 2.6, it is instructive to obtain the various transfer functions of the PLL. Assuming that the closed loop bandwidth of the PLL is less than the reference frequency by at least an order of magnitude, the PLL may be thought of as a linear device. The transfer functions are given as

$$H_1(s) = \frac{K_d K_o F(s)}{s + K_d K_o F(s)/N} \quad \text{from input to VCO output} \tag{2.7}$$

$$H_2(s) = \frac{sK_d F(s)}{s + K_d K_o F(s)/N} \quad \text{from input to LF output} \tag{2.8}$$

$$H_3(s) = \frac{sK_d}{s + K_d K_o F(s)/N} \quad \text{from input to PFD output} \tag{2.9}$$

$$H_4(s) = \frac{K_o F(s)}{s + K_d K_o F(s)/N} \quad \text{from PFD to VCO output} \tag{2.10}$$

$$H_5(s) = \frac{K_o}{s + K_d K_o F(s)/N} \quad \text{from LF to VCO output} \tag{2.11}$$

These equations may be used to evaluate the effect of injecting noise at different points in the PLL on the output response of the PLL.

Equation (2.7) is the basic PLL transfer function. Assuming a charge pump PLL with a first order loop filter, $H_1(s)$ may be rewritten as:

$$H_1(s) = \frac{\frac{I_p}{2\pi} K_o (R + \frac{1}{sC})}{s + \frac{I_p}{2\pi} K_o (R + \frac{1}{sC})/N} \tag{2.12}$$

Equation (2.12) is a second order system. The natural frequency and damping factor are then given as

$$\omega_n = \sqrt{\frac{K_o I_p}{2\pi NC}} \tag{2.13}$$

$$\zeta = \frac{\omega_n RC}{2} \tag{2.14}$$

The 3-dB bandwidth of the 2nd order PLL is given as

$$\omega_{3dB} = \frac{\omega_n}{2\pi}\sqrt{2\zeta^2 + 1 + \sqrt{(2\zeta^2 + 1)^2 + 1}} \text{ Hz} \tag{2.15}$$

Since the loop filter introduces one pole and the VCO acts as a second pole, both the frequency and phase are maintained at a constant value during locked conditions.

As long as the PLL is within the lock-in range, the PLL can lock to within one beat note. The lock-in range is given as

$$\Delta\omega_L = 4\pi\zeta\omega_n \tag{2.16}$$

Using equations (2.13) and (2.14), the lock-in range can also be expressed as

$$\Delta\omega_L = \frac{RK_oI_p}{N} \quad \text{Hz} \tag{2.17}$$

As long as the frequency deviation is less than the lock-in range, the PLL will lock within one beat note (i.e. within 2π radians), which is given as

$$T_P = \frac{(\Delta\omega)^2}{2\zeta\omega_n^3} \tag{2.18}$$

where $\Delta\omega$ is the step frequency change in rads/sec. Using equations (2.13) and (2.14), the lock-in time can also be expressed as

$$T_P = \frac{(\Delta f)^2}{\frac{R}{C}\left(\frac{K_oI_p}{N}\right)^2} \quad \text{seconds} \tag{2.19}$$

where Δf is the step frequency change in hertz.

2.3.5 Phase noise performance

As eluded in section 2.2.2, there are two types of phase noise in PLL frequency synthesizers. The first is close-in phase noise, usually caused by the external frequency reference source and noise in the charge pump circuitry, and the other is short term phase noise, usually contributed by the VCO [3]. Fig. 2.7 demonstrates the effect of the PLL loop on the phase noise of the VCO. It seems that the closed loop PLL effectively suppresses the phase noise of the VCO up to its bandwidth. To understand how this works, consider a simple PLL model with additive noise injected into the VCO input. If the input is taken to be from the input referred VCO noise source and the output from the VCO output, the transfer function becomes that of (2.11). Assuming a first order loop filter and a linear charge pump based PLL, (2.11) becomes

$$H_5(s) = \frac{K_os^2}{s^2 + \frac{K_0I_pR}{2\pi NC}s + \frac{K_0I_p}{2\pi NC}} \tag{2.20}$$

This is clearly a high pass filter. The denominator of $H_5(s)$ is the same as $H_1(s)$. This means that the bandwidth of the high pass filter is equal to the bandwidth of the PLL, which is consistent with Fig. 2.6. The larger the PLL bandwidth, the more VCO generated phase noise is suppressed. This makes sense since the larger the PLL's bandwidth, the faster it can correct for any phase deviations that the VCO produces and hence the magnitude of the phase deviations are limited. This type of phase noise is referred to as short-term phase noise.

Another potential source of phase noise in phase locked-loops is from power supply bounces. For PLL-based clock generators used in microprocessor and digital signal processor (DSP) chips, this is a major source of phase noise, or jitter, since the PLL resides on the same chip with a large digital circuit block. In the design of front-end wireless systems, however, the phase locked-loop frequency synthesizer resides on an independent chip and its power supplies are well regulated. Therefore, this source of phase noise is not a major issue in the design of front-end phase locked-loops for wireless applications.

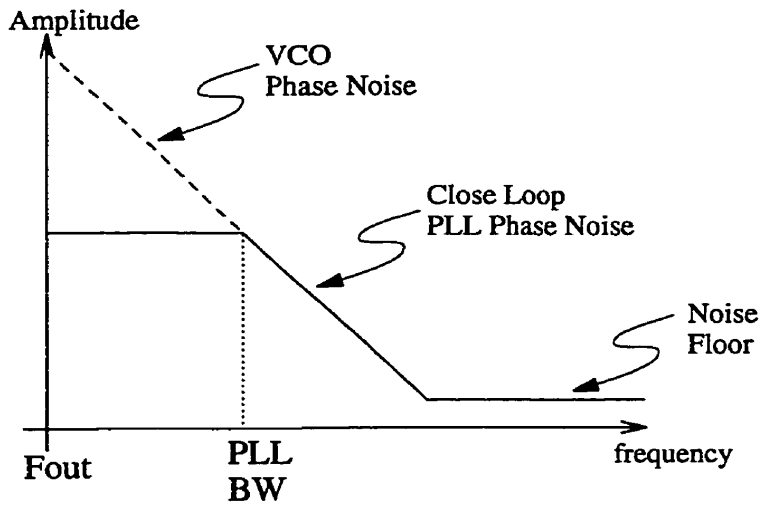


Figure 2.6. Phase noise in a closed loop PLL system

Long term phase noise (or more commonly referred to as close-in phase noise) results from the phase noise contributed by all other PLL components as well as the crystal oscillator driving the PLL. Noise generated by the crystal oscillator is shaped by the transfer function $H_1(s)$, which is a low pass filter. The other transfer functions $H_2(s)$ to $H_4(s)$ are all low pass filter transfer functions. This means that the larger the bandwidth, the more phase noise contributions are generated by all PLL components other than the VCO. Therefore, there is a trade-off in choice of the PLL bandwidth. Larger bandwidth leads to more phase noise suppression generated by the VCO, but allows more phase noise generated by the other PLL components. For a properly designed PLL, the close-in phase noise is dominated by the charge pump.

2.4 Fractional-N Phase-Locked Loop

2.4.1 Basic Operation

Fractional-N PLLs [34] offer a cost effective method of achieving low phase noise and fast lock times. The frequency divider in fractional-N PLLs toggles between one of two division ratios to generate an average division ratio, N.f, that lies between these two division ratios, where N is the integer division ratio and f is the fractional division ratio. The division ratios are usually chosen such that they differ by a factor of one. For example, if a division ratio of 4 is chosen in four clock cycles and a division ratio of 5 is chosen in one clock cycle, an average division ratio of (4*4+5*1)/5=4.2 is achieved. A diagram of a fractional-N PLL is shown in Fig. 2.7.

Fractional-N PLLs have several advantages. Since fractions of the reference frequency can be achieved, the reference frequency can exceed the channel spacing. This enables a reduction of the size of the frequency divider, N. This helps in both reducing the close-in phase noise as well as reduce the lock time. The major disadvantage of fractional-N PLLs is that it produces fractional spurs, which are difficult to attenuate, even with a narrow loop bandwidth.

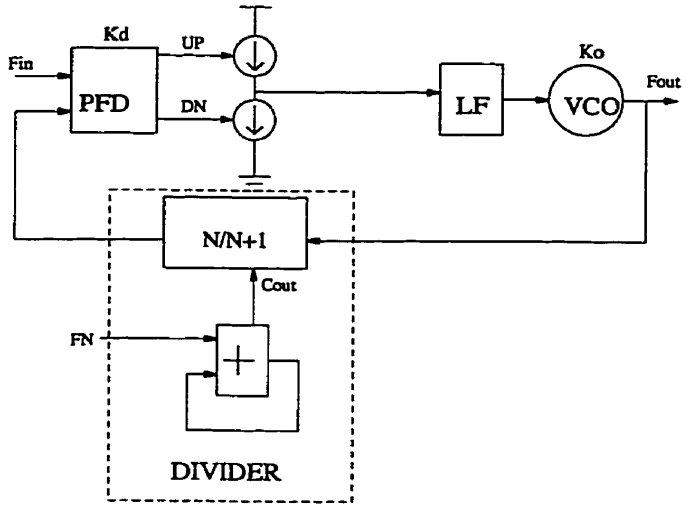


Figure 2.7. Fractional-N Phase-locked Loop

2.4.2 Analog Compensated Fractional-N PLL

One common solution to eliminate spurs generated by fractional-N PLLs is referred to as *analog compensation* [35]. A block diagram of a fractional-N PLL using this technique is shown in Fig. 2.8. This technique is based on the fact that the contents of the accumulator used in the fractional-N divider is proportional to the phase error at the output of the VCO. Therefore, the phase error can be eliminated by converting the digital contents of the accumulator into an analog signal and subtracting it from the control voltage, V_c. This is accomplished by using a current-type digital-to-analog converter (DAC). A current-type DAC is nothing but a bank of current sources which are switched on and off by digital control signals. In this case, the digital control signals come from the accumulator output bits. The size of the DAC currents with respect to the charge pump currents depends on the fractionality used. If a fractionality of eight is used, 3 bits are taken from the accumulator, and the DAC current sources are sized to by one-eighth of the charge pump currents. The larger the fractionality used, the higher the reference frequency becomes, which reduces the PLL's lock time.

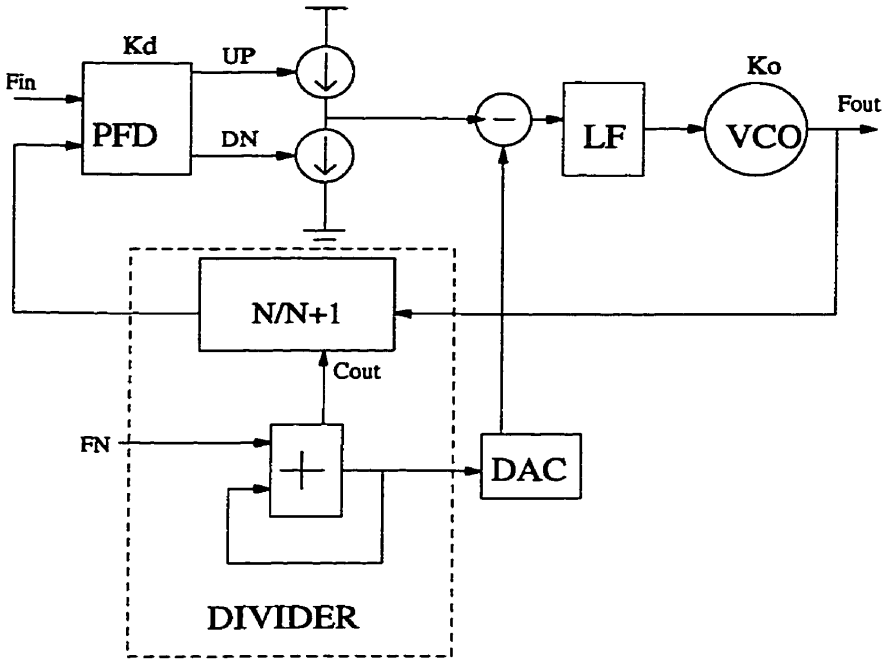


Figure 2.8. Analog compensated fractional-N PLL

Unfortunately, fractionality beyond 16 is hardly used since it is difficult to match the DAC current sources; furthermore, the DAC current sources start to occupy a significant portion of the overall PLL chip area. DAC nonlinearity errors are not an issue in this application since the DAC size is very small. Other limitations of this technique includes in the inability of the analog integrated circuits to accurately produce the error signal to be subtracted from the control voltage. This will be explained in more detail in chapter 3. Most commercial products implementing this technique report a spur reduction to nearly –70dBc [4,5,15]. The major advantage of this architecture is that it is a low-cost and low-power solution to frequency synthesis with reasonable frequency accuracy.

2.4.3 Sigma-Delta Compensated Fractional-N PLL

Another method, which has recently gained popularity, is known as *sigma-delta compensated fractional-N PLL*, or simply ΔΣ PLL [36]. Instead of attempting to

eliminate the phase errors in fractional-N PLLs, this technique noise shapes the phase errors to higher frequencies. This is done by using the output bits of a digital ΔΣ modulator to change the frequency division ratio in a random manner. The closed loop bandwidth then acts as a low-pass filter, which eliminates the noise shaped error terms. Fig. 2.9 shows a block diagram of a ΔΣ PLL. The major advantages of this technique include larger loop bandwidths (which result in faster lock times), less phase noise due to noise in the PFD and charge pumps. The two major disadvantages of this technique are increased circuit complexity (which entails larger power dissipation and area) and high phase noise at higher frequency offsets.

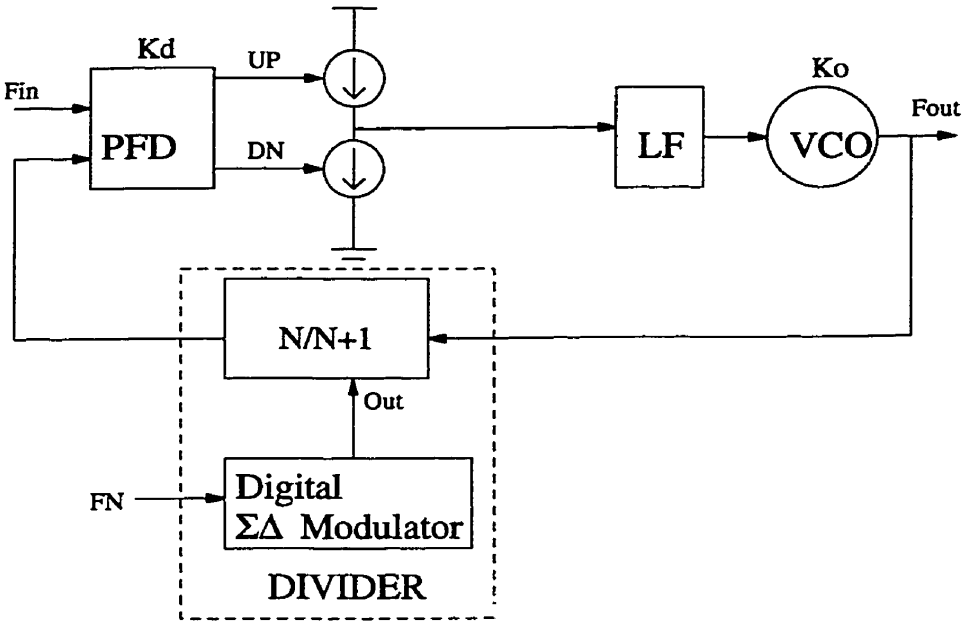


Figure 2.9. ΣΔ PLL Architecture

2.5 Fast Lock Phase-Locked Loop Techniques

The four key performance metrics in designing a frequency synthesizer for wireless systems are power dissipation, cost, spectral purity, and lock time. In this section,

techniques for reducing lock time in PLL for wireless systems are reviewed. Fast lock techniques used for PLLs can be divided into five categories:

- DSP/digital methods,
- Dual-loop/dual-PLL architectures,
- Feed-forward compensation methods,
- Variable loop bandwidth methods, and
- Frequency-to-voltage conversion.

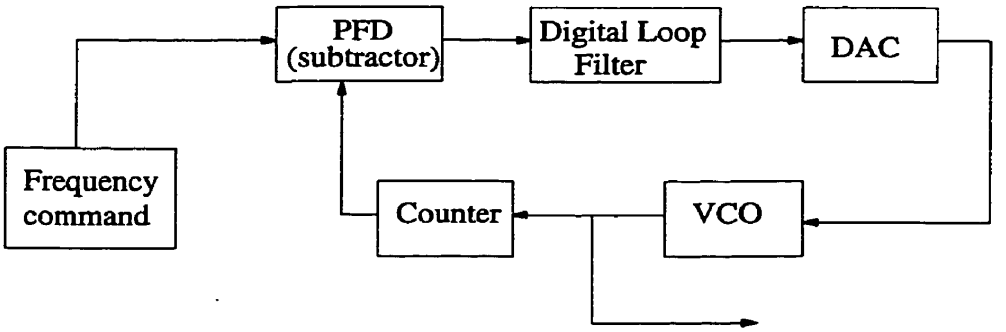


Figure 2.10. DSP PLL Architecture

In the first method[6], digital signal processing techniques are used to control the VCO control voltage, as shown in Fig. 2.10. A digital word is used to represent the reference frequency and a digital word representing the divided frequency from the VCO are subtracted from each other to produce an error term. This subtractor replaces the PFD. The error term is then fed into a digital loop filter. The output of the loop filter is fed into a digital-to-analog converter (DAC). In this implementation, there are no reference spurs produced, which means that very large loop bandwidths can be produced. Large loop bandwidth translates to fast lock time. Three main limitations exist in this architecture. First the size of the DAC may be large. It was estimated that a 27-bit DAC was required to achieve the required accuracy for the GSM and GPRS wireless standards. Secondly, spurs are generated due to finite word length truncation. Furthermore, overflow in the subtractor (used in place of the PFD) or the loop filter causes additional spurs. The third main disadvantage of this architecture is that the

complexity DSP portion is quite large, which negatively affects the power consumption and area of the frequency synthesizer.

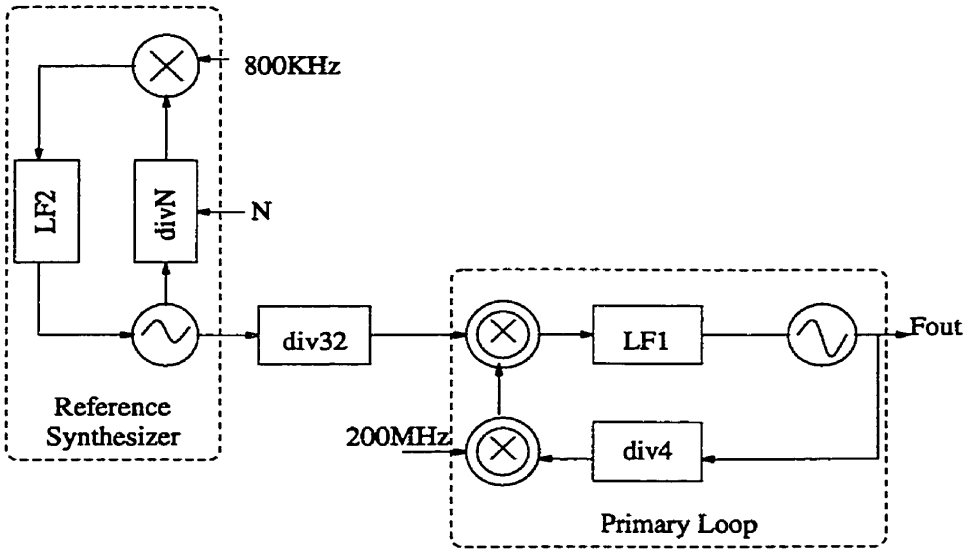


Figure 2.11. Dual loop PLL architecture

Dual-loop PLL architectures have also been reported in literature[7], which are capable of fast frequency lock times. A block diagram of the dual-loop PLL is shown in Fig. 2.11. The first loop is used for channel selection. It produces an RF output signal and is then frequency divided to achieve good phase noise. This signal is fed to the second loop, which contains a fixed divider (div 4), and an image rejection mixer with a fixed input reference frequency of 200MHz. The advantage of this approach is that there is no need for fractional-N divider (less spurs). The second PLL has a very high bandwidth and the close-in VCO noise is well suppressed. High bandwidth also translates to fast lock time. The major disadvantages of this architecture include integration difficulty (2 PLLs + mixer), large power consumption and area, and nonlinearities in the image rejection mixer may produce large spurs.

A dual-PLL architecture [8] may also be used to reduce lock time. A dual-PLL architecture is shown in Fig. 2.12. This architecture has been shown to have a 1μs lock

time when used for frequency hopping applications. The two PLLs work in parallel, but only one is active. While one PLL is producing the desired output frequency, the other is locking to a new frequency. A switch is used to select between the two. The lock time is set by the speed of the switch. The two disadvantages of this architecture are integration difficulties (2 separate PLLs), and the switch may produce nonlinear distortion when switching from one PLL to the other.

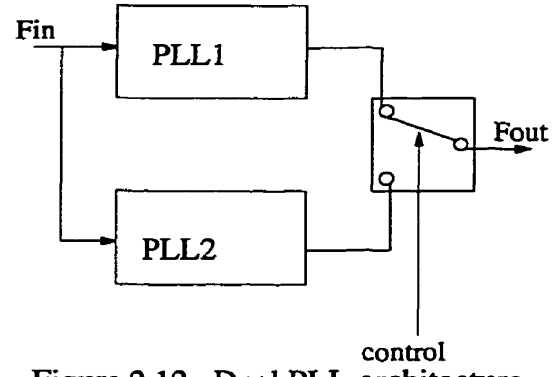


Figure 2.12. Dual PLL architecture

In a feed-forward compensated PLL [9], shown in Fig. 2.13, the knowledge of the desired frequency is used to pre-tune the VCO control voltage to near lock conditions. This is done by a DSP and a DAC. The DSP calculates the required control voltage and the DAC translates it to the pre-tuned control voltage. The DAC is implemented as either a standard DAC plus analog adder or a digitally controlled bank of charge pumps. Once pre-tuned, the regular PLL loop takes over to perform final lock. The lock time is limited by the accuracy of the DSP and DAC to produce the correct control voltage. Limitations of this architecture include the difficulty in implementing the DAC. For the standards such as the GSM and GPRS standards, a large DAC (>25 bit DAC) is required to cover the entire frequency range with reasonable accuracy. Another major difficulty with this architecture is that the accuracy of the feed-forward technique is undermined by the VCO inaccuracy, which significantly affects switching speed. VCO inaccuracy results from process variation (if the VCO is implemented on-chip) or from variation of VCO characteristics from one VCO chip to another.

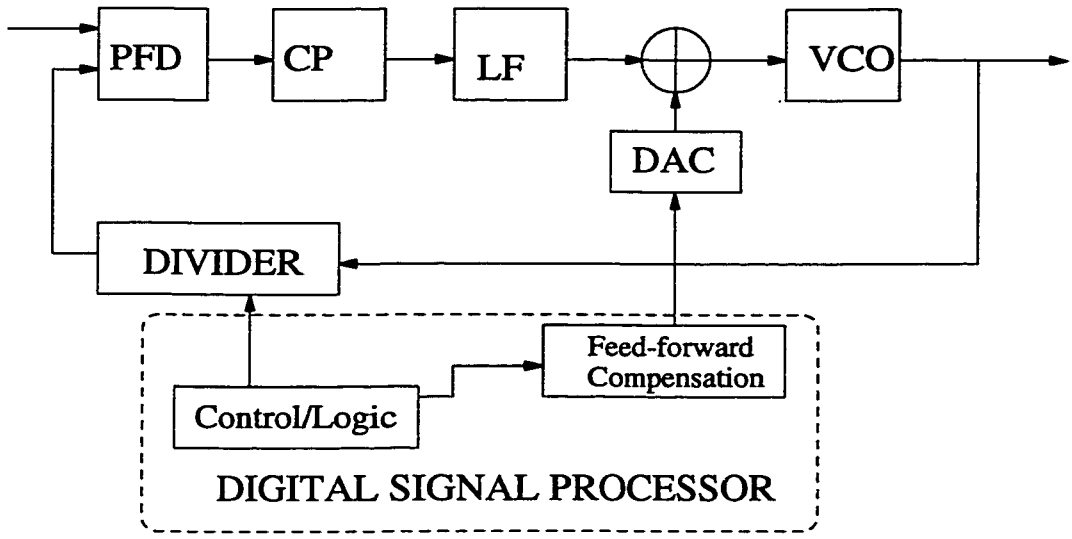


Figure 2.13. Feedforward compensated PLL architecture

Variable loop bandwidths are the most common approach used in industry to achieve fast lock. This method typically achieves 2x to 5x reduction in lock time. In [10], a DFF-based lock-detection circuit is used to detect cycle slipping. If cycle slipping is detected, the charge pump current is multiplied by a certain factor. This increases the loop bandwidth of the PLL. Once cycle slipping stops, charge pump current is returned to its original value, and the closed loop bandwidth is returned to its original value. The charge pump current during cycle slipping is limited by PLL stability requirements.

In [11], the charge pump current as well as the PLL loop filter parameters are changed, as shown in Fig. 2.14. This is known as a *two-mode PLL*. Larger loop bandwidths are achievable using this technique. One of the major limitations of this technique, however, is that switching from one bandwidth to another causes a discontinuity in the locking characteristics of the PLL. If the difference between the two bandwidths is large, this discontinuity may cause the PLL to lose lock when switching from the large bandwidth back to the narrower bandwidth. In this case, the

technique would have lost its advantage. Optimal switching from one bandwidth to another is still an active research topic.

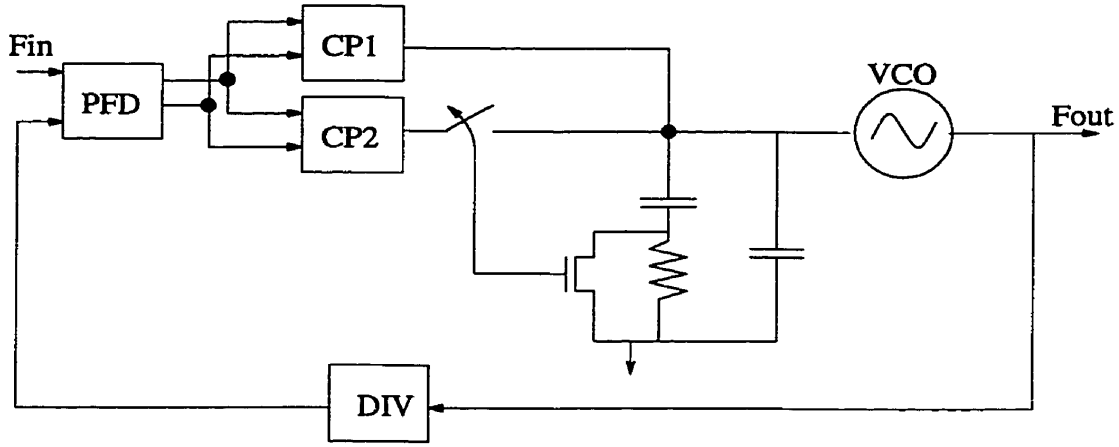


Figure 2.14. Two mode PLL architecture

One recent proposal to switching from one bandwidth to another is reported in [12]. In this approach, which is referred to as optimal gear shifting, a measure of how far the PLL is away from lock condition is used to tune the current sources in the charge pumps. This way, the bandwidth is increased or decreased to achieve optimal lock time conditions. The major limitation of this technique is that the charge pumps produce a significant amount of phase noise.

One technique that mixes digital techniques with variable loop bandwidth techniques is reported in [13]. In this implementation, shown in Fig. 2.15, a frequency difference detector (FDD) circuit is used. A FDD detects the difference in frequencies and produces a digital word representing this difference. The digital word is then fed into a DAC, which transfers this word into an analog signal which drives the loop filter directly, as shown in Fig. 2.15. First, the feedback of the main loop is inhibited, and instead the FDD-DAC-LF-VCO-FDD loop is enabled. As long as the FDD detects frequency differences from the VCO and Fin, it charges/discharges the charge on the loop filter's capacitor. This brings the PLL close to lock condition. The regular loop

filter then takes over. The major disadvantages include increased circuit complexity and only a marginal 2x improvement over a two-mode PLL.

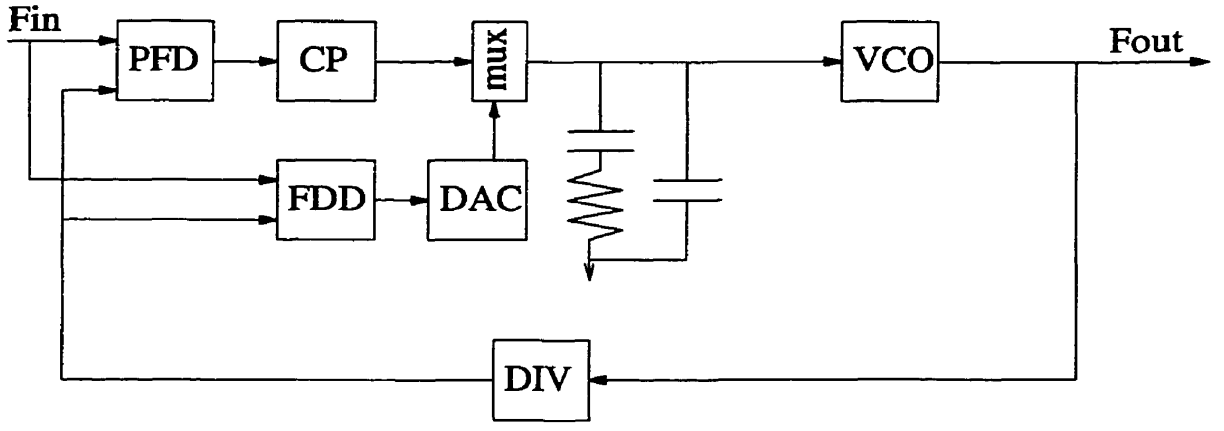


Figure 2.15. Fast lock PLL architecture employing FDD circuitry

One recent attempt to improve lock times in PLLs employs frequency-to-voltage conversion [14]. The main idea behind this approach is to convert both reference frequency and feedback frequency from the VCO (or feedback divider) to a voltage using special frequency-to-voltage converter circuitry. The two signals are subtracted in the analog domain using an operational amplifier (opamp). The signal is then fed to the VCO control voltage. A lock time of 2μs is reported for 200MHz output frequency. The major disadvantage of this approach is that it only frequency locks, no phase lock is performed; this makes modulation (indirect or direct) difficult. Another concern is that noise contribution of the opamp and frequency-to-voltage converters may be significant. Also, mismatch in the two frequency-to-voltage converters can cause an output frequency error offset.

Chapter 3

Analysis of Fractional-N PLL

3.1 Introduction

Fractional-N PLL-based frequency synthesizers have been the mainstream frequency synthesis method for nearly all wireless applications. Its merits include fast lock time and low phase noise. Its design has a large impact on the power consumption, area, and overall transceiver performance.

In this chapter, fractional-N phase-locked loops are analyzed in more detail. Closed form solutions for the prediction of the spur location and strength are developed. This is done by determining the Fourier Series coefficients of the output response of fractional-N PLLs. Both uncompensated fractional-N PLLs and analog compensated fractional-N PLLs are analyzed. The lock time performance of fractional-N PLLs are also investigated. Issues such as frequency and phase locking phenomena are analyzed.

This includes derivation of frequency lock time based on a time-domain analysis of the PLL.

3.2 Uncompensated Fractional-N PLLs

3.2.1 Time Domain Analysis

The overall PLL model for an uncompensated fractional-N PLL is shown in Fig. 3.1. The "R" signal, shown in Fig. 3.1, is the input reference signal and the "V" signal is the feedback signal from the frequency divider. The width of the R and V signals are assumed to be one reference frequency period and one prescaler period, respectively. The fractional divider is assumed to divide only by two values N and N+1. The dual modulus control signal comes from the carry-out signal of an accumulator. A timing diagram of the uncompensated PLL is shown in Fig. 3.2. Note that the "U" and "D" signals are the UP and DOWN signals which come from the phase-frequency detector (PFD) and go into the charge pump. As the figure reveals, each pulse of the "V" signal is delayed by one time step (prescaler width) with respect to the previous "V" pulse. The pulses of the "R" signal, on the other hand, are equally spaced. This creates a monotonically increasing error signal, which if uncorrected would cause the PLL to lose lock, and potentially become unstable.

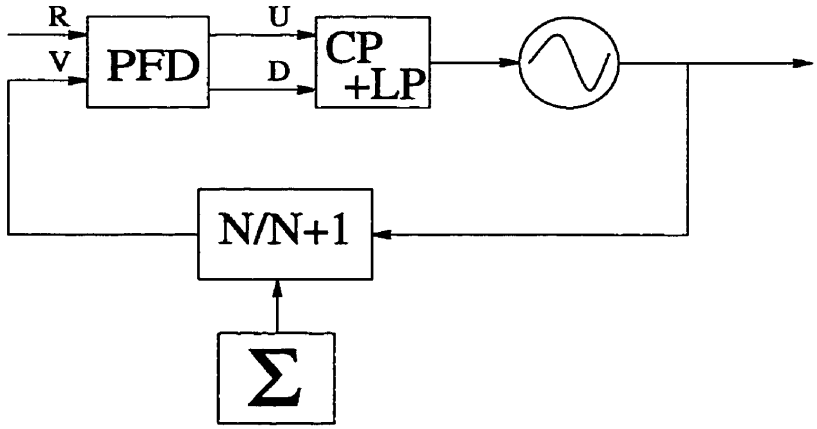


Figure 3.1. Model for uncompensated fractional-N PLL

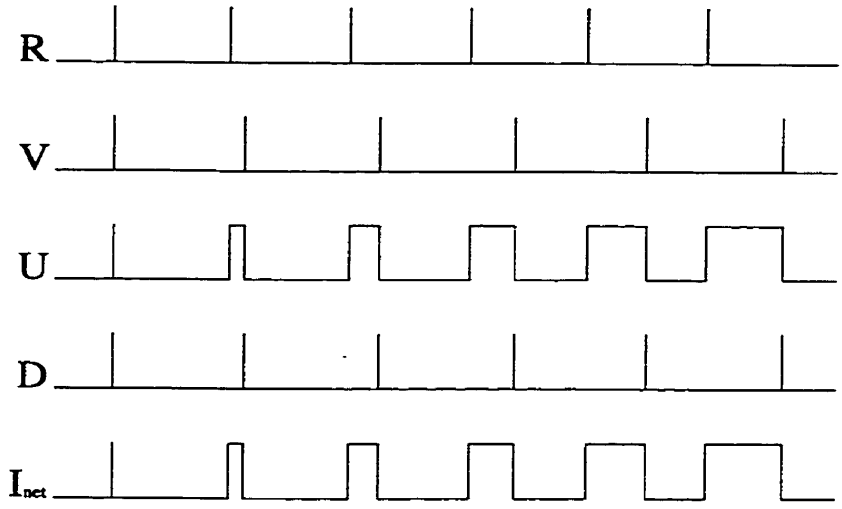


Figure 3.2. Ideal waveforms for uncompensated fractional-N PLL

Note that the total current is incrementally increased as the V signal lags further behind the R signal. This should reduce the control voltage and reduce the output frequency of the VCO. During lock condition, the control voltage should be constant over an averaged amount of time. In other words, the total average current from the charge pump should be summed to zero. The PLL accomplishes this by phase shifting the V signal, with respect to the R signal, in such a way that the total current over one fractional period is equal to zero. If the charge pump current is denoted by I_{pump}, then the following relationship must hold

$$I_{pump}\left\{\left(t_{R_0}-\left(t_{V_0}-\Delta\right)\right)+\cdots+\left(t_{R_{m-1}}-\left(t_{V_{m-1}}-\Delta\right)\right\}=0 \tag{3.1}$$

where

$$m=\frac{FD}{GCD(FD,FN)}$$

and Δ is the phase shift of the V signal, FN is the fractional numerator, and FD the fractional denominator of the desired fractional channel. Note that the V signal is periodic with period equal to $m \cdot T_{ref}$, where T_{ref} is the period of the reference signal. The fractional period, m, is maximum when the fractional numerator and denominator

are relatively prime. The t_{Ri} and t_{Vi} series are the start times of the positive edges of the pulses of the "R" and "V" signals, respectively. Equation (3.1) can be rewritten as

$$t_{R_0} + t_{R_1} + \cdots + t_{R_{m-1}} - (t_{V_0} + t_{V_1} + \cdots + t_{V_{m-1}}) + m \cdot \Delta = 0 \tag{3.2}$$

Since t_{Ri} occur at regular intervals and the spacing between two t_{Vi} signals depends on the contents of the fractional accumulator, equation (3.2) can be written as

$$\underbrace{\sum_{i=0}^{m-1} i \cdot T_{ref}}_{t_{Ri}\text{-terms}} - \underbrace{\sum_{i=0}^{m-1} (T_{ref} + T_{pre} \cdot FRD_i)}_{t_{Vi}\text{-terms}} + m \cdot \Delta = 0 \tag{3.3}$$

where FRD_i is the contents of the accumulator at time instant "i", where "i" ranges from 0 to m-1, and T_{pre} is the period of the output signal from the prescaler. It is assumed that the granularity of the fractional divider is limited to one prescaler period. Simplifying equation (3.3) further yields

$$-T_{pre} \sum_{i=0}^{m-1} \underbrace{\text{mod}(i \cdot FN, FD)}_{FRD_i\text{-term}} + m \cdot \Delta = 0 \tag{3.4}$$

Solving equation (3.4) for Δ yields

$$\Delta = \frac{1}{m} T_{pre} \sum_{i=0}^{m-1} \text{mod}(i \cdot FN, FD) \tag{3.5}$$

which is the general equation describing the phase shift of the V signal during lock condition of the uncompensated fractional-N PLL. Considering the simple case when $(m-1) \cdot FN < FD$, the "mod" operator in equation (3.5) can be dropped and the phase shift can be expressed as

$$\Delta = \frac{m-1}{2} \cdot FN \cdot T_{pre} \tag{3.6}$$

Table 3.1 below shows an example of how t_{shift} varies with the fractional ratio. A fractional denominator of 8 was fixed. Note that equation (3.6) applies for the fractional channels of $\frac{1}{8}$, $\frac{2}{8}$, and $\frac{4}{8}$.

Table 3.1. Variation of time shift with fractional ratio

Fractional Ratio	$\frac{1}{8}$	$\frac{2}{8}$	$\frac{3}{8}$	$\frac{4}{8}$	$\frac{5}{8}$	$\frac{6}{8}$	$\frac{7}{8}$
t_R-t_V	$3.5 \cdot T_{pre}$	$3 \cdot T_{pre}$	$3.5 \cdot T_{pre}$	$2 \cdot T_{pre}$	$3.5 \cdot T_{pre}$	$3 \cdot T_{pre}$	$3.5 \cdot T_{pre}$

Fig. 3.3 below shows the waveforms of an uncompensated fractional-N PLL with the phase shift phenomenon taken into account.

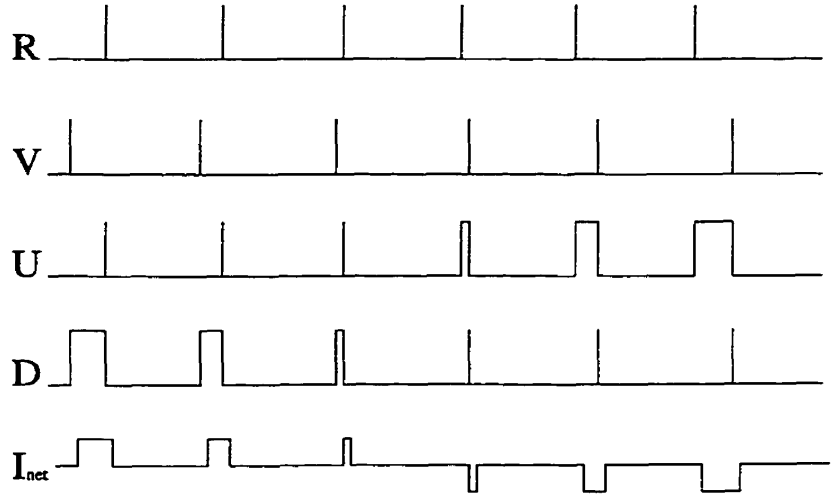


Figure 3.3. Waveforms for uncompensated fractional-N PLL with time shift

3.2.2 Frequency Domain Analysis

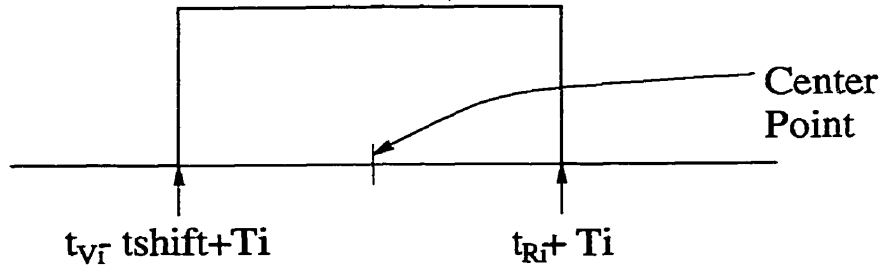


Figure 3.4. Ideal current pulse from charge pump

In order to be able to predict their spurs, the frequency response of uncompensated fractional-N PLLs must be analyzed. First, a single current pulse from the charge pump

is analyzed. An ideal rectangular pulse, as shown in Fig. 3.4, is first assumed. The center point of this rectangular pulse is given as

$$R_c = \frac{t_v + t_R - t_{shift}}{2} \tag{3.7}$$

For i=0, the center point becomes $t_R - \frac{1}{2} \cdot t_{shift}$. For general i, the center point may be expressed as

$$R_c = \tfrac{1}{2}[(t_{Vi} - t_{shift} + T \cdot i) + t_R + T \cdot i] \tag{3.8}$$

which can be simplified to

$$R_c = t_R + T \cdot i + \tfrac{1}{2}(\Delta t_i - t_{shift}) \tag{3.9}$$

where $\Delta t_i = t_R - t_{Vi} = T_{pre} \cdot \text{mod}(i \cdot FN, FD)$. From Fig. 3.4, the pulse width of the rectangular pulse can be given as

$$R_w = t_R - (t_V - t_{shift}) = \Delta t_i + t_{shift} \tag{3.10}$$

Using the following two Fourier Transform pairs [70]

$$\text{rect}(t) = \begin{cases} 1, & |t| < a \\ 0, & |t| > a \end{cases} \Leftrightarrow 2a \frac{\sin \omega a}{\omega a} \tag{3.11}$$

and

$$x(t - t_0) \Leftrightarrow e^{-j\omega t_0} X(\omega) \tag{3.12}$$

the frequency response of the rectangular pulse can be given as

$$R_i(\omega) = (-\Delta t_i + t_{shift}) \cdot \frac{\sin\left[\frac{1}{2}\omega(-\Delta t_i + t_{shift})\right]}{\frac{1}{2}\omega(-\Delta t_i + t_{shift})} e^{-j\omega t_0} \tag{3.13}$$

where

$$t_0 = t_R + T \cdot i + \tfrac{1}{2}(-\Delta t_i + t_{shift}) \tag{3.14}$$

Note that $R_i(\omega)$ can take on positive or negative values depending on whether or not Δt_i is greater or less than t_{shift}. Equation (3.13) gives the frequency response for one "m" period (where m is given by Equation (3.2)). Since the rectangular pulse of Fig. 3.4 is periodic, the correct frequency response is given by the Fourier Series, not the Fourier

Transform. The Fourier Series (F.S.) coefficients of the periodic rectangular pulse is given as

$$c_{k,i} = \frac{\Delta t_i + t_{shift}}{T_o} \cdot \frac{\sin\left[k\omega_0 \frac{1}{2}(\Delta t_i + t_{shift})\right]}{k\omega_0 \frac{1}{2}(\Delta t_i + t_{shift})} e^{-jk\omega_0 t_0} \tag{3.15}$$

where t_0 is given by equation (3.14) and $\omega_0 = \frac{2\pi}{T_0}$, and T_0 is equal to "m".

3.2.3 Model Verification

In order to verify the above model, a time-domain simulation was constructed. A Fast Fourier Transform (FFT) was then applied to this simulation in order to obtain the frequency spectrum. These simulations were done using Matlab [16].

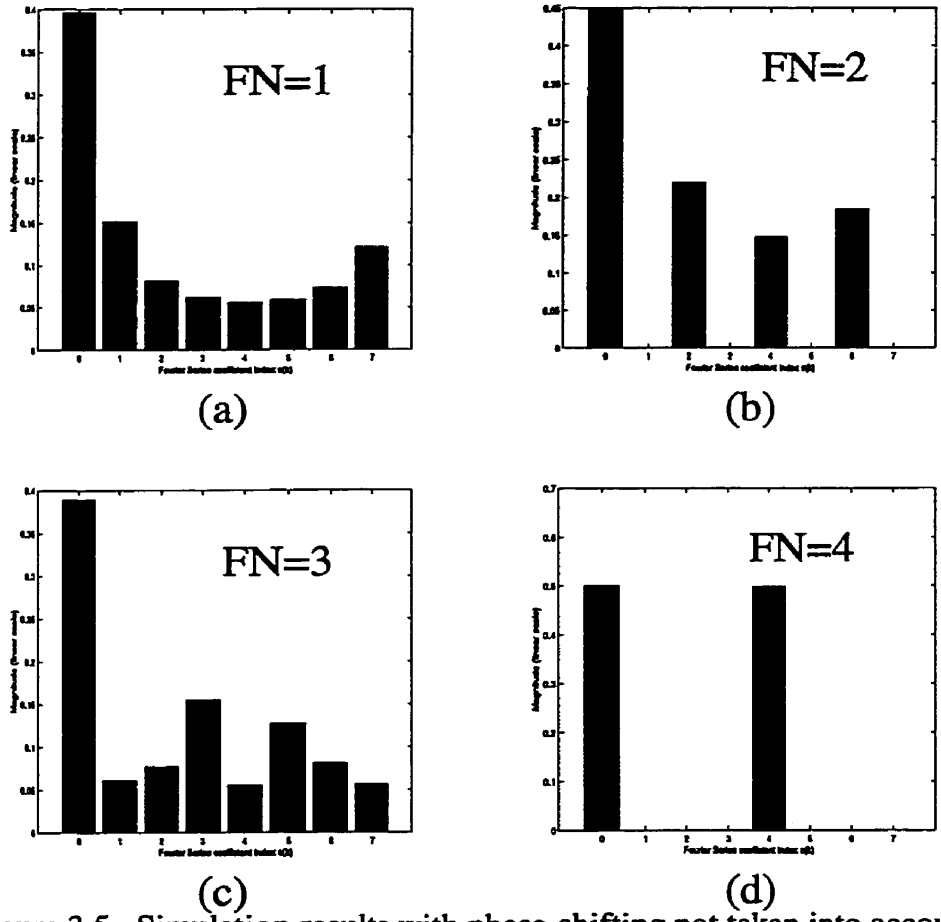


Figure 3.5. Simulation results with phase shifting not taken into account

Fig. 3.5 shows the simulation results of an uncompensated fractional-N PLL without taking into account the phase shifting phenomenon. The plot was normalized such that the sum of the coefficients is equal to one. A fractionality of 8 was assumed (i.e. FD=8). As shown, the DC term, c_0, is nonzero. This indicates that there will always be a net positive or negative charge applied to the loop filter, which would cause the VCO frequency to drift. Also, the spectrum is symmetric. This means that the spectrum of FN=1 is equal to that of FN=8, and the spectrum of FN=2 is equal to that of FN=6, and so on.

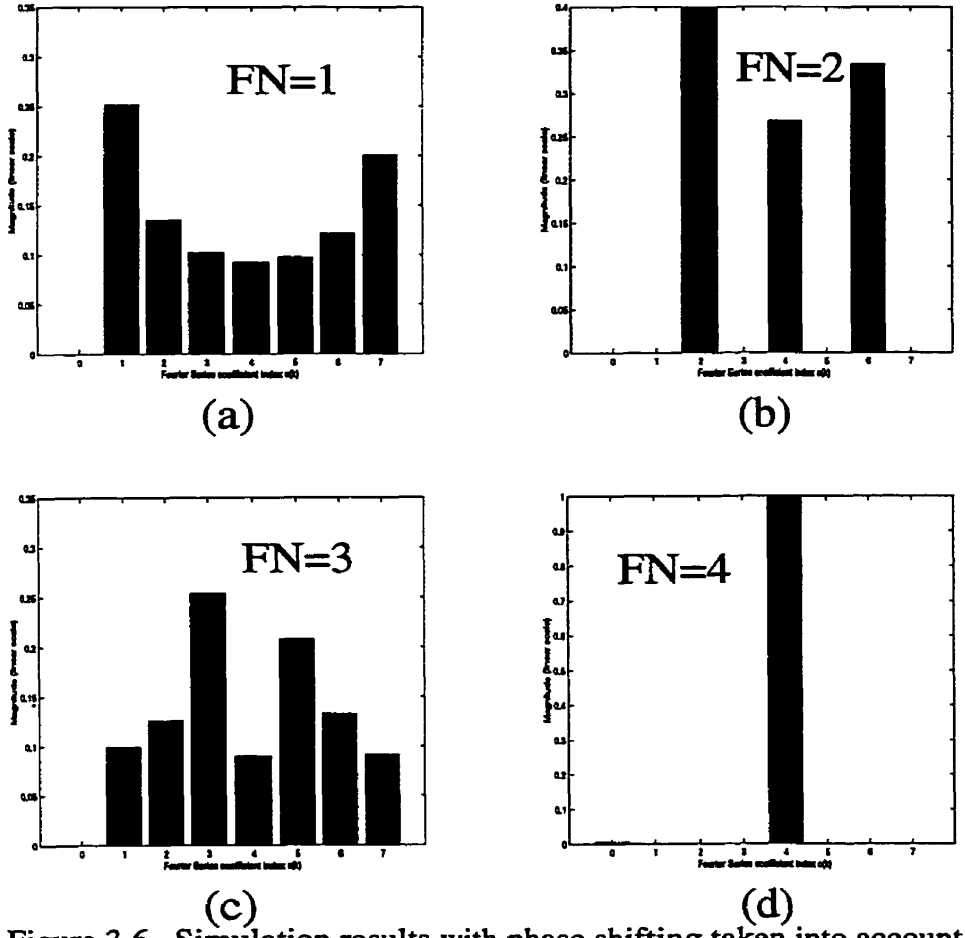


Figure 3.6. Simulation results with phase shifting taken into account

Fig. 3.6 shows the simulation results of an uncompensated fractional-N PLL with the phase shifting phenomenon taken into account, as described by equation (3.5). As with Fig. 3.5, the Fourier Series coefficients were normalized such that the sum of the coefficients is equal to one. In this case, all DC terms, c_0, are equal to zero. This verifies that the phase shift value, predicted by equation (3.5), is correct. Also note that the maximum spur level is equal to the desired fractional channel, as expected. As in the previous case, the frequency spectrum of the Fourier Series coefficients is also symmetric. Also note that the shape of the spectrum is also the same as in Fig. 3.5, except the DC term is equal to zero.

Equation (3.15) must be verified to be correct. A plot of the Fourier series must be equal to that of Fig. 3.6. In order to plot the coefficients of the Fourier Series, it is necessary to take the following summation first:

$$c_k = \sum_{i=0}^{m-1} c_{k,i} \tag{3.16}$$

Equation (3.16) was plotted and found to be identical to Fig. 3.6. This verifies that the Fourier Series coefficients predicted by equation (3.15) are correct.

3.3 Analog Compensated Fractional-N PLLs

3.3.1 Time Domain Analysis

Analog compensated fractional-N PLLs behave quite differently from their uncompensated counterparts. Fig. 3.7 displays the model of the analog fractional-N PLL used. As shown, the contents of the accumulator are fed to a digital-to-analog converter (DAC), which produces a current proportional to the value contained in the accumulator. This value is then subtracted from the loop filter. This has the effect of ideally completely eliminating the unwanted charge pump current generated by the accumulating quantization error in the V signal.

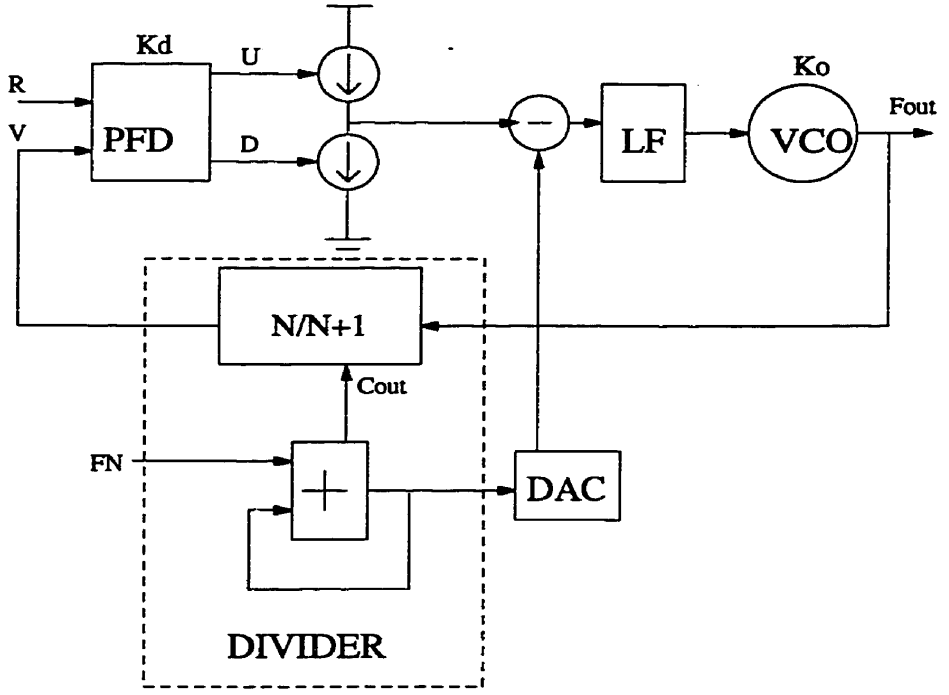


Figure 3.7. Model for analog compensated fractional-N PLL

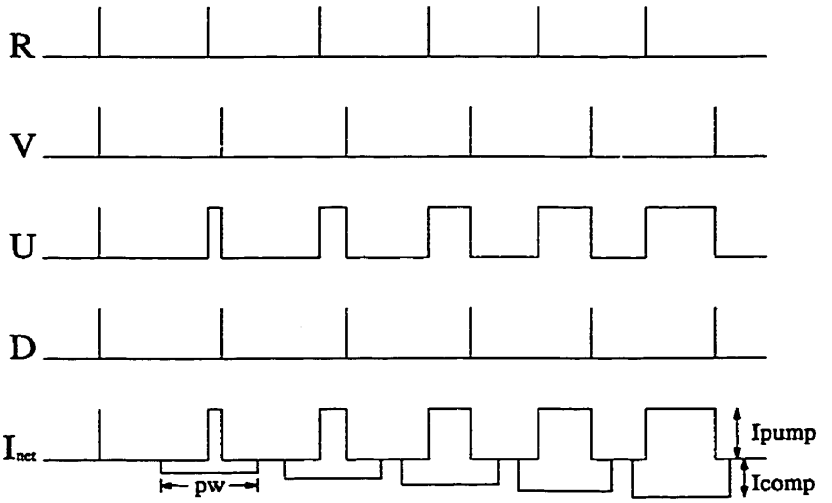


Figure 3.8. Waveforms for analog compensated fractional-N PLL

Fig. 3.8 shows the waveforms of an analog compensated fractional-N PLL. As indicated, the negative cancellation current has a larger width than the fractional error

current. This is mainly due to the difficulty in constructing electronic circuitry which can accurately produce the same narrow pulse width as the fractional error current. Instead, a wide pulse width correction signal is constructed and its amplitude is adjusted such that the area of the correction pulse is equal to the fractional error pulse. Since the net current over one fractional period is ideally equal to zero, there is no phase shifting as in the uncompensated fractional-N case.

3.3.2 Frequency Domain Analysis

In order to compute the F.S. coefficients of the analog compensated fractional-N PLL, the waveforms shown in Fig. 3.8 are used. First note that the fractional error current pulses have a width equal to Δt_i, as before. The F.S. coefficients of the fractional error pulses then become

$$c'_{k,i} = I_{pump}\frac{\Delta t_i}{T_0}\cdot\frac{\sin(k\omega_0\Delta t_i/2)}{k\omega_0\Delta t_i/2}e^{-j\omega_0\left[\frac{1}{2}T+(T\cdot i-\Delta t_i/2)\right]k} \tag{3.12}$$

The compensation current pulses have the same center points as the fractional error pulses, however, their widths and heights are different. Assuming a fixed pulse width of "pw", the compensation current becomes

$$I_{comp} = \frac{\Delta t_i \cdot I_{pump}}{pw} \tag{3.13}$$

This ensures that the total charge delivered by the analog compensation circuitry is equal to the charge delivered by the fractional error pulse. Using this value, the F.S. coefficients of the compensated current pulses become

$$c''_{k,i} = -I_{comp}\frac{pw}{T_0}\cdot\frac{\sin(k\omega_0\cdot pw/2)}{k\omega_0\cdot pw/2}\cdot e^{-j\omega_0\left[\frac{1}{2}T+(T\cdot i-\Delta t_i/2)\right]k} \tag{3.14}$$

Therefore, the F.S. coefficients for the analog compensated fractional-N current pulses are

$$c_{k,i} = \frac{1}{T_0}\left[I_{pump} \cdot \Delta t_i \frac{\sin(k\omega_0 \cdot \Delta t_i / 2)}{k\omega_0 \cdot \Delta t_i / 2} - I_{comp} \cdot pw \cdot \frac{\sin(k\omega_0 \cdot pw/2)}{k\omega_0 \cdot pw/2} \right] \cdot e^{-j\omega_0\left[\frac{1}{2}T + (T \cdot i - \Delta t_i/2)\right]k} \tag{3.15}$$

where $\omega_0 = \frac{2\pi}{T_0}$; and T_0 is the period which is equal to "m" in equation (3.2).

3.3.3 Model Verification

In order to verify the above model, a time-domain simulation was constructed. A Fast Fourier Transform (FFT) was then applied to this simulation in order to obtain the frequency spectrum. These simulations were done using Matlab [16].

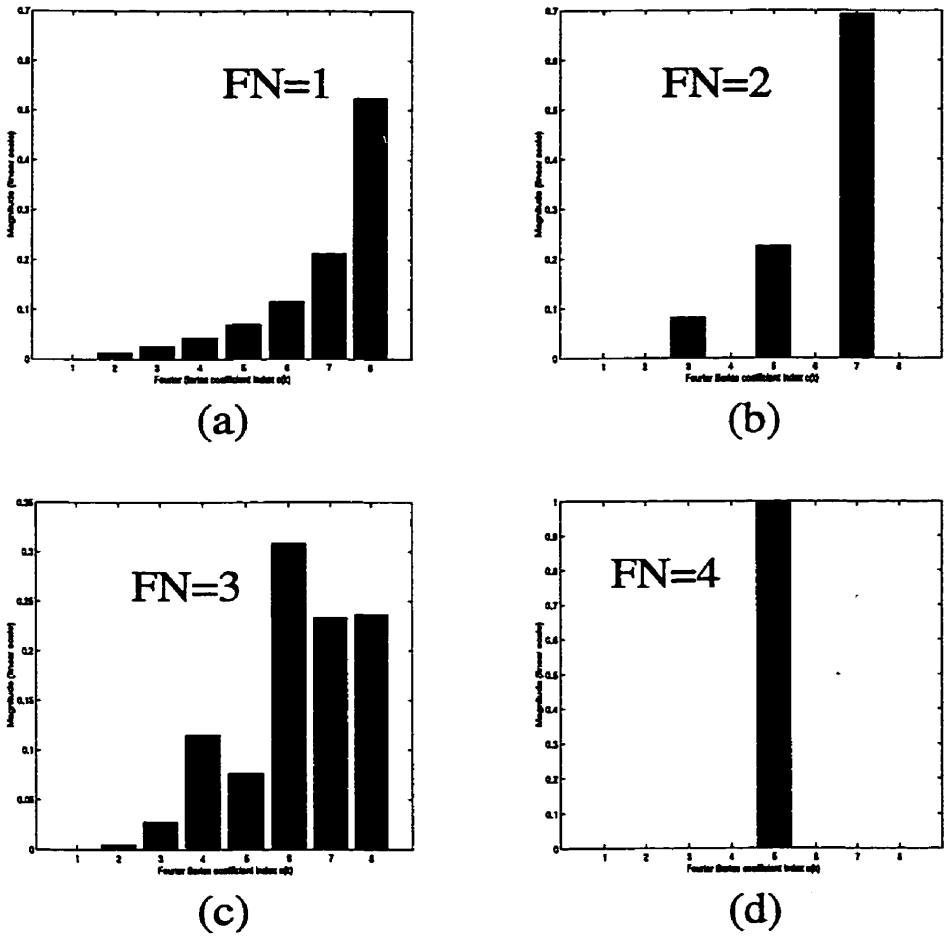


Figure 3.9. Simulation results of the analog compensated Fractional-N PLL

Fig. 3.9 shows the simulation results of an analog compensated fractional-N PLL. The plot was normalized such that the sum of the coefficients is equal to one. A fractionality of 8 was assumed (i.e. FD=8). Other parameters included a prescaler period of 2ns, compensation pulse width of 64ns, and a reference period of 256ns. As shown, the DC term, c_0, is zero. This is a direct result of the negative compensation currents used. It is interesting to show that the output spectrum at higher frequencies is nonzero. This means that even under ideal conditions, where the compensation current is exactly equal to the fractional error current, analog fractional-N PLLs do not completely cancel out fractional spurs. As before, the analytical model was consistent with the simulation plot shown in Fig. 3.9.

One interesting issue is the placement of the compensation pulse. In the analysis in this section, it was assumed that the center points of the fractional error pulse and the compensation current pulse are lined up. The compensation current pulse can also be inserted before or after the fractional error current pulse. In order to determine which is more optimal, relative F.S. coefficient magnitudes for the case of FN=1 is shown in Table 3.2 below. As shown, inserting the compensation current pulse left or right of the fractional error pulse has the same effect, since the two options are equally spaced with respect to the fractional error pulse. The table also reveals that aligning the centers of the fractional error pulse and the compensation pulse gives the best results.

Table 3.2. Effect of varying compensation pulse position on magnitudes of F.S. coefficients

k	c_k value using Center Postion	c_k value using Right Position	c_k value using Left Position
1	0dB	29.05dB	29.05dB
2	0dB	22.92dB	22.92dB
3	0dB	19.27dB	19.27dB
4	0dB	16.62dB	16.62dB
5	0dB	14.43dB	14.43dB
6	0dB	12.55dB	12.55dB
7	0dB	10.93dB	10.93dB

Another interesting experiment is to vary the width of the compensation pulse. When the width of the compensation pulse is equal to that of the fractional error pulse, the output frequency spectrum of the charge pump is flat. This means that the fractional spurs are completely eliminated. However, as discussed earlier, it is difficult to control the width of the compensation current pulse. Table 3.3 shows that varying the pulse width has a large impact on how much the fractional spurs are suppressed. The F.S. coefficients can vary by as much as 50dB depending on the width of the pulse width. As shown in the table, the narrower the compensation pulse, more of the frequency spectrum is attenuated.

Table 3.3. Effect of varying the compensation pulse width on magnitude of F.S. coefficients

k	pw=16ns	pw=32ns	pw=64ns	pw=128ns
1	0dB	11.02dB	22.41dB	52.04dB
2	0dB	16.40dB	28.08dB	38.52dB
3	0dB	18.60dB	30.29dB	41.42dB
4	0dB	19.27dB	30.84dB	36.45dB
5	0dB	18.59dB	30.03dB	40.40dB
6	0dB	16.23dB	27.50dB	37.28dB
7	0dB	10.88dB	21.70dB	30.76dB

3.4 Lock time characteristics of Fractional-N PLLs

3.4.1 General Locking characteristics

The locking characteristics of a PLL can be divided into two categories: frequency locking and phase locking. Frequency locking is a highly nonlinear mechanism and depends strongly on the type of phase detector and loop filter used. A phase-frequency detector (PFD) and a charge pump are assumed throughout this study. Fig 3.10 shows the typical locking behaviour of a charge pump based PLL. If the control voltage is

much lower than its final value, this means that the period of the "V" signal is many times wider than the "R" signal. In this case, the UP signal may be asserted for more than one "R" period. After this period, the control voltage is held constant before it can be raised (or lowered) to another value.

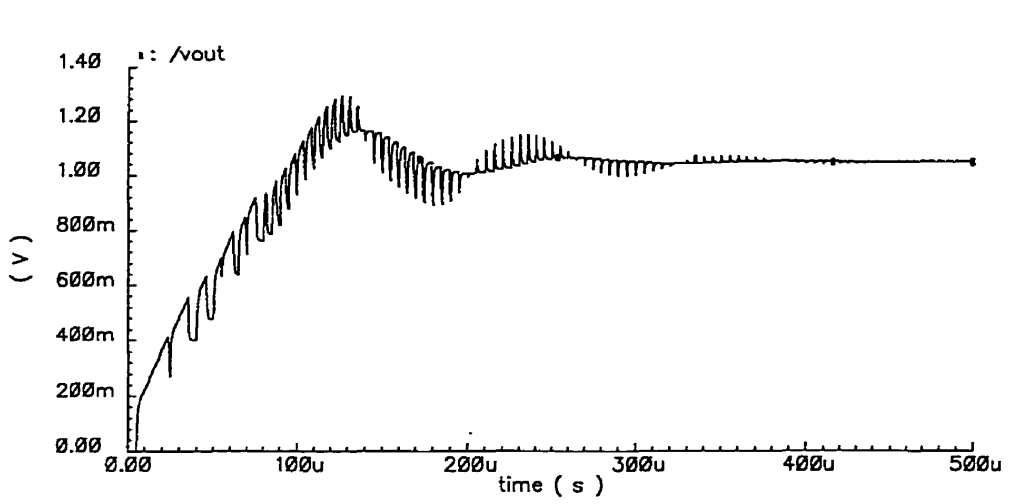


Figure 3.10. Typical locking behaviour of a charge pump based PLL

Two cases are shown in Fig. 3.11. The first case, Fig. 3.11(a), shows the best case conditions when the UP signal is monotonic and always asserted. The second case, Fig. 3.11(b), shows the case where the "V" period is larger than the "R" period, but the DN signal is asserted for a certain amount of time. This is caused by mismatches in phase between the "R" and "V" signals. It is this phenomenon, sometimes referred to as "cycle slipping", which causes the rippling in the control voltage during frequency locking, as shown in Fig. 3.10.

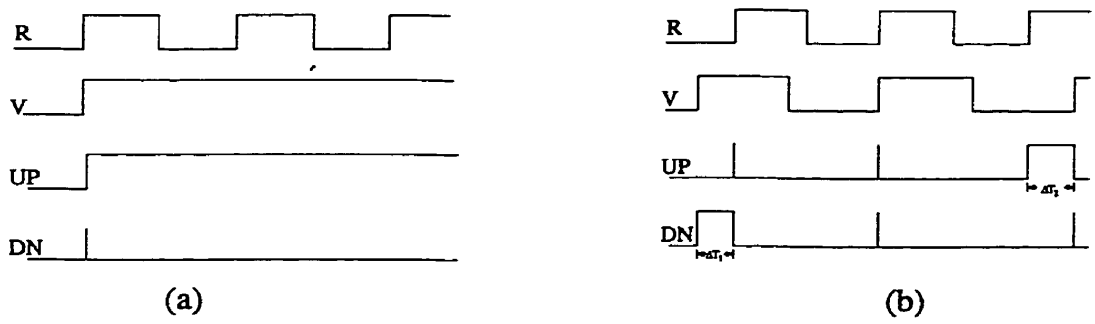


Figure 3.11. Two cases during frequency lock (a) in phase signals and (b) out of phase signals

The transition from frequency to phase lock is marked when the control voltage first reaches its final value. The control voltage continues to oscillate around its final value several times due to the fact that the phases of the R and V signals were not matched at the first crossing of the final value of the control voltage. Recall that the loop filter transfer function from PFD to the VCO output is given as

$$H_4(s) = \frac{K_o F(s)}{s + K_d K_o F(s)/N} \tag{3.16}$$

Assuming that the initial control voltage is denoted as V_o and the final value of the control voltage is denoted as V_f, the following relationship holds

$$V_f - V_o = I_p \Im^{-1}\{H_4(s)\}\Big|_0^{T_{FL1}} \tag{3.17}$$

where $\Im^{-1}\{\bullet\}$ indicates the inverse Laplace transform and T_{FL1} is the frequency lock time assuming the best case condition shownin Fig. 3.11(a).

Assuming the worst case condition shown in Fig. 3.11(b), the charge pump initially drives the VCO in the wrong direction due to the fact that the V signal leads the R signal. As shown in Fig. 3.11, the difference in the periods of the R and V signals is given as ΔT_1. After two R periods, the R signal leads the V signal and the UP signal drives the VCO in the correct direction. Note that since the last operation caused the VCO to slow down, the period of the V signal is now larger, and hence the UP pulse is

larger than the previous down pulse. This causes the VCO to be steered in the correct direction after two R periods. In this case, the difference in VCO frequency over two R periods is given as

$$\Delta f\big|_{2T_{REF}} = I_p\left(-\Im^{-1}\{H_4(s)\}\Big|_0^{\Delta T_1} + \Im^{-1}\{H_4(s)\}\Big|_0^{\Delta T_2}\right) \tag{3.18}$$

where

$$\Delta T_1 = T_V - T_R \tag{3.19}$$

and

$$\Delta T_2 = \frac{1}{\frac{1}{T_V} - I_p \Im^{-1}\{H_4(s)\}\Big|_0^{\Delta T_1}} - T_R \tag{3.20}$$

Recall, from chapter 2, that this pull-in process continues until the PLL reaches the lock-in range (defined by equation (2.16)). This means that the total change in frequency before lock-in range can be given as

$$\Delta f_{tot} = \sum_{i=1}^{n} (-1)^i I_p \Im^{-1}\{H_4(s)\}\Big|_0^{\Delta T_i} \tag{3.21}$$

where i is summed until the PLL is in lock-in range. Note that for each summation iteration i, a time period of $2T_{ref}$ elapses. Therefore, the frequency lock time can be given as

$$T_{FL2} = 2T_{ref}\, n \tag{3.22}$$

where n is the number of charge pump samples required to reach the lock-in range and T_{ref} is the period of the R signal. The total frequency lock in time is bounded by T_{FL2} (given by equation (3.22)) and T_{FL1} (given by equation (3.17)).

3.4.2 Phase Locking Characteristics

Phase locking is a better understood phenomenon. During phase locking, the PLL transient response behaves as a linear system. Since the transfer function of the PLL

linear model is known, its step response should give the correct phase locking characteristics. The phase lock time can be approximated as [17]

$$T_{PL} = \frac{2\pi}{\omega_n} \tag{3.23}$$

If a first order loop filter is considered, then the phase lock time can also be expressed as

$$T_{PL} = 2\pi\sqrt{\frac{2\pi NC}{K_o I_p}} = \frac{\pi RC}{\zeta} \tag{3.24}$$

where ω_n is the natural frequency and ζ is the damping factor.

An important issue is how much phase lock time is required if the control voltage is near its final value; in other words, how much phase lock time is required if the phase error between the R and V signals is small? In order to address this issue, several simulations were run in order to characterize the lock time behaviour of charge-pump based PLLs. In all these simulations the PLL parameters listed in Table 3.4 were used.

Table 3.4. PLL parameters used for phase lock characterization

Parameter	**Value**
F_{ref}	13MHz
N	75
R	1KΩ
C	10nF
C_2	1nF
K_v	40MHz/V

Fig 3.12 shows the variation of lock time and peak voltage excursion with phase error. In this experiment, the control voltage is assumed to have an initial value equal to its final value. The phase error is defined as "R-V". This means that a negative phase error implies that the reference signal, R, leads the feedback signal, V. During phase locking, the control voltage oscillates around the final value of the control voltage, with the amplitude of each period being smaller than the previous oscillation period. This is

because the system used is an underdamped system. The peak voltage excursion is defined as the maximum amplitude between two crossings of the final control voltage (labeled "zero crossing" in Fig. 3.12). The figure reveals three interesting facts. First, the zero crossings occur at regular time intervals. This time interval is directly related to the PLL parameters. For smaller values of the damping coefficient, the time between zero crossings shortens. Another interesting phenomenon is that the peak amplitude excursions decrease exponentially with each zero crossing interval. In the case shown in Fig. 3.12, the peak amplitude excursions decrease by a factor of five after every zero crossing. Again, this factor is directly related to the loop parameters. For larger values of the damping coefficient, this factor increases.

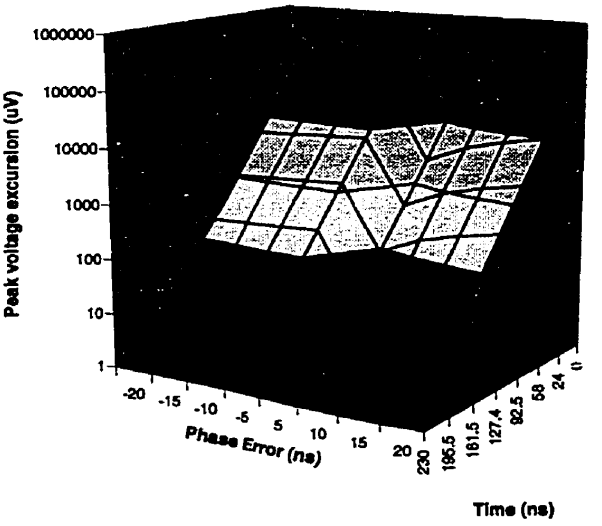


Figure 3.12. Variation of lock time and peak voltage excursion with phase error

The third interesting phenomenon is the variation of the lock time with the polarity of the phase error. As shown in Fig. 3.12, positive phase errors lead to substantially faster lock time that negative phase errors. The reason for this is that negative phase errors are really positive phase errors with a phase error equal to $2\pi - \theta_{error}$, or in the time domain, the phase error can be expressed as $T_{ref} - t_{error}$. For phase errors less than half a period, the negative phase error is really a large positive phase error.

One important conclusion from the above experiment is that even if the frequencies are initially aligned, a large amount of time might be needed for phase locking. This is true even for small phase errors. Fig. 3.13 shows the variation of frequency error with time for different phase error offsets. As shown the lock time does not vary a great deal for phase errors between 5ns up to 20ns.

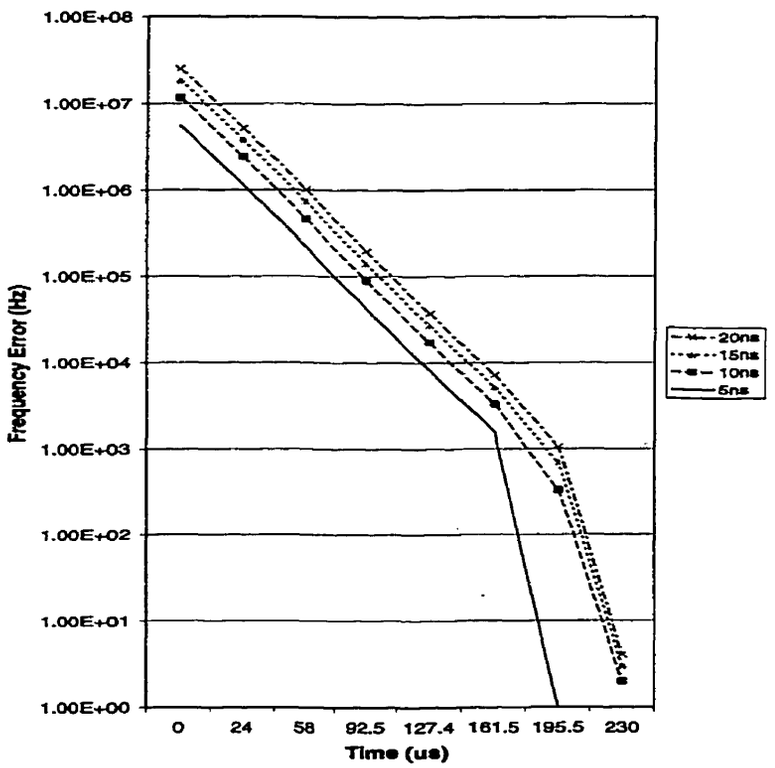


Figure 3.13. Variation of frequency error with time for different phase error offsets

Fig. 3.14 shows the results of another experiment. In this experiment both the phase error and the initial control voltage are varied simultaneously and the lock time is measured. Locked condition is defined as when the frequency variation is less than 50Hz (to meet GSM and GPRS applications). As the figure reveals there are two local minima in the graph. One occurs when the phase error is 3ns and the control voltage error is 5mV, and the other occurs when the phase error is 5ns and the control voltage is 10mV. This seems to indicate that there is an optimum trajectory in which the lock time is minimized. This is achieved by varying both the phase error and initial frequency error (or control voltage).

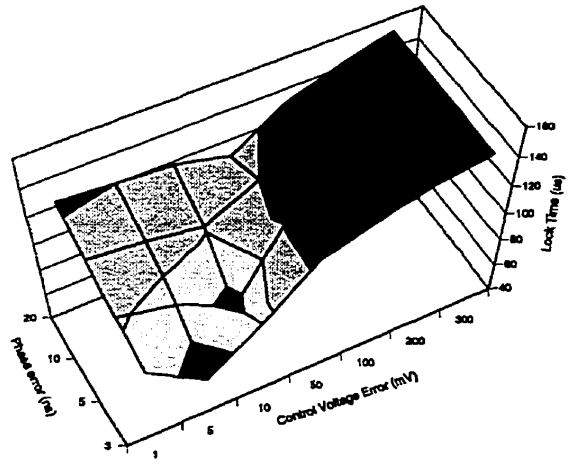


Figure 3.14. Variation of lock time with phase error and initial frequency error

In order to verify this, another simulation was conducted in which the initial output frequency was set to 200KHz away from the targeted frequency. The phase lock was

then measured at different phase error offsets. Fig. 3.15 shows the result of this experiment. As shown there is a local minimum when the phase error is 2.7ns. At this phase error, the lock time is 38.4μs. This constitutes more than a 2x reduction in phase lock time as compared to a 2ns phase error.

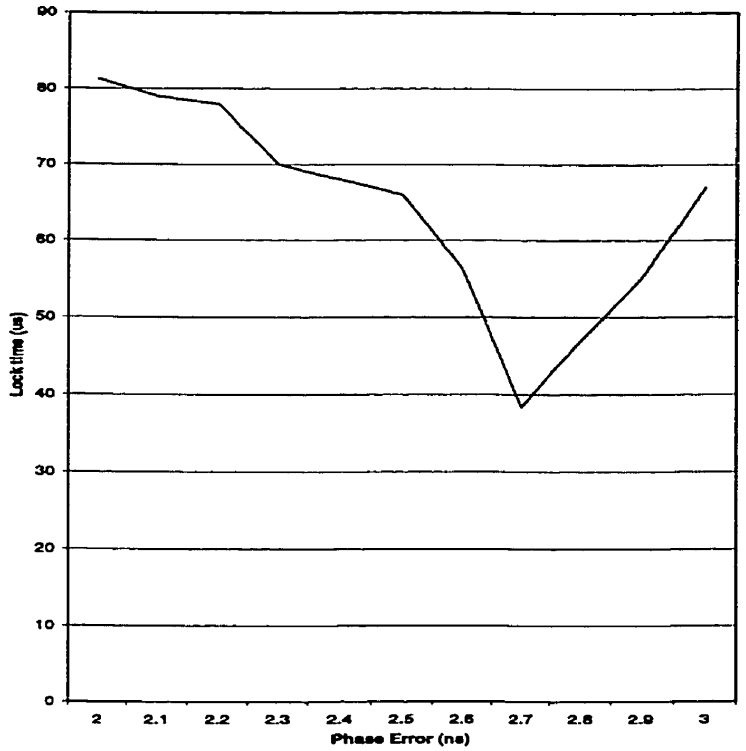


Figure 3.15. Variation of lock time with phase error for initial frequency error of 200KHz

In chapter 6, a fast phase lock technique is described which takes advantage of this phenomenon.

3.5 Conclusions

Two important PLL parameters for wireless systems are spurious performance and lock time. Both of these parameters have been analyzed in detail. Closed form expressions predicting the spur location and strength have been developed using Fourier Series

analysis. These expressions have been developed for both the uncompensated fractional-N PLL as well as the analog compensated fractional-N PLL. The effects of varying the compensation pulse parameters, for the case of the analog compensated fractional-N PLL, have been investigated. The closed form expressions were then validated through simulations. The lock time performance of PLLs has also been analyzed in detail. Previously, only linear models for PLLs have been used in order to predict the locking behaviour of PLLs. In this study, a nonlinear time-domain based approach has been adopted in order to better understand the frequency locking phenomenon in PLLs. Bounds on expressions of frequency lock time have been developed. Phase locking phenomena have also been investigated. It has been determined that there seems to be an optimal trajectory in which the phase lock time of the PLL is minimized.

Chapter 4

A MASH Sigma-Delta Fractional-N PLL

4.1 Introduction

As discussed in chapter 2, digital sigma-delta modulation techniques can be applied to fractional-N PLLs in order to digitally compensate for fractional spurs. The advantage of this technique over analog compensation is that it avoids the maximum fractional spur reduction limitation due to device mismatching. Furthermore, arbitrarily small resolution may be obtained by sigma-delta modulation techniques.

In this chapter, an implementation of a MASH sigma-delta PLL is detailed. The GPRS wireless data communication standard was targeted. First, architectural

parameters are adjusted to meet the specification requirements. Then, the circuit implementation of the PLL components is detailed. Finally, performance of this design is evaluated through a test chip.

4.2 MASH Sigma-Delta PLL Architecture

As stated in chapter 2, limitations of analog compensated fractional-N PLLs include limited spur reduction and limited resolution. These two factors lead to limitations in lock time and spectral purity performance of the synthesizer. One alternative to this architecture is to provide a digital solution to eliminating spurs found in fractional-N PLLs. A digital sigma-delta modulator may be used to accomplish this. In this section, parameters such as oversampling ratio, modulator order, and bitwidth are investigated.

4.2.1 Digital MASH Sigma-Delta Modulator

Sigma-delta modulators have been studied extensively in the context of oversampled A/D converters with a 1-bit quantizer [24]. This type of data converter produces large quantization noise. This quantization noise is filtered out by the noise shaping characteristics of the ΣΔ modulators. ΣΔ modulators are used to shift the quantization noise to a high frequency while still preserving the spectrum of the input signal. The high frequency noise would then be filtered by a subsequent low pass filter. A similar technique may be used in fractional-N PLL frequency synthesizers, since the closed loop bandwidth acts as a low pass filter [25,69]. The quantization noise in fractional-N PLLs would be the frequency spurs due to the periodic frequency division ratio switching. Hence, ΣΔ modulators transfer the correlated frequency spurs into high frequency random noise. A block level diagram of a digital version of the ΣΔ modulator is shown in Fig. 4.1. Note that this can be implemented as a digital accumulator with the output being the carry-out signal.

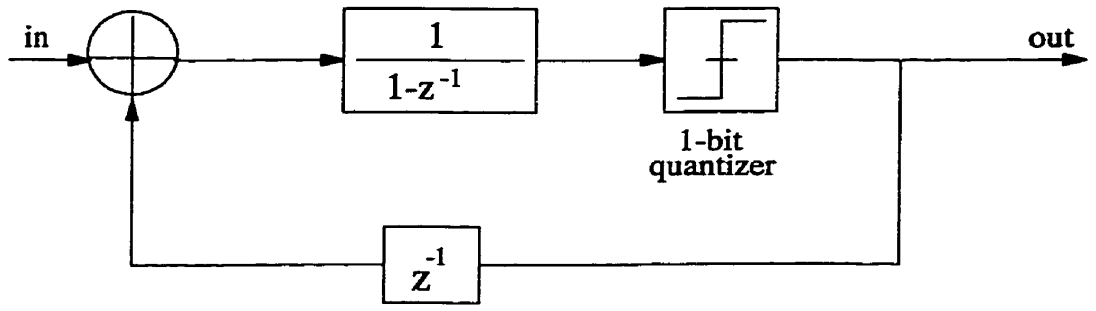


Figure 4.1. Digital Sigma-Delta Modulator

The frequency shaping property of ΣΔ modulators is best understood in terms of its transfer function. A diagram of a conventional first order ΣΔ modulator is shown in Fig. 4.2. Note that the quantization noise is modeled as additive noise. This model is only justified if the input to the modulator has sufficient activity [18]. Under this assumption, the quantization noise will be uniformly distributed with σ^2=1/12. The transfer function of this feedback loop is

$$Y(z) = \frac{1/(1-z^{-1})}{1+z^{-1}/(1-z^{-1})}G(z) + \frac{E_q(z)}{1+z^{-1}/(1-z^{-1})} \tag{4.1}$$

which may be simplified to

$$Y(z) = G(z) + (1-z^{-1})E_q(z) \tag{4.2}$$

As Equation (4.2) indicates, the input signal passes unchanged and the quantization noise is shaped by the $(1-z^{-1})$ function, which is a high-pass filter.

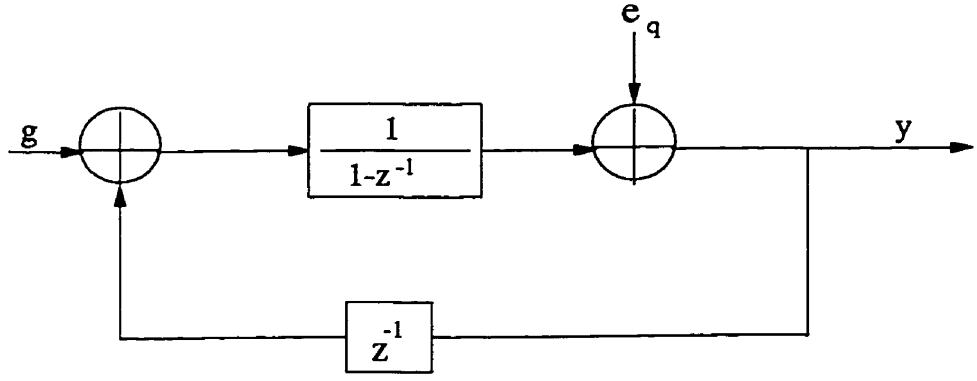


Figure 4.2. First order ΣΔ modulator model

One way of achieving even less quantization noise at higher offset frequencies is to cascade several first order ΣΔ modulators, as shown in Fig. 4.3. This type of architecture is called a MASH ΣΔ modulator [26]. Since there are three first-order ΣΔ modulators (each accumulator is a first order sigma-delta modulator), this is a third order MASH ΣΔ modulator. The sum term of the accumulator is the error term output of the sigma-delta modulator. The transfer functions of each modulator section is given as

$$Y_1(z) = G(z) + (1-z^{-1})E_{q1}(z) \tag{4.3a}$$

$$Y_2(z) = -E_{q1}(z) + (1-z^{-1})E_{q2}(z) \tag{4.3b}$$

$$Y_3(z) = -E_{q2}(z) + (1-z^{-1})E_{q3}(z) \tag{4.3c}$$

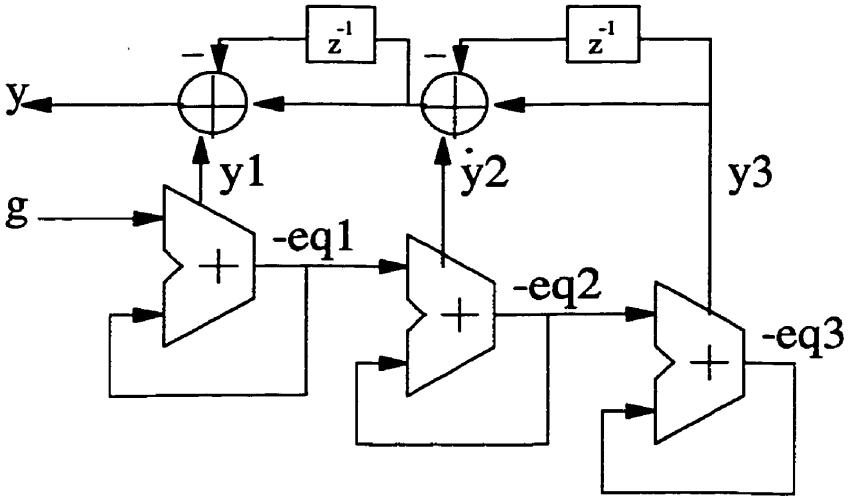


Figure 4.3. Third order MASH ΣΔ digital modulator

The modulator's total output is given as

$$Y(z) = Y_1(z) + (1-z^{-1})\left[(1-z^{-1})Y_3(z) + Y_2(z)\right] \tag{4.4}$$

which can be simplified as

$$Y(z) = G(z) + (1-z^{-1})^3 E_{q3}(z) \tag{4.5}$$

In general, for n^{th} order MASH ΣΔ modulators, the output of the modulator becomes

$$Y(z) = G(z) + (1-z^{-1})^n E_{qn}(z) \tag{4.6}$$

where E_{qn} is the quantization noise of the n^{th} sigma delta modulator. In the context of fractional-N PLLs, the quantization error has a variance of $\delta^2/12$ over a bandwidth of F_{ref}, where F_{ref} is the comparison frequency at the PFD and δ is the step size. Assuming division ratio varies between N and N+1, the step size (δ) is equal to one. Consequently, the power spectral density of the quantization noise is $1/(12F_{ref})$. The power spectral density of the modulation noise (or frequency fluctuation) then becomes

$$N(f) = E_q(f)\left|(1-z^{-1})^n F_{ref}\right|^2 = \left|1-z^{-1}\right|^{2n}(F_{ref}/12) \tag{4.7}$$

where n is the order of the modulator and $E_q(f)$ is the power spectral density of the quantization noise . Since we are interested in the phase noise contribution of the ΣΔ modulator, not the frequency noise, Equation (4.7) must be integrated. Integrating in the z-domain amounts to multiplying by $\frac{1}{1-z^{-1}}$; therefore, the power spectral density of the phase fluctuations becomes

$$N_\phi(f) = \left(\frac{2\pi T_{ref}}{1-z^{-1}}\right)^2 \cdot \left(\frac{(1-z^{-1})^{2n} F_{ref}}{12}\right) = \frac{(2\pi)^2}{12F_{ref}}\left|1-z^{-1}\right|^{2(n-1)} \tag{4.8}$$

Converting this from the z-domain to the frequency domain gives

$$N_\phi(f) = \frac{(2\pi)^2}{12F_{ref}}\left[2\sin\left(\frac{\pi f}{F_{ref}}\right)\right]^{2(n-1)} \tag{4.9}$$

One important issue is to note the exponent term on the sine function in Equation (4.7). In an analog ΣΔ modulator, this term would simply be "2n". The input to a digital ΣΔ modulator, however, is a "frequency" control word. This frequency control word must first be integrated to produce a phase. In the digital domain, an integrator may be realized by an accumulator. Note, however, that an accumulator would add a pole, or factor of $1/(1-z^{-1})$ in the digital domain to the ΣΔ transfer function (Equation 4.2). This has the effect of reducing the order of the ΣΔ modulator by one.

One disadvantage of using a higher order modulator is increased hardware complexity. Fig. 4.4 shows the spectral densities of quantization noise shaped by second and third order modulators. As the figure shows, using a higher order ΣΔ modulator results in less quantization noise at lower frequencies; however, the noise rises faster and becomes more difficult to suppress at frequencies near $F_{ref}/2$.

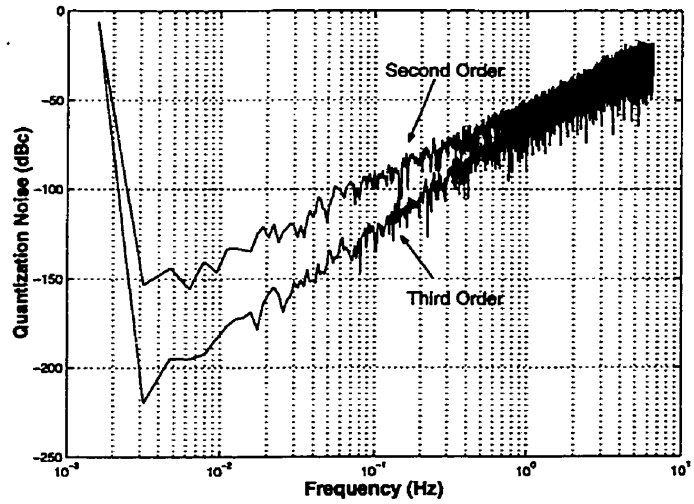


Figure 4.4. Spectral densities of second and third order modulators

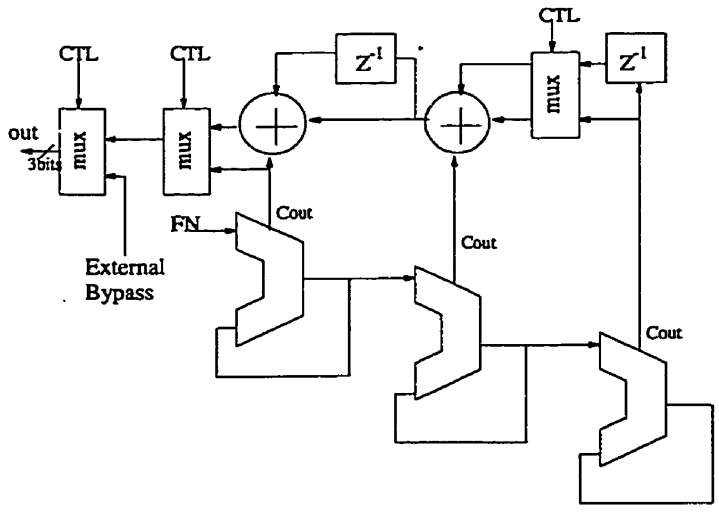


Figure 4.5. A flexible $2^{nd}/3^{rd}$ Order MASH ΣΔ digital modulator

In this work, a programmable order digital MASH ΣΔ modulator was implemented, as shown in Fig. 4.5. The "CTL" control bits determine whether a first, second, or third order ΣΔ modulator is chosen. The datapath width of the modulator was set to 24-bits to achieve high frequency resolution at high reference frequencies (around 10MHz to 50MHz).

4.2.2 Sigma-Delta Dithering

As mentioned earlier, the additive noise model of the quantizer in the ΣΔ modulator only hold true for busy signal inputs. In the case of frequency synthesizers, the input of the ΣΔ modulator is constant (since a constant output frequency is desired). When a constant input is applied to the ΣΔ modulator, a periodic signal results in the output. Assuming the input can be expressed as a fraction, the period of this signal depends on the divisibility of the numerator and denominator of the input signal and the size of the denominator (similar to the fractional period mentioned in chapter 3). A periodic signal manifests itself as a spur in the frequency domain. If the frequency of the spur is comparable or less than the closed loop bandwidth of the PLL, it will show up in the output frequency response of the PLL.

One simple method of ensuring that the numerator and denominator are mutually prime is to add a small offset to the denominator. For example, if the input can be expressed as $\frac{2}{8}$, a small seed value can be added in such a way that the fraction is $\frac{20}{81}$. This has the effect of adding activity to the LSBs of the ΣΔ modulator.

4.2.3 PLL Simulation Performance

The GSM and GPRS standards were targeted when determining the optimal parameters of the ΣΔ modulator. On one hand, maximum spur suppression is required. On the other hand, the quantization noise must be suppressed sufficiently at high frequencies in order to meet the phase noise specifications at high frequency offsets. A reference

frequency of 13MHz is specified for the GSM/GPRS standards. This sets F_{ref} for the modulator to be $2^i \cdot 13MHz$, where i is either positive or negative. For simplicity only 6.5MHz and 13MHz reference frequencies were considered. For sufficient spur suppression, it has been assumed that the closed loop PLL response is one order higher than that of the ΣΔ modulator.

The oversampling ratio (OSR), which is defined as the reference frequency divided by the closed loop bandwidth of the PLL, determines how much quantization noise is suppressed. Fig. 4.6 shows the reduction of phase noise (at 3MHz offset) generated by a 2nd and 3rd order ΣΔ modulators for different OSRs, respectively. Recall that the phase noise specification of GSM and GPRS at 3MHz offset is –141dBc/Hz. Using these two graphs it can be seen that an OSR of 280 is required for a 2nd order ΣΔ modulator, while an OSR of 340 is required for a 3rd order ΣΔ modulator.

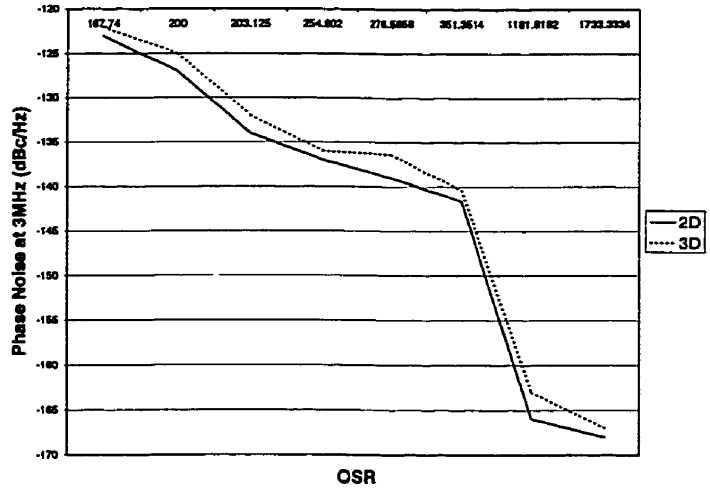


Figure 4.6. Effect of OSR on quantization noise reduction

Assuming a reference frequency of 13MHz, a closed loop bandwidth of 30KHz is required. Using this information, the necessary loop filter parameters may be computed in order to meet the lock time specification of 200μs. Table 4.1 below shows the loop component values that were necessary in order to meet this specification. Fig. 4.7

shows the lock time simulation of the ΣΔ PLL using these parameters. A two-mode fast lock technique [11] was used. The speedup current was 1.6mA and a switch was used to change the resistor value from 956Ω to 478Ω to preserve loop stability. The switch time was set to 40μs.

Table 4.1. Loop Filter component values

Loop Component	Value
R2SPD	478Ω
R2G	478Ω
C_1	1.18nF
C_2	17.58nF
I_p	200μA
K_{VCO}	50MHz/V

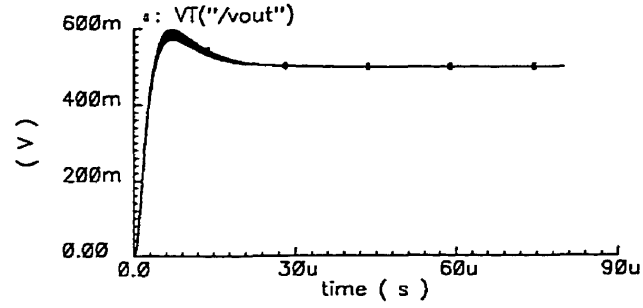


Figure 4.7. Lock time simulation of ΣΔ PLL

4.3 PLL Circuit Implementation

The PLL architecture described above has been implemented in a 0.6μm BiCMOS technology. All loop components have been included on chip, except for the VCO and loop filter. The PLL operated at from a 3.3V supply voltage. A close-in phase noise target of –80dBc/Hz was targeted. This specification is well below that of most wireless standards.

4.3.1 Frequency Divider Design

The frequency divider is the most complicated portion of a fractional-N PLL. As Fig. 4.8 shows, the divider consists of a high speed dual-modulus prescaler [22], and two programmable counters[21]. The advantage of this architecture is that only a small portion of the divider, the prescaler, needs to work at the VCO frequency. This is important for reducing power consumption.

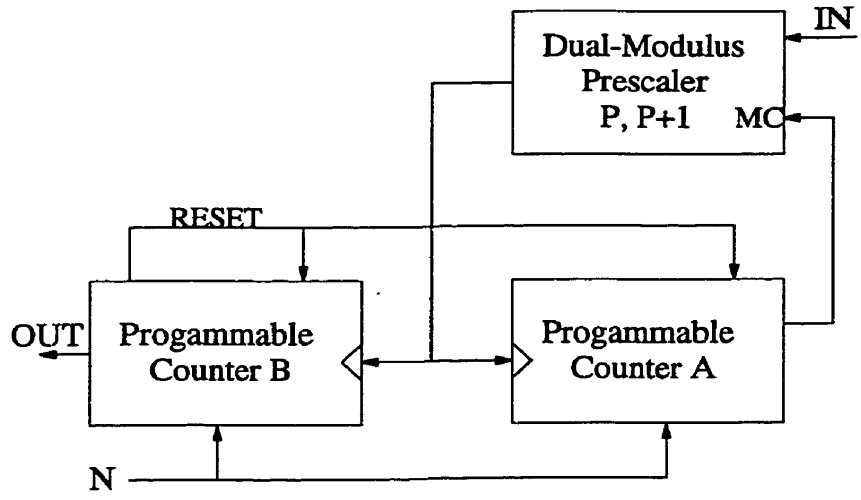


Figure 4.8. Divider architecture used

A description of the overall operation of the divider is as follows. Initially, counters A and B are loaded with their respective values. The size of counter B is assumed to be larger than that of A. The prescaler division ratio is initally set to P+1. When counter A completes, it sends its complete signal to the dual modulus prescaler to change its division ratio from P+1 to P. At this time instant, the total division ratio is A(P+1), where A is the value loaded into the A counter. The B counter continues for more cycles. When the B counter completes counting, its completion signal resets both A and B counters, and the process repeats. At this time instant, the counters would have counted an additional (B-A)P VCO cycles. Therefore, the total division ratio is

$$\begin{aligned} N &= (B-A)P + A(P+1) \\ &= BP + A \end{aligned} \tag{4.1}$$

Note that for correct frequency division, several conditions must hold. First, the value loaded into the B counter must be greater than that of the A counter. In the case of a fractional-N divider with sigma-delta modulation techniques, a negative value may be added to N. In this case, the minimum value for B becomes

$$B > A + k \tag{4.2}$$

where k is the maximum negative value produced by the sigma-delta modulator. Secondly, the value loaded into the B counter must also be greater than P and the value loaded into the A counter cannot exceed P. These two conditions are necessary in order to guarantee a contiguous division range. Note that these constraints give rise to minimum division ratio being

$$N_{\min} = P \cdot (P-1) \tag{4.3}$$

Since the dual-modulus prescaler runs at the VCO frequency, its size must be limited in order to reduce its power dissipation. Since the voltage swing of the VCO is small, the input sensitivity of the prescaler is also of concern. In order to satisfy these two constraints, the flip-flops gates used in the prescaler were designed using bipolar ECL logic with 250mV swing. An input buffer was designed to work with –20dBm input power levels. The current in the ECL gates were used to meet the required voltage swing and the phase noise specification of –90dB/Hz (referred to the output frequency). In order to enable two reference frequencies of 6.5MHz and 13MHz, as discussed in section 4.2, a dual-modulus prescaler of 8/9 and 16/17, shown in Fig. 4.9, was implemented. The SEL control signal selects between a 8/9 dual-modulus division ratio or a 16/17 dual-modulus division ratio. The MC signal selects between an even or odd division ratio. Table 4.1 shows the truth table for the division ratio of the prescaler. The top portion shown in Fig. 4.9 is a divide-by-4/5 synchronous counter. The bottom portion, called the extender, uses the divide-by-4/5 counter to produce a total division ratio of either 8/9 or 16/17.

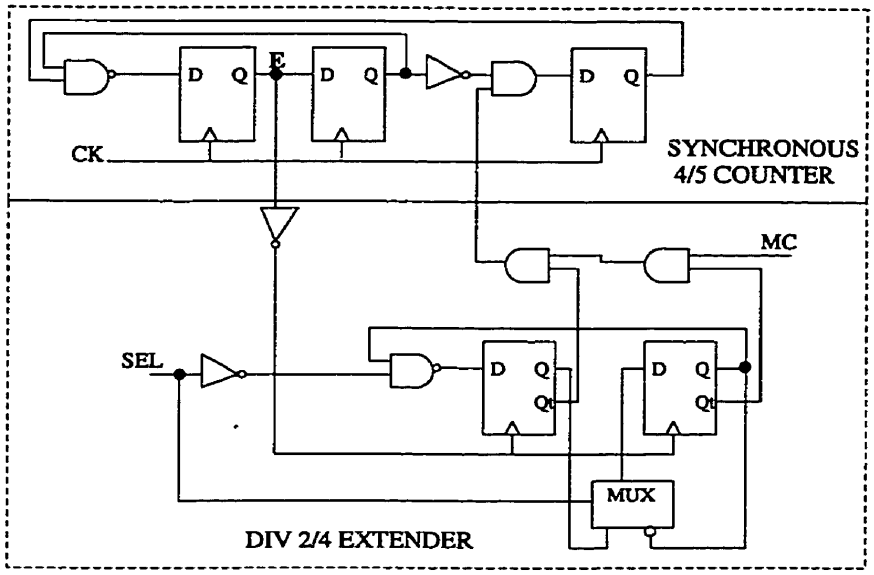


Figure 4.9. Dual-modulus 8/9 and 16/17 prescaler

Table 4.2. Truth Table for division ratio for the prescaler

SEL	MC	Division ratio
0	0	Divide-by-16
0	1	Divide-by-17
1	0	Divide-by-8
1	1	Divide-by-9

The rest of the divider was implemented in CMOS using a standard cell library. In order to cover an output frequency of around 1GHz, the sizes of the B and A counters were 6-bits and 3-bits, respectively. This gives a frequency division ratio of 56 to 519. Table 4.2 shows the frequency plan of the MASH ΣΔ PLL. Note that 1.2GHz is the maximum operating frequency of the prescaler, which is limited by the critical path of the divider (as is discussed below).

An ECL-to-CMOS level converter was necessary in order to interface between the prescaler and the rest of the divider. Fig. 4.10 shows a circuit schematic of the level

converter used. The level converter was designed to give a reasonable trade-off between delay, power, and phase noise.

Table 4.3. Frequency plan of the MASH ΣΔ PLL

Prescaler Division Ratio	**Reference Division Ratio**	**F_{MIN}**	**F_{MAX}**
8/9	1	728MHz	>1.2GHz
	2	364MHz	>1.2GHz
	4	182MHz	>1.2GHz
16/17	1	Cannot be used (out of range)	
	2	Cannot be used (out of range)	
	4	780MHz	>1.2GHz

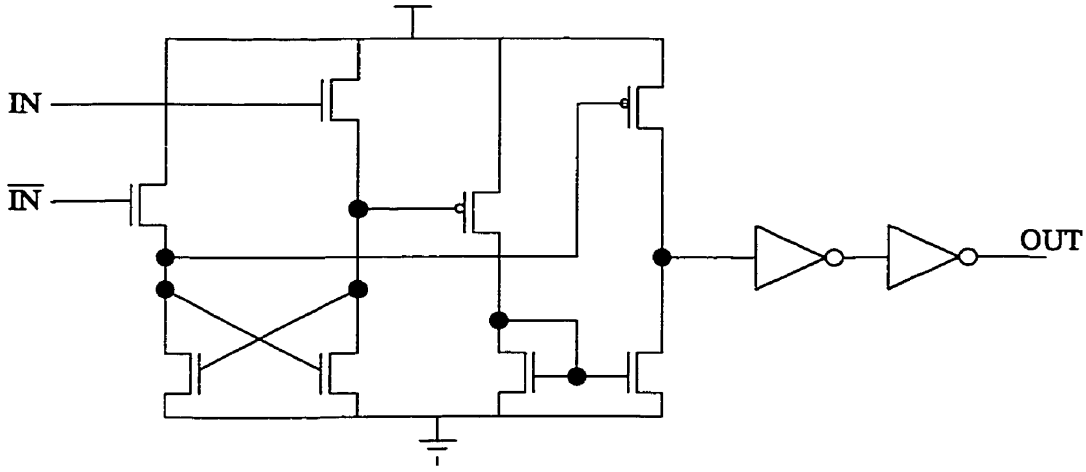


Figure 4.10. Circuit schematic of level converter

The critical path of the divider is of major concern. Fig. 4.11 shows a diagram of the critical path of the divider. This critical path sets the maximum frequency of the divider, and hence the PLL's maximum operating frequency. The critical path starts by a 0→1 transition at node E in the prescaler (see Fig. 4.9). This signal must propagate all the way to the output of the prescaler, which is connected to the clock input signals to the counters. The delay in the counters includes the setup time of the flip-flops used in the counters and the delay in the logic associated with the counters. The output of

the A counter then toggles and changes the value of the MC input to the prescaler. The MC input then must propagate through 3 ECL AND/NAND gates and one flip-flop. The worst case critical path was measured to be 6.78ns. Note that the critical path only occurs when the prescaler is dividing by 9. This means that the PLL's maximum operating frequency is approximately 1.3GHz (9/6.78ns ≈ 1.3GHz). This was simulated using worst case process, temperature, and supply voltage variation on the extracted layout of the frequency divider. Taking a safety margin of 0.1GHz gives the maximum reliable operating frequency for the prescaler of 1.2GHz.

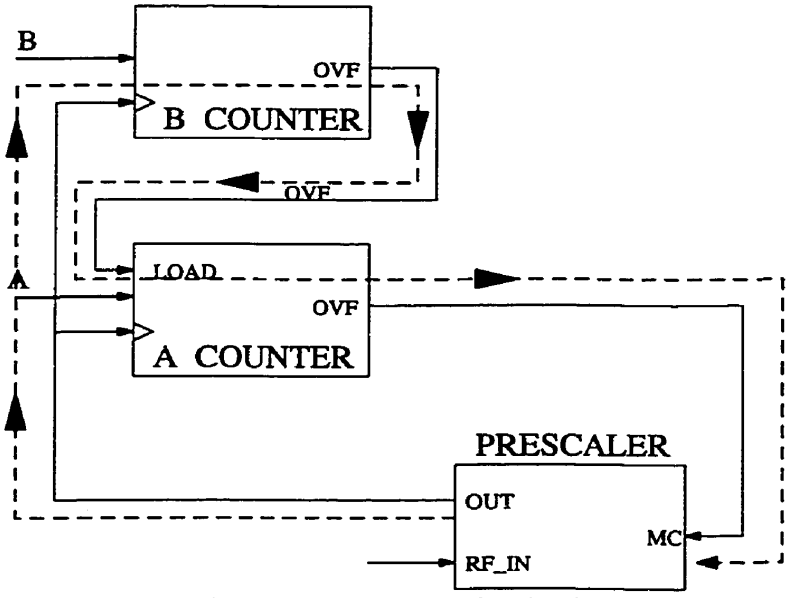


Figure 4.11. Critical path delay in the divider

4.3.2 Other Loop Components

4.3.2.1 Charge Pump Design

The charge pump was also included on-chip. Fig. 4.12 shows a schematic diagram of the charge pump used. This is a charge pump with a current steering switch [23]. One advantage of this architecture is that it has a fast settling time. Simulations indicated that the settling time of this charge pump to within 1% of final value is less than 7ns.

The reason for this small settling time is that the current sources are never turned off; their current either flows to the output or to one of the supply rails. In short, the structure provides a high-speed single-ended charge pump. The disadvantage of this architecture is that up/down current mismatches may occur due the difficulty in matching PMOS and NMOS devices.

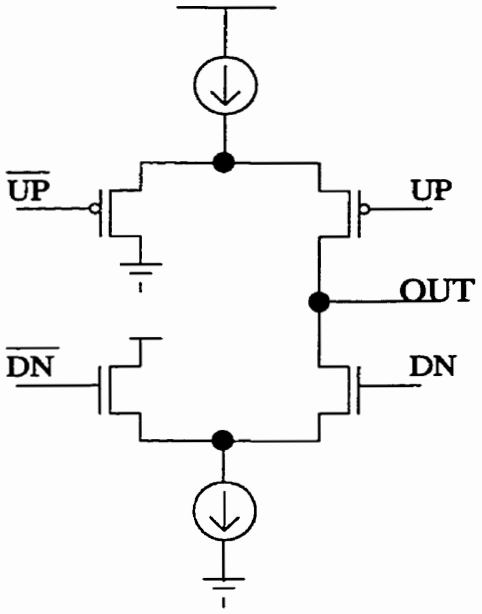


Figure 4.12. Circuit schematic of charge pump

4.3.2.2 Phase-Frequency Detector

A phase-frequency detector (PFD) is capable of detecting both phase and frequency differences. A conceptual diagram of the PFD implemented is shown in Fig. 4.13. Input $u_2(t)$ is the signal from the frequency divider and $u_1(t)$ is the reference signal (external input). If $u_1(t)$ lags behind $u_2(t)$, then this means that the signal generated by the VCO has too high of a frequency. In this case, the DN signal is enabled. The DN signal goes into the charge pump, which lowers the control voltage of the VCO and decreases the output frequency of the VCO. A similar explanation holds true for the case where $u_1(t)$ leads $u_2(t)$. Conceptually, the PFD can be thought of as an ideal

sampler with two mutually exclusive outputs. One output is asserted if the VCO frequency is lagging the reference frequency, and vice versa.

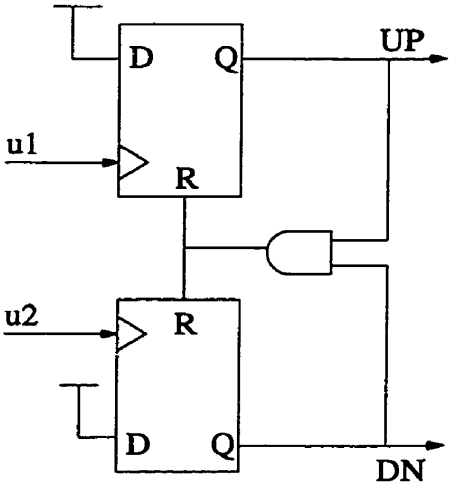


Figure 4.13. Conceptual diagram of a PFD

Note that since there are two flip-flops, there are four possible states. One state, which is when both flip-flops are set, is eliminated by feeding the signals from both flip-flop's outputs to and AND gate which in turn resets both flip-flops. The three remaining states are listed in Table 4.4 below.

Table 4.4. Valid phase-frequency detector states

State	State Condition	Meaning
-1	DN=1, UP=0	VCO frequency too high
0	DN=0, UP=0	PLL is in phase lock
1	DN=0, UP=1	VCO frequency too low

One important issue concerning PFDs is the phase resolution that they are able to detect. For example, if the PFD can detect phase differences between $u_1(t)$ and $u_2(t)$ that are at least 100 picoseconds apart, and the feedback division ratio is 10, this translates to a 1ns jitter on the PLL's output. The region in which the PFD cannot distinguish between the phases of the two input signals is known as the *deadzone band* [19]. In this deadzone band, the PFD behaves as a nonlinear device. This means that

instead of acting as an ideal sampler, the PFD can cause mixing of unwanted high frequency noise back to the baseband.

One method of reducing the effect of the deadzone is by using a PFD with a delayed reset signal [20], as shown in Fig. 4.14. This is similar to a conventional PFD except that there is a delay on the feedback reset path. Assuming the PLL is in lock, the PFD will always send two short UP and DN pulses simultaneously at every positive edge of $u_1(t)$ and $u_2(t)$. Assuming that a small differential phase develops between $u_1(t)$ and $u_2(t)$, the fall times of the two pulses may be much less than one gate delay. Using a PFD with a delayed reset signal, timing resolution of less than 10ps can be achieved. Note that the deadzone band may still exist if slow circuit techniques are used, or if the circuit delay paths between the input and the UP and DN outputs are not balanced.

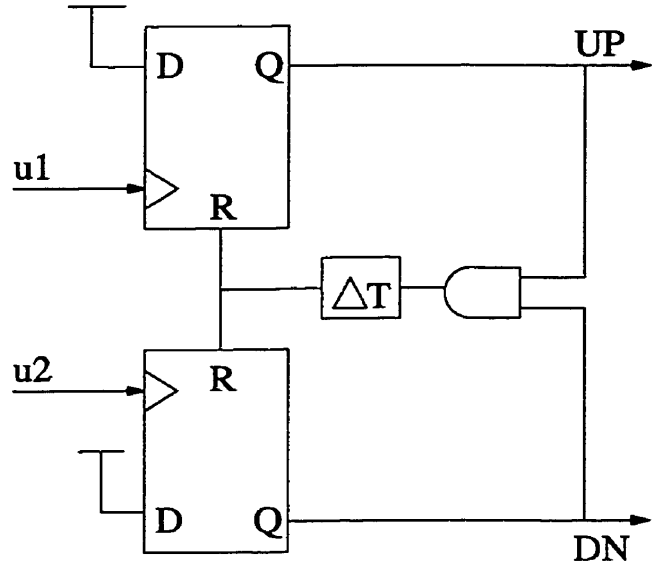


Figure 4.14. PFD with a delayed reset signal

4.3.2.3 Digital Circuit Implementation

A serial interface was used to program and test the PLL chip. This serial interface controls various parameters of the MASH ΣΔ PLL such as the division ratio, operating mode of the ΣΔ modulator, charge pump current, etc. Both the serial interface as well

as the entire digital ΣΔ modulator were implemented using standard cells.

Although its maximum operating frequency is higher than the targeted 13MHz, pipelining was used to speed up the ΣΔ modulator even further. This was done for two reasons. First, it enables an even higher reference frequency to be used during testing to determine the effect of increasing the reference frequency on the phase noise performance of the ΣΔ PLL. Second, it enables the ΣΔ modulator to work at lower supply voltages while still operating at 13MHz. This helps in reducing the power consumption of the ΣΔ modulator. A schematic diagram of a pipelined 24-bit accumulator is shown in Fig. 4.15.

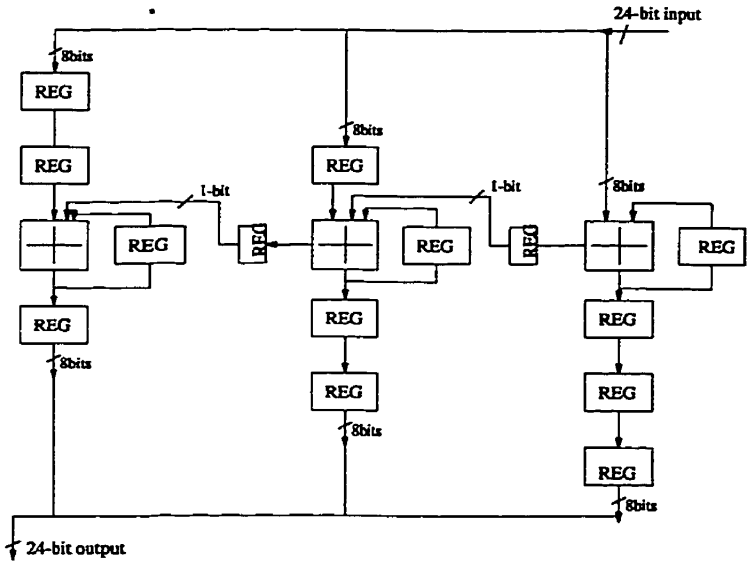
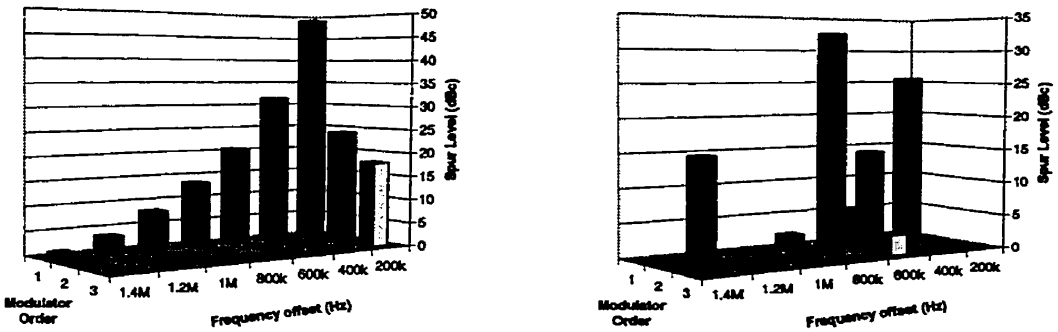


Figure 4.15. Pipelined 24-bit accumulator

4.3.3 Experimental Results

A test chip based on the design discussed above was fabricated and tested. The power consumption, phase noise, and spur level were measured for different fractional channels. A programming serial interface was created to ease testing of the chip. A detailed description of this programming interface is shown in Appendix A. The fabricated chip included the programmable digital ΣΔ modulator, serial interface,

charge pump, phase-frequency detector, divider, and reference buffer amplifier. An external VCO with center frequency at 800MHz and gain of 50MHz/V was used. For a 13MHz reference frequency, a fractionality of 1/65 is required to obtain a channel spacing 200KHz.



(a) (b)

Figure 4.16. Spurious performance for (a) 1/65 fractional channel and (b) 3/65 fractional channel

Several measurements were taken. The first set of measurements was the spurious performance of the ΣΔ PLL. The spurs for various orders of the ΣΔ modulator for fractional channels of 1/65 and 3/65 are shown in Fig. 4.16(a) and 4.16(b), respectively. The channel spacing is 200KHz. The first order ΣΔ modulator corresponds to a single accumulator controlling the division ratio. As the figures show, maximum spur for a 1^{st} order ΣΔ modulator is at the desired fractional ratio, as expected. Using 2^{nd} and 3^{rd} order ΣΔ modulators, the spurs at all channels but the desired fractional ratio are fully suppressed. It was expected that all spurs would be eliminated. Even when constant seed values with different magnitudes was used, the spurs were not completely eliminated. The spur levels, however, were less than that of the 1^{st} order ΣΔ PLL. The 2^{nd} order ΣΔ PLL helped reduce the spur level by 25dB and the 3^{rd} order ΣΔ PLL by 30dB, compared to the 1^{st} order ΣΔ PLL. Another significant measurement to mention

was that spurs for 2nd and 3rd order modulators for fractional channels beyond $\frac{5}{65}$ were completely suppressed. This is because the spur would be generated at 5*200KHz=1MHz offset away from the carrier, which is far enough in frequency to be suppressed by the closed loop low-pass filter response.

Another measurement taken was the close-in phase noise of the ΣΔ PLL. As mentioned earlier, the closed loop bandwidth is 30KHz. The close-in phase noise was measured at 10KHz offset. For the first and second order ΣΔ PLLs, a close-in phase noise of around –90dBc/Hz was measured for all fractional channels. For the third order ΣΔ PLL, the close-in phase noise was measured to be between –75 to -80dBc/Hz. This represents 10dBc/Hz to 15dBc/Hz degradation in phase noise. This phase noise has been attributed to the nonlinear behaviour of the phase-frequency detector due to the presence of a dead-zone region.

The power consumption of the ΣΔ PLL has also been measured. Assuming a 3.3V supply voltage powered to all parts of the PLL, the total power consumption of the test chip was 30mW. This is almost double the power consumption of the more competitive commercial fractional-N PLLs. During this design, the emphasis was on a robust and safe design, rather than a low-power design. As the results have shown, the close-in phase noise of the PLL is far less than the –70dBc/Hz, required for the GSM and GPRS standards. This extra performance can be traded off for less power consumption in future designs. The power breakdown of the chip is shown in Fig. 4.17 (assuming a third order ΣΔ modulator is used). As the figure reveals, the ΣΔ modulator is the largest contributer to power consumption. Using a first or second order ΣΔ modulator gives a total power consumption of 21.78mW and 27.78mW, respectively. If a 1.5V supply voltage is used for the ΣΔ modulator, the total power consumption is reduced to 27.6mW. This represents an overall power savings of 7.3% and reduces the ΣΔ modulator's power contribution to 9%, as shown in Fig. 4.18.

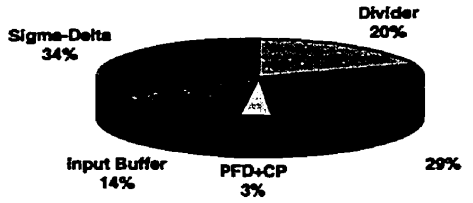


Figure 4.17. Power breakdown of MASH ΣΔ PLL for $V_{DD,SD}$=3.3V

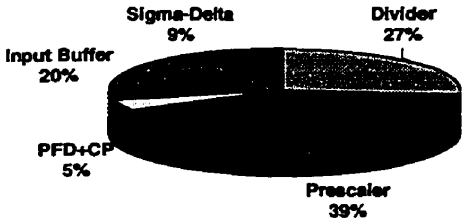


Figure 4.18. Power breakdown of MASH ΣΔ PLL for $V_{DD,SD}$=1.5V

4.4 Conclusions

In this chapter, the design of a MASH ΣΔ fractional-N PLL has been detailed. The design has been optimized for GSM and GPRS wireless communications standards. It has been demonstrated that sigma-delta modulation has succeeded in significantly reducing the spurs, but not completely eliminating them. The spur levels, however, meet the GSM and GPRS specifications. The power consumption of the ΣΔ PLL has

been dominated by the divider. In future designs, lower power can be achieved by using less current in the prescaler. The effect of using a constant seed has been investigated. A constant seed did not succeed in eliminating the spurs as anticipated. To completely eliminate the spurs generated by the fractional-N divider, different architectures must be explored.

Chapter 5

A Wideband Sigma-Delta Phase-Locked Loop Modulator

5.1. Introduction

The proliferation of wireless products over the past few years has been rapidly increasing. This has helped increase the momentum of making wireless computing a reality. New wireless data standards such as GPRS and HSCSD have brought new challenges to wireless transceiver design. One pivotal component of the wireless transceiver is the frequency synthesizer.

In mobile computing applications, narrow channel spacing is necessary to efficiently utilize the available frequency spectrum. On the other hand, fast switching from one channel to another is necessary for high-data rates. One way to satisfy these two conflicting requirements is to use a fractional-N phase-locked loop (PLL) architecture. Fractional-N PLLs are capable of synthesizing frequencies at channel spacings at less than the reference frequency. This helps increase the reference frequency and hence reduce the PLL's lock time [34].

One major disadvantage of fractional-N PLLs is the generation of high tones at multiples of the channel spacing. The use of digital sigma-delta (ΣΔ) modulation techniques in fractional-N PLL frequency synthesis has been previously suggested [38] to eliminate spurs. This is achieved by randomizing the feedback division ratio such that the quantization noise of the fractional-N divider is transferred to higher frequencies. Other advantages of ΣΔ PLLs include arbitrarily small frequency resolution, wide tuning bandwidth, and fast switching speed.

Although wider closed loop bandwidths are possible with ΣΔ PLLs, they are not wide enough to be readily used for closed loop modulation for current wireless standards, such as GSM. In [39], a ΣΔ PLL has been used as a closed loop modulator for the digital enhanced cordless (DECT) standard. In this technique, a narrow closed loop bandwidth was used to transmit a wideband signal by filtering the transmitted data by a high pass compensation filter. Using this technique, high frequency data is amplified, so that when filtered by the low pass closed loop PLL response, the high frequency data would not be attenuated. This approach, however, assumes perfect matching between the analog and digital filters, which is difficult to achieve in practice.

In this chapter, an alternate method of increasing the loop bandwidth of a ΣΔ PLL is investigated. This approach relies on modifying the digital ΣΔ modulator transfer function in order to attenuate quantization noise within the closed loop bandwidth of the loop filter. In section 5.2, existing ΣΔ PLL architectures are reviewed. In section

5.3, the proposed ΣΔ PLL architecture is explained. In section 5.4, the implementation of the various components of the ΣΔ PLL is detailed. This includes a discussion of the optimization techniques used to simplify the hardware necessary for the digital ΣΔ modulator. The overall performance of the PLL is also given. Conclusions are drawn in section 5.5.

5.2. Conventional Sigma-Delta PLL

Fractional-N frequency synthesis based on ΣΔ modulators offer wide bandwidth with arbitrarily small channel spacing. The first reported ΣΔ PLL used a MASH ΣΔ modulator[38]. A 1-1-1 MASH ΣΔ modulator (or three cascaded 1st order ΣΔ modulators) was used. It was shown that for a sufficiently small loop bandwidth, this architecture, is able sufficiently to filter out quantization noise and reference feedthough [17].

In another study [25], it was shown that as the loop bandwidth is widened, residual spurs start to appear. This is due to the fact that the assumption that the quantization noise is uniformly distributed does not always hold true. The reason for this is that using a ΣΔ for frequency synthesis is equivalent to using a ΣΔ modulator with a dc input. ΣΔ modulators are known to produce tones with dc inputs, even when higher order ΣΔ modulators are used. In [25], a feedback-feedforward sigma-delta modulator architecture [24] with single bit output was used. Single bit output was used in order to reduce the level of quantization noise[1]. The disadvantage of the architecture is that the range over which the ΣΔ modulator is stable is reduced, and hence the tuning range of the modulator is reduced. Furthermore, the hardware complexity of the ΣΔ modulator is higher than that of the MASH ΣΔ modulator.

---

[1] Note that in digital ΣΔ modulator the amplitude cannot be reduced; therefore, increase in number of output bits amounts to increase in quantization noise.

In [40], Rhee proposes another type of feedback feedforward ΣΔ configuration with three output bits. The peak of the quantization noise is flattened by introducing an extra pole into the digital ΣΔ modulator. This approach helps meet phase noise specifications at high frequency offsets while still using a high order ΣΔ modulator. Furthermore, dithering was used in order to introduce sufficient randomization in the ΣΔ modulator. The dithering circuitry was implemented by a linear feedback shift register (LFSR) [57] to produce a pseudo random number generator and then added to the LSB input of the ΣΔ modulator. Again, the disadvantage of this architecture is reduced tuning range and increased complexity compared to a MASH ΣΔ PLL.

In all the above approaches, the loop bandwidth was limited to nearly three orders of magnitude less than the reference frequency. This means that if the ΣΔ PLL is to be used for closed loop modulator, the power dissipation of the digital ΣΔ modulator may dominate the synthesizer. Other means of widening the loop bandwidth must be investigated.

5.3. Proposed Sigma-Delta PLL

In this study, another approach of modifying the ΣΔ modulator is investigated [62]. In order to be able to widen the loop bandwidth without sacrificing phase noise performance, the close-in phase noise must be kept within tolerable levels and the rise of quantization noise must be limited to meet high frequency offset phase noise requirements.

Note that the phase frequency detector (PFD) acts as a sampler which samples the frequency at the comparison frequency and downconverts it to dc. Since the ΣΔ transfer function is zero at this point, no quantization noise is mixed down to dc. In a ΣΔ PLL, the instantaneous sampling frequency may be different from the average comparison frequency, which means that some quantization noise may be sampled

down to dc. For a single bit digital ΣΔ, this is not too much of a concern. However, for a multibit ΣΔ, this may considerably raise the PLL's close-in phase noise.

In order to reduce the close-in phase noise, the ΣΔ transfer function is modified in such a way that one of the zeros is moved away from dc to a frequency equal to a multiple of the minimum fractional division ratio. This has the effect of inserting a notch at that frequency. The total quantization noise between this frequency and dc is effectively suppressed. Fig. 5.1 shows the noise transfer function (NTF) of the proposed ΣΔ modulator in comparison to others discussed in Section 5.2.

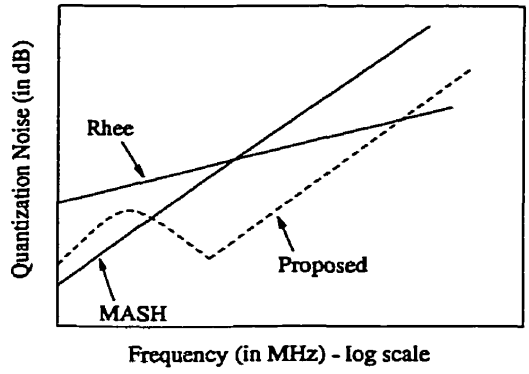


Figure 5.1. Quantization noise shaping by different ΣΔ modulators

To reduce the quantization noise at higher frequencies, the step size is reduced by producing fractional division ratios. This is achieved by using a modified phase selection divider [41]. A phase selection divider produces phase shifts of the VCO signal and changes the division ratio by selecting different phases from the VCO. In this study, this type of divider is used to produce quarter division ratios. Along with a 3-bit ΣΔ modulator, the total step size is reduced to that of a single bit ΣΔ modulator.

5.4. Implementation

5.4.1 Digital ΣΔ Modulator

The 3-bit ΣΔ modulator was implemented as an error feedback configuration. This is

the most suitable configuration when implementing a digital ΣΔ modulator [24]. For testing purposes, a selectable notch filter was implemented. Four notch frequencies were available, which were the first four fractional channel offsets away from the carrier. A high-level diagram of the modulator is shown in Fig. 5.2. A datapath width of 21-bits was chosen in order to obtain a frequency resolution of less than 10Hz when using a 13MHz reference frequency.

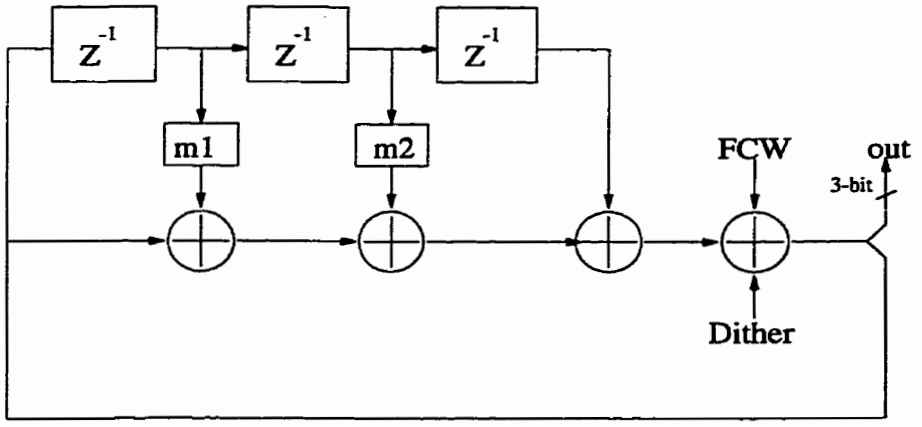


Figure 5.2. High-level diagram of proposed ΣΔ modulator

In order to reduce the hardware complexity of the divider, several optimizations were implemented. First, partial products of the multipliers that contributed negligibly to the output were eliminated. Furthermore, it was noted that in 3rd order notch filtering the second and third coefficients had the same magnitude, but opposite signs. This helped reduce a dot product operation to a simple negation operation. In order to reduce the number of expensive parallel adders, all addition operations were merged into one 20-input adder. This included the additions that are performed in accumulating partial products of the multiplier. The final modulator structure that was implemented is shown in Fig. 5.3.

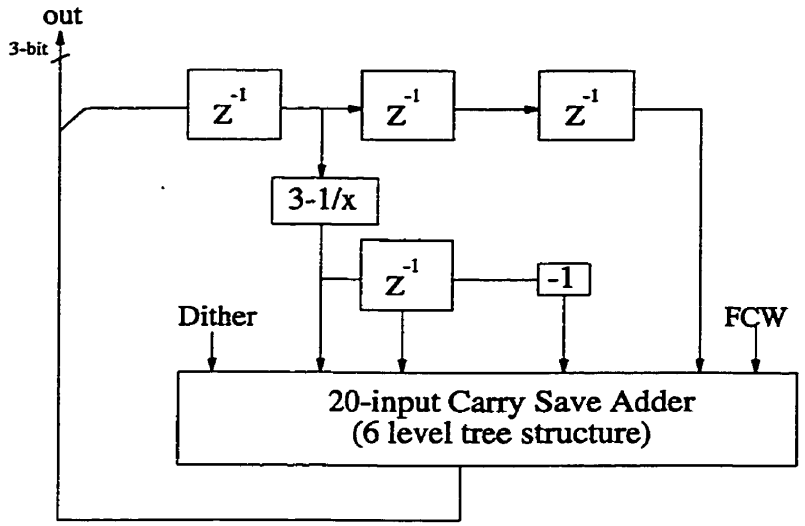


Figure 5.3. ΣΔ modulator with optimized arithmetic structures

5.4.1.1 Multiplier Reduction

As was explained earlier, partial products of the multipliers that contributed negligibly to the output were eliminated. This is now explained in more detail. In designing a sigma-delta modulator with four different cut-off frequencies, the exact values of the transfer functions was known. In this case the transfer functions were:

$$H_{200}(z) = 1-(3-\frac{612}{2^{16}}z^{-1})+(3-\frac{612}{2^{16}}z^{-2})-z^{-3} \tag{5.1}$$

$$H_{400}(z) = 1-(3-\frac{2442}{2^{16}}z^{-1})+(3-\frac{2442}{2^{16}}z^{-2})-z^{-3} \tag{5.2}$$

$$H_{600}(z) = 1-(3-\frac{5473}{2^{16}}z^{-1})+(3-\frac{5473}{2^{16}}z^{-2})-z^{-3} \tag{5.3}$$

$$H_{800}(z) = 1-(3-\frac{9676}{2^{16}}z^{-1})+(3-\frac{9676}{2^{16}}z^{-2})-z^{-3} \tag{5.4}$$

which correspond to sigma-delta transfer functions with notch frequencies at 200KHz, 400KHz, 600KHz, and 800KHz, respectively, assuming a reference frequency of 13MHz.

One way to reduce complexity is to express the coefficients of the transfer functions in powers of two. This way, the minimum number of bits needed to express these coefficients can be found. Circuit complexity can be reduced by truncating a sufficient number of bits without significantly altering the transfer function characteristics. Table 5.1 shows the effect of truncating the number of non-zero terms in representing the coefficients on the frequency accuracy of the notch filter. As the table shows, for an accuracy of 1KHz, it is necessary to use up to six nonzero bits. These terms are negated and added to "3" to produce a term in the form of $3-\frac{1}{x}$, which is the term shown in all four equations listed in (5.1) – (5.4). The block which performs this computation is the "$3-\frac{1}{x}$" block shown in Fig. 5.3.

Table 5.1. Effect of coefficient bit truncation on frequency accuracy

Notch Freq.	**4 terms**	**5 terms**	**6 terms**	**7 terms**
200KHz	12.4Hz	--	--	--
400KHz	368Hz	29Hz	--	--
600KHz	6.51KHz	159Hz	39Hz	--
800KHz	15.2KHz	2.47KHz	880Hz	86Hz

5.4.1.2 Multi-input Adder

As Fig. 5.3 shows, the largest block in the proposed ΣΔ modulator is the multi-input adder. Optimizing this adder structure is pivotal in decreasing the area and power consumption of the entire PLL. One way of implementing this adder is by using a simple carry save addition tree (CST) [42]. However, a CST is not capable of handling negative numbers. Furthermore, the operands of the adder vary from 4 bits to 21 bits. This means that if sign extension is used to handle negative number, it would be a very expensive solution. Alternate addition algorithms must be explored.

In this study, a multi-input addition algorithm is proposed which is capable of adding several operands without sign extension. Several algorithms are proposed in the

literature which are capable of performing this operation, but these algorithms are optimized specifically for the case of adding the partial products of a multiplier [43]. Generally speaking, these algorithms cannot be used to add several operands that have the same weight. In the proposed algorithm, the most significant bit (MSB) of each operand is given a negative weight. Adding the MSB of an operand with another number would be the same as subtracting the MSB from the other number. Using this type of arithmetic, four different cells are possible depending on the number of negatively weighted inputs. The four possible combinations are:

$$\begin{aligned} (C,S) &= X+Y+Z \\ (C,-S) &= -X+Y+Z \\ (-C,S) &= -X-Y+Z \\ (-C,-S) &= -X-Y-Z \end{aligned} \qquad (5.5a\text{-}d)$$

where X, Y, Z are the inputs the each cell, and C and S are the Carry and Sum outputs of the adder/subtractor cells. Note that the first and last equations are implemented by the same circuit. This means that two's complement addition of several operands is possible using only three cells in the carry save tree (CST). To add 20 operands, a six level CST is required. Using the type of arithmetic illustrated by equations (5.5a-d), the number of required cells was reduced by more than half in comparison to a sign extension solution. This greatly aids in reducing the area and power dissipation of the digital sigma-delta modulator.

The CST reduces the 20 operands to two operands. A special adder is then required to add these two remaining operands since they may contain negatively weighted bits. The CST has been interfaced with the last stage ripple adder in such a way that at most only one input of the last stage ripple adder may be negatively weighted. This means that if a carry ripple adder is used as the final stage adder, the output of each full-adder (FA) cell may have positive or negative terms. In order to avoid negative SUM terms, positive and negative weights on the C_{out} term were allowed, as shown in Fig. 5.4(a). This means that the output range of the FA cell is [-2,3]. Since both the C_{in} and C_{out}

signals can take on values of [-1,0,1], two bits were used to represent each of these signals. The binary mapping is shown in Table 5.2. The "B" input to the FA can also take on values of [-1 ,0,1]. Since it can be determined in advance which outputs from the CST are negatively and positively weighted, two different cells were created; in one cell the "B" input is negatively weighted, and in the other cell it is positively weighted. A block diagram of the resulting FA is shown in Fig. 5.4(b).

Table 5.2. Binary representations of carry signals

Cout	CoutP	CoutN
-1	0	1
0	0	0
1	1	0

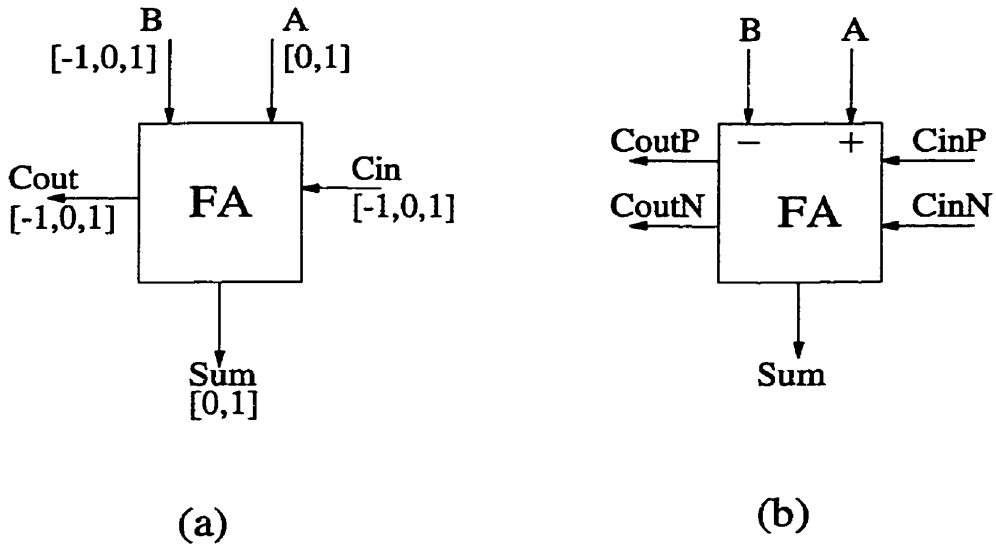


Figure 5.4. Final Stage Full-adder (FA) cell. (a) input/output ranges and (b) block diagram of implemented circuit

In order to demonstrate the effectiveness of the numerical optimizations done to the sigma-delta modulator it was synthesized and its layout was compared to that of the MASH sigma-delta modulator implemented in the previous chapter. Both modulators were implemented in a 0.35μm CMOS technology using a standard cell library. The

MASH sigma-delta modulator was found to occupy an area of $1mm^2$, whereas the notch filter sigma-delta occupied an area of only $0.71mm^2$. This is truly a testimony of the efficiency of the algorithmic and numerical optimizations done in this work.

5.4.2 Fractional Divider

A programmable frequency divider is typically implemented by a prescaler and two accumulators as demonstrated in chapter 4. To achieve low power design, it is desirable to use an asynchronous divider to minimize the amount of circuitry operating at high frequencies. Several dual-modulus dividers implemented as asynchronous dividers have been demonstrated in literature [44-46]. Recently, a multi-modulus divider has been presented which utilizes an asynchronous divider structure.

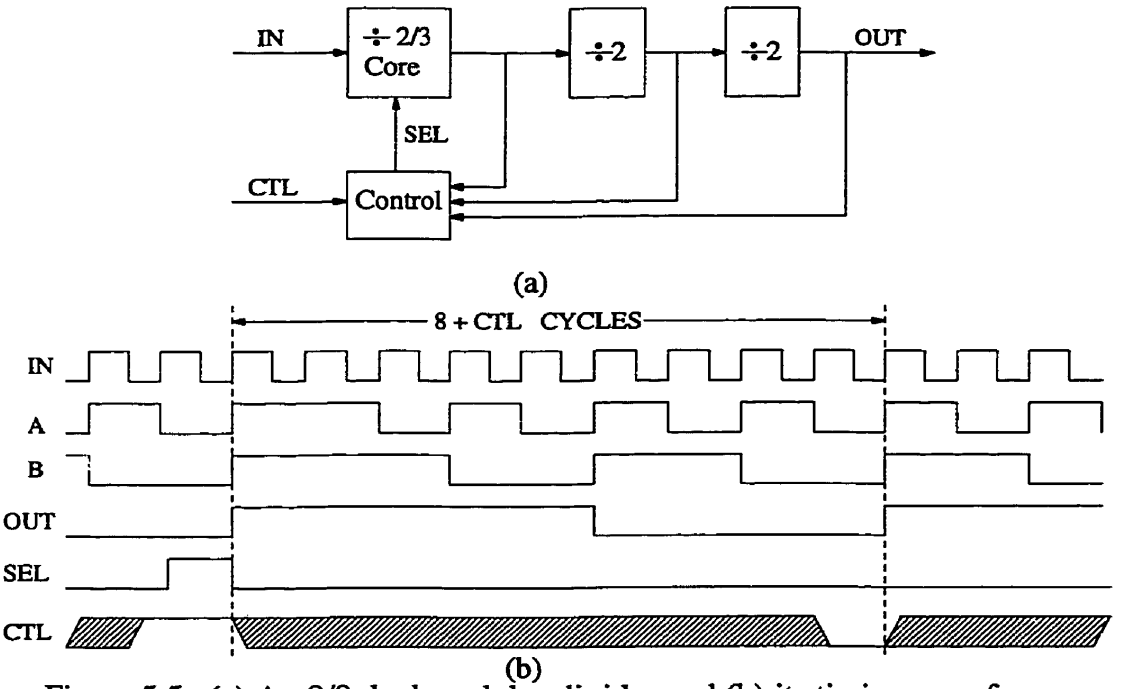


Figure 5.5. (a) An 8/9 dual-modulus divider and (b) its timing waveform

First, the multi-modulus divider architecture based on phase selection is reviewed, followed by an explanation of the proposed divider architecture. A standard 8/9 dual-modulus divider along with its timing waveform are shown in Fig. 5.5. This structure

consists of a divide-by-2/3 state machine followed by an asynchronous divide-by-4 section. When the SEL signal to the divide-by-2/3 block is not asserted, the total division ratio of the divider is eight. When the SEL signal is asserted, an input cycle is swallowed creating a total division ratio of nine, as shown in Fig. 5.5(b).

A multi-modulus divider based on the technique discussed above is shown in Fig. 5.6. This divider supports all integer division ratios between 8 and 15. Each block consists of a divide-by-2/3 cell. As shown in the figure, the control complexity of the divider increases linearly, with the most complex control working at the divider's highest frequency. This fact limits the size and speed at which this divider architecture can be used.

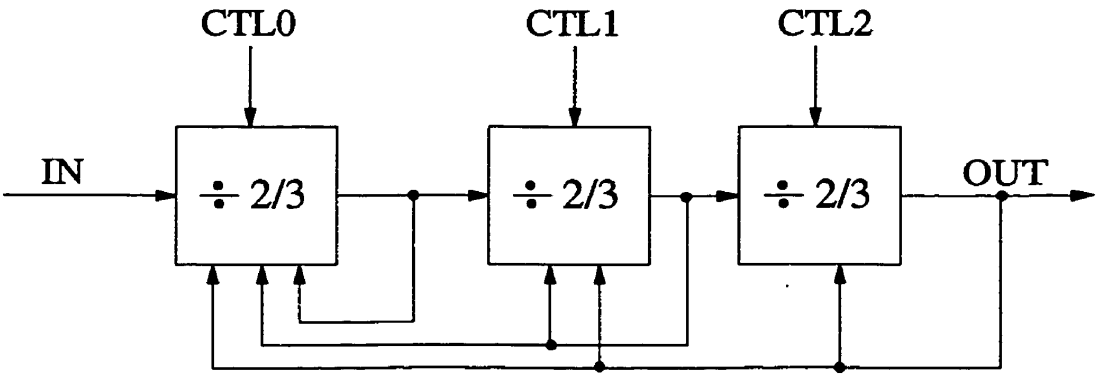


Figure 5.6. A multi-modulus divider architecture

One method of reducing the control complexity is to replace the control logic with phase selection logic. This is illustrated by different implementations of a Johnson counter, shown in Fig. 5.7. In Fig. 5.7(a), a logic implementation of a divide-by-2/3 Johnson counter is shown. The critical path consists of gating logic as well as the delay in the latches. Note that the gating logic operates at the input clock rate. In Fig. 5.7(b), the output of a divide-by-2 Johnson counter is fed into a multiplexer. This has several advantages. First, the gating logic in the critical path is eliminated. This is replaced by multiplexing circuitry at the output of the high-speed divide-by-2 circuit. This multiplexing logic works at half the input clock frequency. Furthermore, the input

clock load capacitance of this structure is less than that of a gated divide-by-2/3 structure.

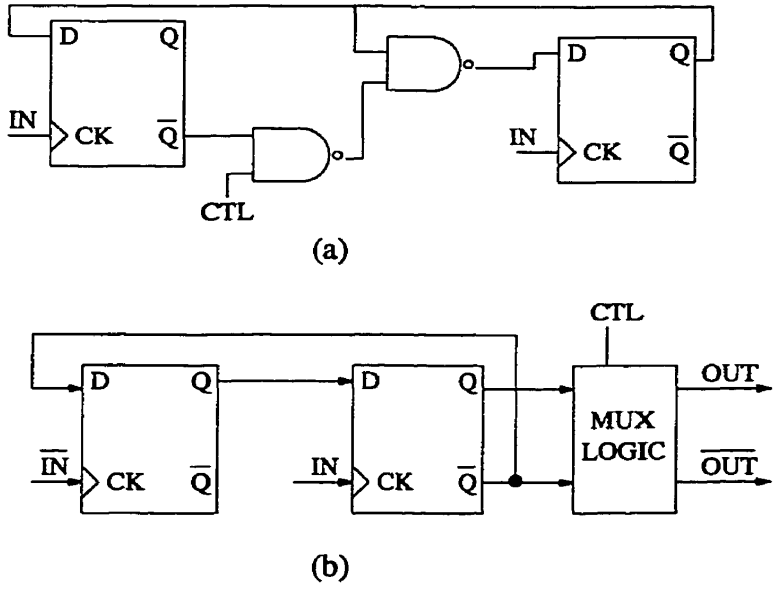


Figure 5.7. A Johnson counter. (a) divide-by-2/3 using gating logic and (b) a divide-by-2/3 using multiplexing

In this work, the concept of multiplexing between different phases of the output of the divide-by-2 circuit is used. Instead of dividing the VCO input frequency by 2, fractional frequency division is achieved by selecting one of four phases of the VCO output, as shown in Fig. 5.8. This structure enables quarter fraction division ratios. Obtaining smaller fractional division ratios is limited by the ability to extract more VCO phases. The high-speed control logic associated with the fractional divider allows it to swallow more than one input cycle in order to obtain four division ratios. Consider, for example, the case when the division ratio is changed from 1 to 1.25. In this case, the phase selection is incremented by 90°, allowing a quarter pulse to be swallowed. If the division ratio is changed from 1 to 1.5, the phase selection is incremented by 180°, allowing two pulses to be swallowed, and so on.

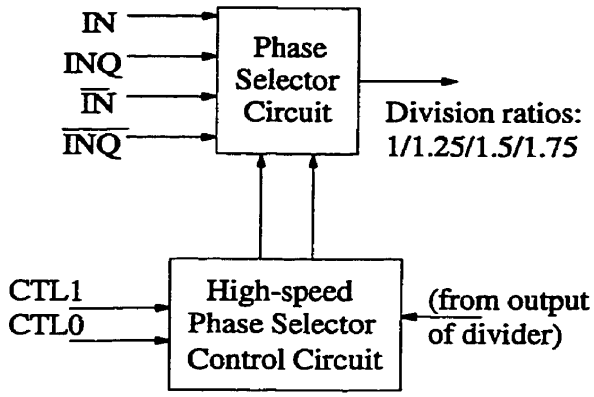


Figure 5.8. Proposed fractional divider circuit

The state machine required for glitch free operation is quite involved. In [47], glitch free operation is realized by switching in a phase that is not active during the time that the switching takes place. This means that intermediate phase switching may have to be done in order to switch to the final desired phase. This method, however, requires latches that can produce complementary outputs that can have both of their outputs, Q and $\overline{Q}$, to be high during certain states. This may complicate the design and limit the type of logic families that can be used to implement the high-speed dividers. Furthermore, this approach cannot be used for the initial fractional division since there is no frequency division between the VCO and the multiplexers.

In this study, a control logic mechanism has been developed which does not require extra intermediate states. Glitches, due to switching in a phase that is active during the time that the switching takes place, are avoided by using redundant logic. A block diagram of the control logic used is shown in Fig. 5.9. As shown in the figure, two phase selectors are required: a primary and an auxiliary phase selector. When a change in phase selection occurs, which generates a glitch, the switching of the primary phase is inhibited until the glitch is over. This is accomplished by buffering the phase selection bits and enabling them only when the outputs of both the primary and

auxiliary phase selectors are the same, after a glitch has occurred on the auxiliary circuit.

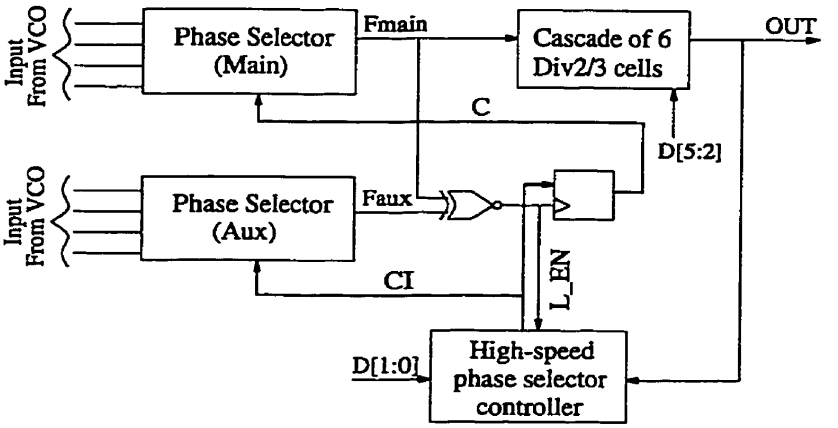


Figure 5.9. Control logic for glitch free operation

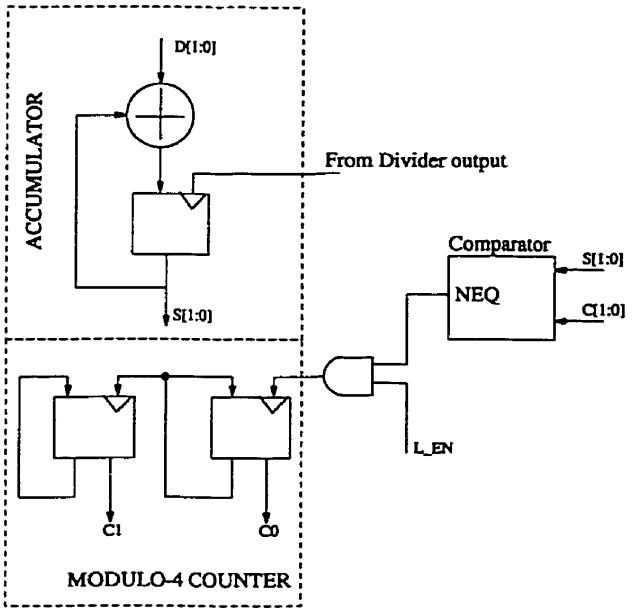


Figure 5.10. High-speed phase selector controller

Careful design was also required in the high-speed frequency control circuit for the phase selector. The digital frequency word, which specifies the desired frequency division ratio, must be translated into the correct control signals for the phase selector. This is done by the control logic shown in Fig. 5.10. An accumulator is required to increment the current stopping state by the required division ratio. A modulo-4

asynchronous counter is used to change the phase selector control signals to be equal to that of the accumulator. The clock input of the asynchronous counter is gated with L_EN so that the phase selector control signals do not change until a glitch in the auxiliary phase selector is resolved. Also note that it is necessary to change from one division ratio to another by single step sizes in order to avoid glitches.

A block level diagram of the proposed multi-modulus divider is shown in Fig. 5.11. The first section is a high-speed multiplexing circuit, which is used to implement a divide-by-1/1.25/1.5/1.75 cell. The rest of the divider consists of cascaded divide-by-2/3 cells. The total division range of this structure is 64 to 127.75 in quarter steps. Gated logic was used to implement the divide-by-2/3 cells to avoid the large control logic needed for phase selection.

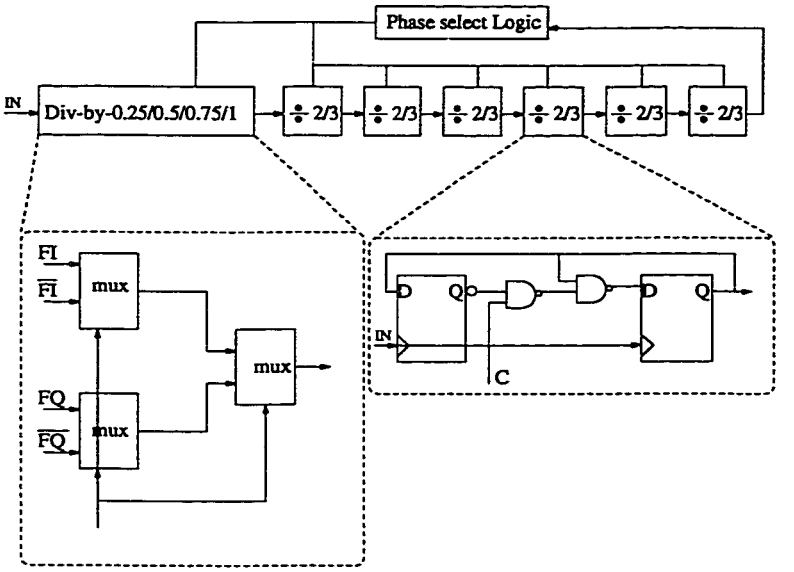


Figure 5.11. Phase selection divider with quarter division ratios

The use of gated logic for the divide-by-2/3 cells greatly simplifies the required control logic. A proposed method of avoiding the linear growth of control logic with increase in divider size is shown in Fig. 5.12. As this figure shows, only a 3-input AND gate is required per stage.

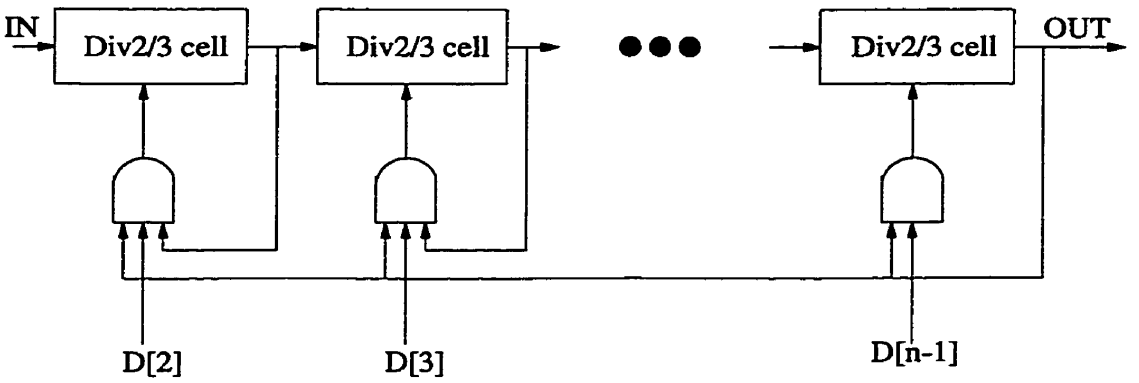


Figure 5.12. Control logic for asynchronous divide-by-2/3 cells

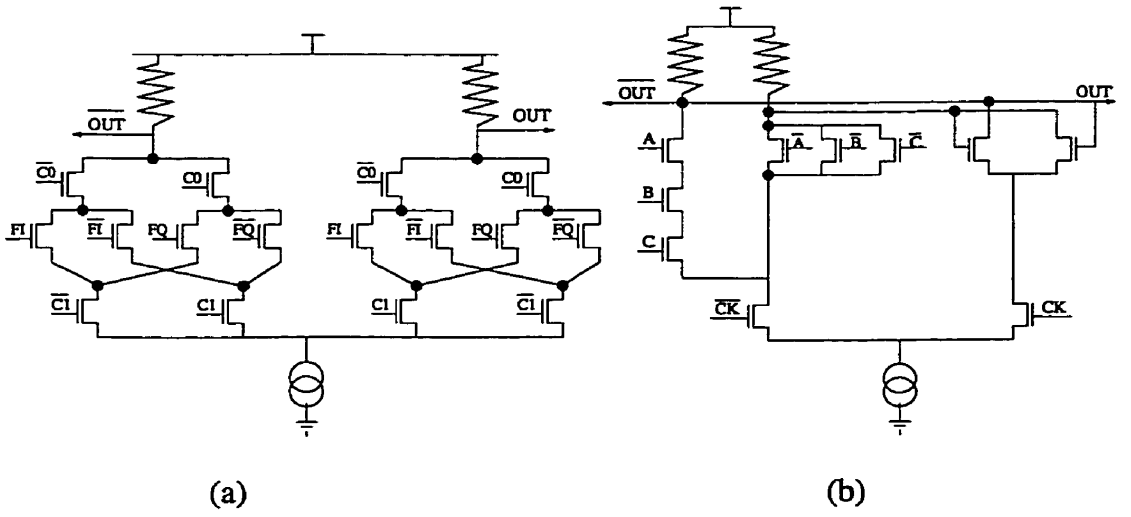


Figure 5.13. Current mode logic cells for (a) phase selector and (b) merged 3-input AND/Latch gate.

A high-speed logic family is required for the first stage multiplexers, since it operates at the VCO frequency. Current mode logic (CML) is used to implement this stage. The first stage divide-by-2/3 cell is also implemented in CML. Note that one major advantage of CML is that several logic functions can be easily cascaded into one gate. A single stage CML 4-to-1 multiplexer and merged 3-input AND/Latch gates are shown in Fig. 5.13(a) and 5.13(b), respectively. The rest of the divider was implemented in standard CMOS logic.

5.4.3 Other Loop Components

5.4.3.1 Charge Pump Circuit

The design of the charge pump is critical in eliminating dead zone as well as in reducing the close-in phase noise performance of the PLL. The switching time of the charge pump is also critical. For fast current switching, a current steering type charge pump is used. This avoids turning the charge pump current sources on and off, which helps to reduce the time it takes for the current switches to settle. A schematic diagram of the charge pump circuit is shown in Fig. 5.14.

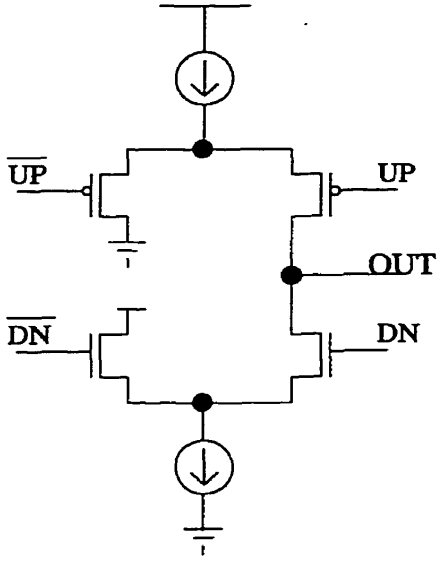


Figure 5.14. Schematic diagram of charge pump current

5.4.3.2 Loop Filter Parameters

The choice of the loop filter components are critical since they directly impact the overall loop transient characteristics, reference feedthrough suppression, and the close-in phase noise. The loop parameters have been chosen such that they meet the following requirements:

- Closed loop bandwidth of 200KHz,
- Damping factor is 0.707,
- Charge pump current is 50μA,
- VCO gain is 50MHz/V, and
- The phase margin is 60°.

Based on these requirements, the loop parameters in Table 5.3 were used. Note that the capacitor values are nearly two orders of magnitude less than that of the PLL designed in chapter 4 (which had a closed loop bandwidth of 30KHz). Also note that the speedup technique used in the previous chapter was also used in this implementation.

Table 5.3. Loop parameters used for the proposed ΣΔ PLL

Loop Parameter	Value
C_1	6.12pF
C_2	152.56pF
R2G	9.73KΩ
R2SPD	9.73KΩ

5.5. Simulation Results

The proposed ΣΔ PLL was implemented in a 0.35μm CMOS technology. A nominal supply voltage was used for the frequency divider and the charge pump. The layout of the chip is shown in Fig. 5.15. The chip included the digital ΣΔ modulator, frequency divider, phase-frequency detector, charge pump, and loop filter. An external VCO with frequency gain of 50MHz/V was assumed. The control signal for speed-up was made external for programming flexibility. Other digital pins for the chip included the 8-bit frequency control word and seed enable signal. Separate supply voltages were used for the divider, charge-pump, and digital circuitry. In order to reduce the power consumption of the digital ΣΔ modulator, its supply voltage was lowered to 1.5V. Pipelining was used in order to compensate for the loss in performance due to lower

supply voltage. This resulted in almost a 4.5x reduction in power consumption. The power consumption breakdown is shown in Fig. 5.16. Note that most of the power is consumed by the frequency divider and the digital circuitry. Although the size of the digital circuitry is large, its power consumption was kept to a minimal due to the reduction in supply voltage. The frequency divider consumes a sizable amount of power due to the high component count working at the VCO frequency.

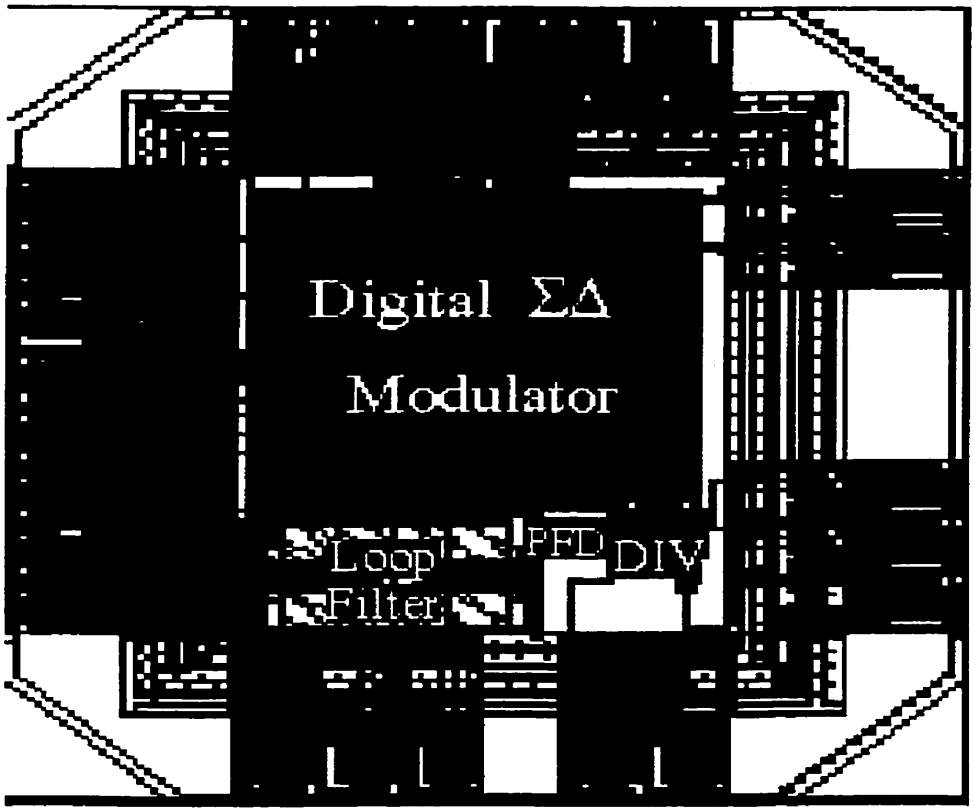


Figure 5.15. Layout of ΣΔ PLL

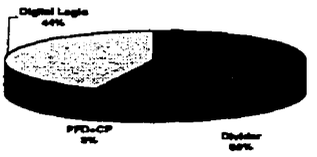


Figure 5.16. Power consumption breakdown of the ΣΔ PLL

In order to accurately predict the PLL's performance in a time feasible manner, a high-level description language was used to model the ΣΔ PLL. The digital components of the PLL were modeled in Verilog [48] and the analog components in Verilog-A [49]. The mixed mode simulator, SpectreRF-Verilog [50] (which is available through cadence), was used to simulate the PLL. This enabled the full lock characteristics of the PLL to be simulated in less than an hour, as opposed to a few days when using a SPICE based simulator.

The lock time performance of the ΣΔ PLL is critical. The frequency spacing between the minimum and maximum frequency channels for GSM is 25MHz. Fig. 5.17 shows that the maximum lock time required for 100Hz resolution is less than 15μs. Note that although it was built into this PLL, two-mode fast lock technique was not used to obtain this small lock time. Although this is much less than what is required for the GSM specifications, it has several advantages. First, it adds design flexibility to the system designer in that more time of the user's transmission time slot can be used for useful data transmission, as opposed to waiting for the frequency synthesizer to lock. Second, less standby power can be achieved since the time that the receiver must remain on during channel scanning is reduced.

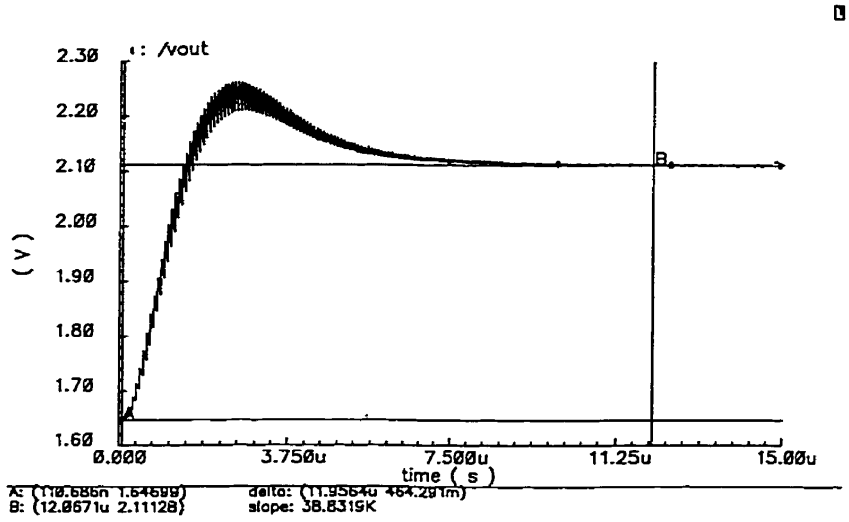
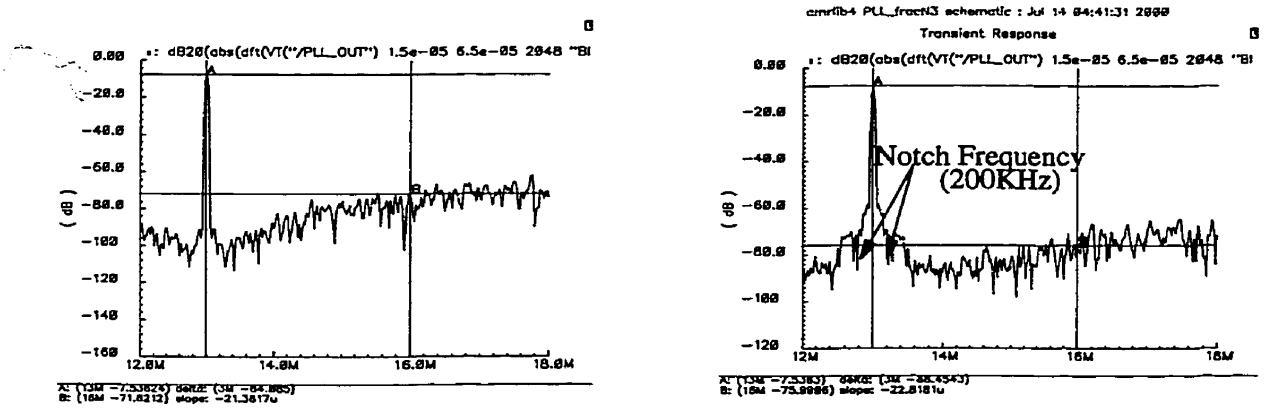


Figure 5.17. Worst case lock time characteristics



(a) (b)

Figure 5.18. Frequency spectrum output of the ΣΔ PLL

The frequency spectrum of the PLL is also important. Shown in Fig. 5.18(a) and (b) below are the frequency spectra of the ΣΔ PLL without and with a notch frequency, respectively. The notch frequency is of 200KHz and a closed loop bandwidth is also 200KHz. Notice the suppression of quantization noise at 200KHz in Fig. 5.18(b). Also notice that the phase noise at 3MHz is -64dBc for a frequency resolution of 10KHz, or

−144dBc/Hz, which meets the GSM specification. This means that this PLL can be used as the entire transmitter in a wireless transceiver. This represents large savings in term of power consumption and area, since a digital-to-analog converter (DAC), analog quadrature mixers, and an extra PLL are all replaced by just one wideband PLL. Notch frequencies at 400KHz, 600KHz, and 800KHz were also simulated. These simulations revealed no significant advantage over the notch filter with 200KHz. This is due to the fact that the further away the notch frequency is located, the more close-in phase noise (resulting from quantization noise) is allowed to accumulate over the closed loop bandwidth.

5.6 Performance Evaluation

In order to quantitatively evaluate the performance of the PLL described above, a figure-of-merit (FOM) suitable for wideband PLLs. The proposed ΣΔ PLL architecture is then compared to other ΣΔ PLLs recently reported in the literature.

The proposed figure-of-merit for evaluating fast lock PLLs for wireless applications is given as

$$FOM = \left\{ \underbrace{10 \cdot \log_{10}\left[\frac{f_{PFD}}{f_{BW}}\right]}_{\substack{\text{oversampling} \\ \text{ratio}}} + \underbrace{\frac{10}{-20 \cdot \log_{10}\left[\Phi_{ref} \cdot \Phi_{in-band}\right]}}_{\text{spectral-purity}} \right\} \cdot \Sigma\Delta_{bits} \qquad (5.6)$$

where Φ_{ref} and $\Phi_{in\text{-}band}$ are the reference spur and the in-band phase noise, respectively. The oversampling ratio is given by the ratio of the PFD comparison frequency (f_{PFD}) and the closed loop PLL bandwidth (f_{BW}), and $\Sigma\Delta_{bits}$ is the effective number of output bits of the digital ΣΔ modulator.

Table 5.4, below, shows the performance comparison of this work with previsously published works. Other works include the ΣΔ PLLs described in section 5.2 as well as

some other recent ΣΔ PLL implementations. Not all PLL implementations were tuned for the GSM standard. All PLL implementations had similar output and input reference frequencies. As the table shows, the proposed PLL has the best FOM measure.

Table 5.4. Comparison of this work with other ΣΔ PLLs

Ref.	Technology	Architecture	$\Sigma\Delta_{bits}$	f_{PFD} (MHz)	f_{BW} (kHz)	$\Phi_{in\text{-}band}$ (dBc/Hz)	Φ_{ref} (dBc)	$\frac{f_{PFD}}{f_{BW}}$	FOM
Filiol et. al. [72]	CMOS(ΣΔ) BJT (PLL)	4th order MASH	4	20	100	-95	-90	200	4.975
Perrot et al [39]	0.6μm CMOS	2nd order MASH	6	20	84	-74	-60	238	10.64
Riley et. al. [25]	Discrete	3rd order	1	10	30	-85	-83	333	1.501
Miller et al. [38]	Discrete	3rd oder MASH	3	0.2	0.75	-70	-133	267	3.586
Ree et. al. [40]	0.5μm CMOS	3rd order	3	8	40	-92	-95	200	3.691
This work	0.35μm CMOS	3rd order	1	13	200	-80	-85	65	1.099

5.7 Conclusions

A wideband phase-locked loop (PLL) modulator for wireless applications is reported. This modulator is based on PLL fractional-N frequency synthesis techniques along with sigma-delta modulation to randomize fractional-N spurs. A modified sigma-delta function allows for suppression of sigma-delta noise at lower frequencies, and hence allows for wider loop bandwidth. Also, sigma-delta quantization noise is reduced by using fractional division ratios. Low-power and low-area algorithmic techniques are used in the modified sigma-delta modulator in order to make it a feasible option. The wide bandwidth of 200KHz also makes the proposed ΣΔ PLL suitable for closed loop modulation purposes.

Chapter 6

A Digital Fast Lock PLL Architecture

6.1 Introduction

In the previous chapter, fast lock was achieved by expanding the loop bandwidth large enough to reduce the lock time as much as possible while preserving the PLL's spur suppression capability. In this chapter, a different approach is attempted. A completely digital solution to control the control voltage is attempted. It is shown that this architecture is suitable for fast lock low frequency resolution applications (such as wireless LAN and cordless), and it can also be used as a frequency lock aid for high

resolution applications (such as GSM and GPRS). Furthermore, a technique to reduce the phase lock time in high resolution applications is demonstrated.

In a conventional PLL architecture, the phase-frequency detector (PFD) controls the inputs of the charge pump. As was shown in chapter 3, the charge pump is on for only a fraction of the reference period. This contributes to cycle slipping during frequency locking. During phase lock, the fraction of the period during which the charge pump is on becomes extremely small. This results in a very slow phase lock procedure.

In this chapter, the PFD and divider are replaced by an alternative digital solution. The advantage of this approach is that the charge pumps may be turned on for the entire duration of the reference signal's period. Also, the frequency resolution can be tuned by simply altering the architectural parameters.

6.2 Proposed PLL Architecture

6.2.1 Basic Architecture

A block level diagram of the proposed architecture is shown in Fig. 6.1 [61]. As shown, the phase-frequency detector (PFD) and the frequency divider are replaced by alternative digital circuitry. The main idea behind this architecture is to produce a digital word proportional to the instantaneous frequency of the output of the VCO (or fixed prescaler). This digital representation is then digitally compared to another word, the frequency control word (FCW). The FCW is an input digital word, which effectively controls the output frequency of the PLL. If the instantaneous frequency is less than the FCW, then the UP signal is asserted; on the other hand, if the instantaneous frequency is more than the FCW, then the DN signal is asserted. Once the UP or DN signal is asserted, it is continuously asserted until the output of the digital comparator changes. If the instantaneous frequency is equal to the FCW, then both the

UP and DN signals are deasserted and the loop filter is left unchanged. Unlike the PFD, there is no state where the UP and DN signals are asserted simultaneously.

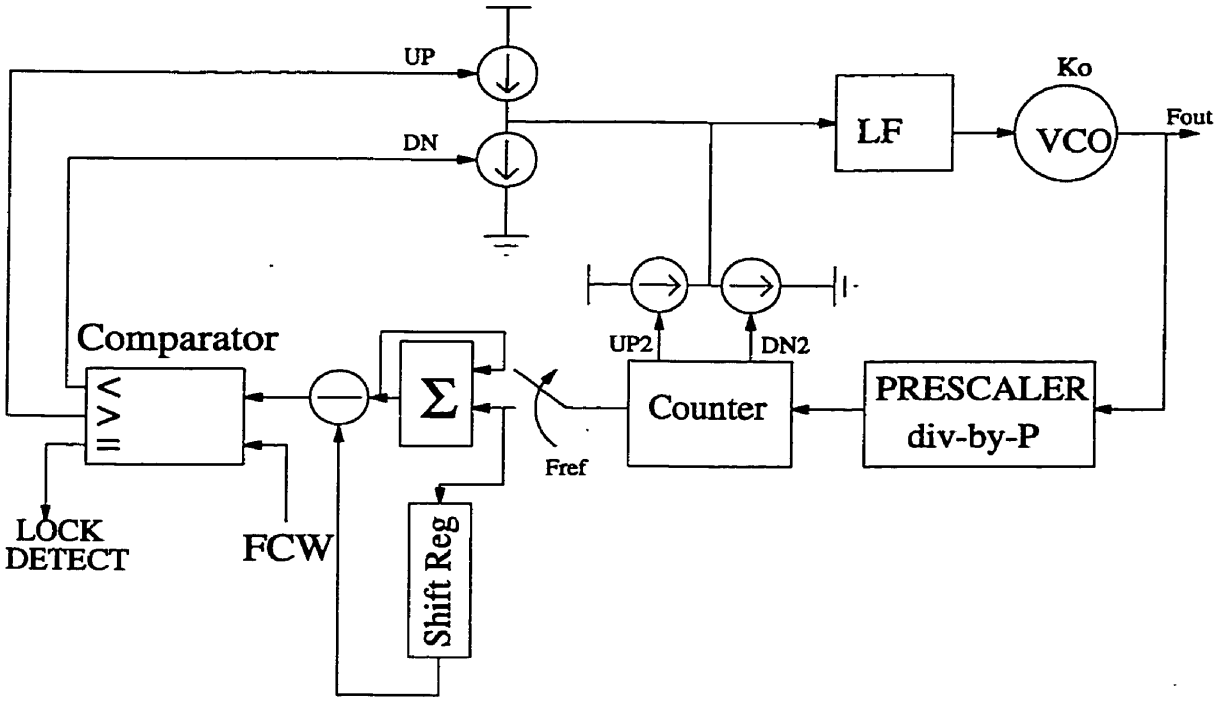


Figure 6.1. Block diagram of proposed PLL architecture

The frequency detection process is accomplished by the counter-accumulator-shift register combination. Its operation is best described by an example. Let the constant prescaler division ratio be 8, and the counter be a 4 bit counter. The counter effectively counts the number of prescaler cycles in one reference period, T_{ref}. Let the desired frequency division range be 64 to 72. If the output frequency of the prescaler is much less than the desired frequency, the counter would count less than 8 cycles. This would cause the UP2 signal to be asserted, which in turn, causes a large amount of current to be injected into the loop filter. On the other hand, if the output frequency of the prescaler is too large then the counter counts more than 9 prescaler cycles per reference period. In this case the DN2 is asserted, which causes a large amount of current to be

drawn from the loop filter. When the output frequency of the prescaler is near the desired value, the counter contents is either eight or nine. If the prescaler and counter are considered to be the total frequency divider, the total division ratio is either 8*8=64 or 8*9=72. The counter outputs a '1' when the counter content is nine, and outputs a '0' when the counter content is eight.

Finer division resolution can be achieved by counting the number of 1's from the counter. If a resolution of 1/64 is required, then the samples from the counter are stored over a 64 sample window. This window is implemented as a shift register. The number of 1's in the shift register is equal to the desired fractional resolution. In order to sum the number of 1's, the output of the counter is simultaneously fed to the shift register as well as an accumulator. The last value popped out of the shift register is subtracted from the contents of the accumulator in order to preserve the 64 sample window. The accumulator contents are then compared to the FCW. Note that finer frequency resolution can be achieved by simply increasing the sample window size.

6.2.2 Resolution Enhanced Architecture

One disadvantage of the above architecture is that the loop filter control voltage is updated once every T_{ref} period. This limits the achievable frequency resolution for a given lock time requirement. If the charge pump current is too large, this may cause the control voltage to advance beyond the desired value before it is updated. This may cause the control voltage to oscillate around its desired value, and hence the PLL would be unstable. One way to circumvent this problem is to reduce the charge pump current. This, however, may result in an excessively large lock time. Another solution is to modify the architecture to use a bank of pipelined counters, as shown in Fig. 6.2. The charge pump update time is decreased linearly with the pipeline depth. The operating frequency of the accumulator, shift register, and the comparator also increase linearly with the pipeline depth. Therefore, the two main disadvantages of this architecture are increased area and power consumption.

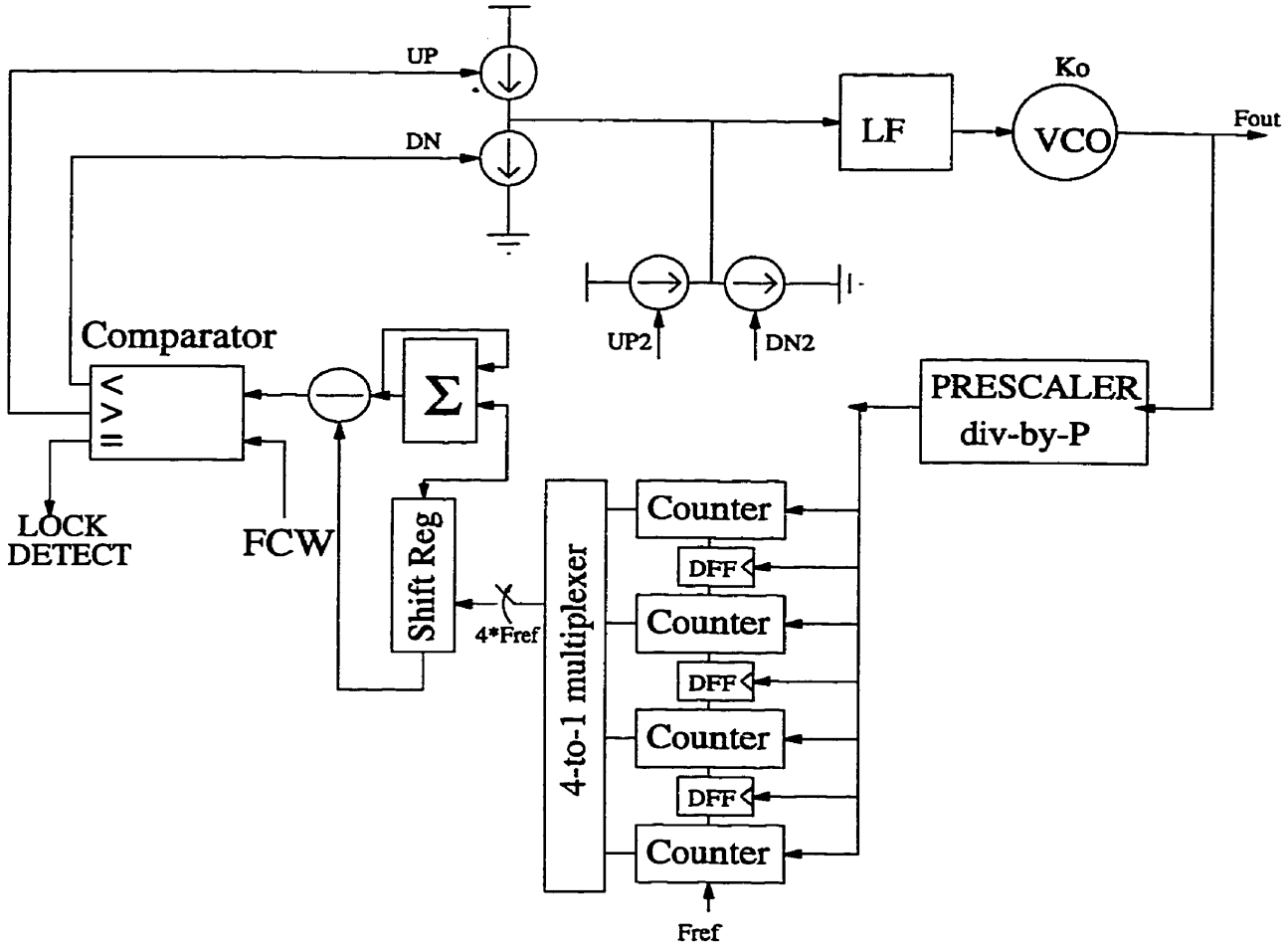


Figure 6.2. Fast Lock Digital PLL with pipelined counters

In order to quantify the increase in power consumption due to pipelining, basic size (which affects the capacitive load) and operating frequency models were derived in terms of the design parameters. The main source of power consumption has been assumed to be dynamic power. Dynamic power consumption in digital CMOS circuits is given as

$$P_{dyn} = \alpha_{0\to1} \cdot C_L \cdot V_{dd} \cdot V_{swing} \cdot f \qquad (6.1)$$

where C_L is the load capacitance, V_{dd} is the supply voltage, V_{swing} is the output logic swing, f is the operating frequency, and $\alpha_{0\to1}$ is the 0→1 data toggling at the output.

The models for the digital portions of the PLL are shown in Table 6.1.

Table 6.1. Models for the digital portions of the PLL

Component	**Size**	**Frequency**	**Power (size × Freq)**
Prescaler	$\lceil \log_2 P \rceil$	$\frac{(P+1)f_{vco}}{2P}$	$\frac{(P+1)f_{vco}}{2P} \cdot \lceil \log_2 P \rceil$
Counter	$\frac{F_l}{F_{ref} \cdot P}$	$\frac{f_{vco}}{P}$	$\frac{f_{vco} F_l}{F_{ref} P^2}$
Shift Register	$\left\lceil \frac{F_{ref} P}{F_{res}} \right\rceil$	F_{ref}	$\left\lceil \frac{F_{ref}^2 P}{F_{res}} \right\rceil$
Accumulator	$\log_2 \left\lceil \frac{F_{ref} P}{F_{res}} \right\rceil + 1$	F_{ref}	$\left(\log_2 \left\lceil \frac{F_{ref} P}{F_{res}} \right\rceil + 1 \right) \cdot F_{ref}$
Comparator	$\log_2 \left\lceil \frac{F_{ref} P}{F_{res}} \right\rceil + 1$	F_{ref}	$\left(\log_2 \left\lceil \frac{F_{ref} P}{F_{res}} \right\rceil + 1 \right) \cdot F_{ref}$

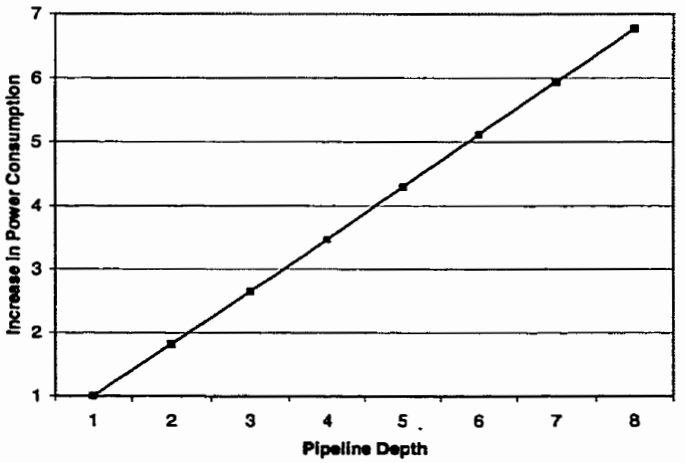


Figure 6.3. PLL's relative power increase due to pipelining

The variable F_l denotes the minimum required output frequency, P is prescaler's

division ratio, F_{ref} is the reference frequency, F_{res} is the required frequency resolution, and f_{vco} is the VCO output frequency. Using typical number of P=16, F_{ref}=10MHz, f_{vco}=2.4GHz, F_1=2.4GHz, the total increase in power consumption due to pipelining can be quantified, as shown in Fig. 6.3.

6.3 Low Resolution Fast Lock Implementation

In this section, two implementations are detailed. One is for the IEEE 802.11 wireless LAN standard and the other is for the DECT standard. Issues such as datapath width, frequency plan, and lock time are discussed.

First, the implementation IEEE 802.11 wireless LAN standard is detailed. Table 6.2 summarizes the relevant specifications for this standard.

Table 6.2. Specifications of 802.11 wireless LAN

Parameter	**Value**
Operating Frequency	2.4GHz-2.483GHz
Freq. Resolution	75KHz
Lock Time	300μs

In order to meet the operating frequency range, while still maintaining a reasonable frequency at the output of the prescaler (around 100MHz), a fixed prescaler division ratio of 16 and counter ratios of 15/16, and a reference frequency of 10MHz were selected. The counter sample window must be large enough in order to produce a frequency resolution of 75KHz. In order to meet this specification, a window of 2200 samples is required. The lock time can be estimated by multiplying the number of samples required in the window by the reference frequency. This assumes that the VCO is in a frequency range such that the value of the counter after one reference period is either 15 or 16. In practice, the lock time may be larger than this value, but not by a significant amount since the UP2 and DN2 signals quickly move the VCO into the correct frequency range. The lock time, in this case, is estimated to be 220μs, which is less than the standard requires, which is 300μs.

The last parameters to be adjusted are the charge pump currents and the loop filter parameters. Note that the frequency resolution and the lock time are independent of the loop filter parameters. This is an extremely beneficial property of the digital PLL architecture since it enables the loop filter parameters to be shrunk to very small values. The loop filter parameters are adjusted to merely ensure that the slew rate of the control voltage is exactly one fractional resolution per reference period. For a VCO gain of 50MHz/V, the slew rate of 1.25mV per reference period is required to change by one channel per reference period. Using a charge pump current of 5μA and a capacitor size of 308pF satisfies this requirement. As stated before, the loop filter is left tristated when the PLL is locked. In practice, leakage current will cause the loop filter voltage to drop. If a leakage current of 1nA is assumed, it would take 310μs for the PLL detect a loss of lock condition and to correct the voltage across the loop filter. Since this action occurs periodically, it would cause a spur at 3.225KHz away from the output frequency. Note however, that a frequency resolution of only 75KHz is required; therefore, this spur does not need to be taken into consideration. Table 6.3 shows the PLL parameters required for the IEEE 802.11 wireless LAN standard, assuming a direct conversion wireless receiver architecture (see Fig. 2.1(c)) is used.

Table 6.3. PLL parameters for the IEEE 802.11 wireless LAN standard

PLL Parameter	**Value**
Prescaler Division Ratio	16
Counter Values	15,16
Shift Register Length	2200
Bitwidth datapath	13 bits
Reference Frequency	15MHz
Lock time	220μs
Charge Pump Current, I_p	5μA
R	1KΩ
C	308pF

If a single IF or double IF receiver is used, then different sets of target output frequencies would be generated depending on frequency plan of the system used. For

example, if a single IF receiver is used with an IF frequency of 200MHz, the required LO signal from the frequency synthesizer would range from 2.6GHz to 2.683GHz.

Fig. 6.4 shows the frequency spectrum of the digital PLL implemented. As shown, the spur levels are well below –80dBc, which confirms the spectral purity of the synthesized signal.

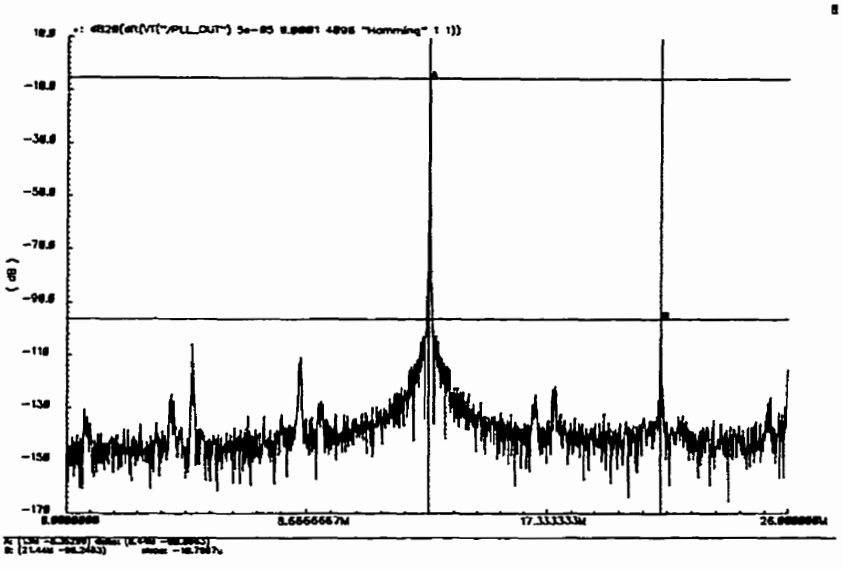


Figure 6.4. Frequency spectrum of the digital PLL

The other standard considered was the DECT cordless standard. The relevant specifications for this standard are summarized in 6.4. As shown, two lock time requirements are shown; one requirement is for the base station, while the other is for the handset.

Table 6.4. Specifications of DECT cordless standard

Parameter	**Value**
Operating Frequency	1880MHz-1900MHz
Freq. Resolution	50KHz
Lock Time (H.S.)	450μs
Lock Time (B.S.)	30μs

The same design procedure is used as before and the resulting PLL parameters are summarized in 6.5. These parameters meet the requirements for the handset, however, the lock time parameter is too large for the base station requirements. In order to meet the lock time specification of 30μs, the 8 counters were pipelined. This results in an excessive amount of power consumption. However, since this is for base station operation, the power consumption constraints are more relaxed. The resulting lock time is 20μs, which meets the base station specification. The frequency spectrum of the resulting PLL was similar to that implemented for the IEEE 802.11 wireless LAN standard.

Table 6.5. PLL parameters for the DECT cordless standard

PLL Parameter	**Value**
Prescaler Division Ratio	8
Counter Values	15,16
Shift Register Length	2400
Bitwidth datapath	13 bits
Reference Frequency	15MHz
Lock time	160μs
Charge Pump Current, I_p	5μA
R	1KΩ
C	257pF

6.4 High Resolution Fast Lock Implementation

6.4.1 Direct Implementation

For higher resolution standards the design parameters of the digital PLL architecture become excessively large. To demonstrate this, the GSM standard is used as an example. The relevant specifications for this standard are summarized in Table 6.6.

Table 6.6. Specifications of GSM wireless standard

Parameter	**Value**
Operating Frequency	880MHz-915MHz
Freq. Resolution	50Hz

Lock Time (voice)	577μs
Lock Time (GPRS)	200μs

Using the same design procedures as before, the resulting PLL design parameters are shown in Table 6.7. As the table shows, the shift register length is excessively large, creating a very long lock time. Even if the PLL reference frequency is doubled and the prescaler is eliminated, a pipelining of 18 counters is needed to meet the GSM voice wireless standard, and a pipelining of 50 counters is required to meet the GPRS wireless data standard's lock time specification. This shows that using this architecture as is, is not practical for high resolution wireless standards.

Table 6.7. PLL parameters for the GSM wireless standard

PLL Parameter	**Value**
Prescaler Division Ratio	8
Counter Values	8, 9
Shift Register Length	2,080,000
Bitwidth datapath	22 bits
Reference Frequency	13MHz
Lock time	160msec
Charge Pump Current, I_p	5μA
R	1KΩ
C	257pF

6.4.2 High Resolution Fast Lock Architecture

An alternative use of this architecture is to use it as an aid for fast frequency lock. Fig. 6.5 shows how this can be used. As the figure shows, a conventional PLL architecture is used along with the digital fast lock architecture integrated into it as a second loop. The output of the VCO goes into both the conventional PLL's prescaler, as well as the fixed prescaler of the digital fast lock PLL. The output of the digital fast lock PLL is merged with that of the PFD to drive the charge pump. The "LOCK DETECT" signal from the digital fast lock PLL is used to switch from the digital PLL back to the regular

PLL once the digital PLL has locked within its required resolution. The digital PLL is then powered down, and the regular PLL completes the lock procedure.

Simulations have shown that even with only 1KHz accuracy, the PLL may still take a long time to lock depending on phase difference between the feedback, V, signal and the reference, R, signal. This is due to the phenomenon discussed in chapter 3. In order to help reduce the phase lock time, counters in the frequency divider are reset so as to attempt to align the R and V phases together once frequency lock is achieved. Mismatches in phase can still occur as a result in delay from the counters in the divider to the PFD. Matching the delays less than 1ns is difficult in practice; therefore, a mismatch of 1ns was assumed.

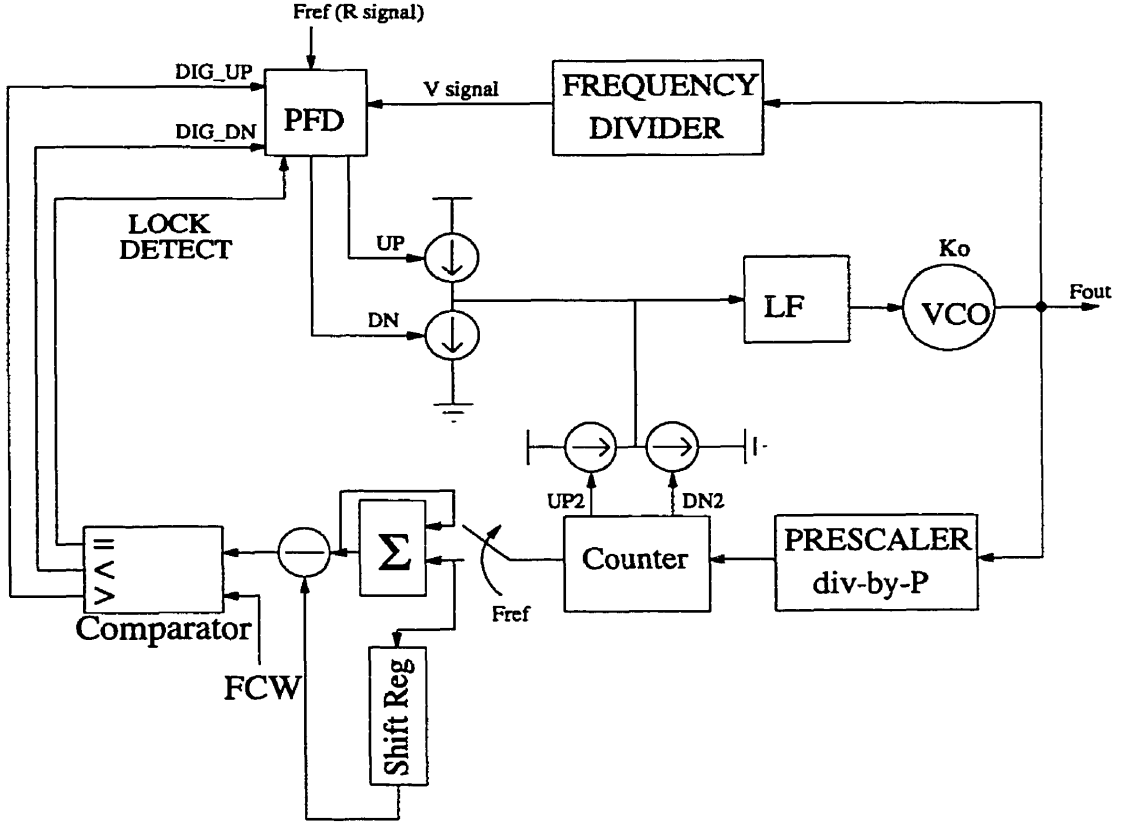


Figure 6.5. Proposed fast lock fine resolution PLL architecture

Once again, GSM, is used as a design example. Both the digital fast lock technique

and the phase resetting technique were used. The digital PLL was optimized for a frequency accuracy of 40KHz. A 2600 bit shift register is required, along with a 12-bit datapath. Eight pipelined counters were required to reduce the frequency lock time down to 25μs. For a VCO gain of 40MHz/V, this puts the control voltage at 1mV away from the desired value. When resetting the phases to within 1ns, the phase lock time requires an additional 83μs. Without the phase resetting technique, the phase lock time is as high as 127μs. This puts the total lock time to a little under 110μs, which is well below the required 200μs for the GSM or even the GPRS standard. Fig. 6.6 shows the reduction of lock time using the various above mentioned techniques.

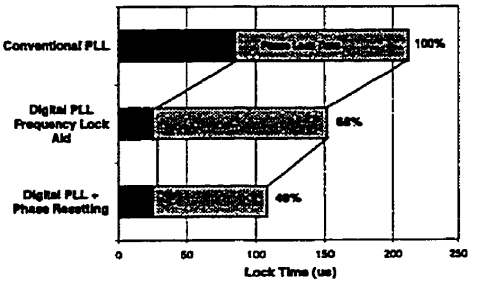


Figure 6.6. Lock time speed-up using fast frequency and phase locking techniques

6.5 Performance Evaluation

In order to quantitatively evaluate the performance of the PLL described above, a figure-of-merit (FOM) suitable for fast lock PLLs is given. The fast lock PLL

architecture reported is then compared to the other fast lock PLL techniques described in chapter 2.

The proposed figure-of-merit for evaluating fast lock PLLs for wireless applications is given as

$$FOM = \left(\underbrace{C_1(nF) + A_{digital}(mm^2)}_{AREA} \right) \cdot P \cdot T_L \tag{6.2}$$

where C_1 is the largest capacitor in the loop filter, $A_{digital}$ is the area of the digital circuitry of the PLL, P is the power consumption of the PLL, and T_L is the total lock time. Note that these measures include all components of the PLL except the VCO. Table 6.7, below, shows the area, power, and lock time estimates of the fast lock PLL architectures detailed in chapter 2 with respect to a conventional fractional-N PLL. All PLL implementations were tuned for the GSM application (although not all were able to meet the lock time specifications). As the table shows, the proposed PLL has the best FOM figure.

Table 6.8. Comparison of this work with other fast lock PLLs

Architecture	Area		Power	Lock Time	FOM
	C_1	$A_{digital}$			
Frac-N PLL [34]	1	1	1	1	1
Two-mode [11]	1	1.33	1	0.333	0.344
DSP PLL [6]	0	44	1.2	1.25	6
Dual Loop [7]	1.5	2	3	0.333	1.545
Feed Forward [9]	1	1	1	2.5	2.5
FDD PLL [13]	1	2	1.5	0.238	0.454
Proposed	1	3	1.5	0.161	0.285

6.6 Conclusions

In this chapter a simple, yet effective method of reducing the lock time has been demonstrated. It has been shown that the lock time and frequency resolution trade-off can be performed in the digital domain, as opposed to the analog domain. The

advantage of this is that in the digital domain, the trade-off deals with the area and power penalties incurred for extra performance gains. The PLL's lock time and frequency resolution are, therefore, independent of the loop filter paramters, which enables a drastic reduction in loop filter sizes. This enables a fully integrated solution. The digital PLL architecture has been shown to be effective for low frequency resolution standards such as cordless and wireless LAN standards. For high frequency resolution standards, such as GSM, the architecture can only be used as a frequency aid circuit. The cost of fine resolution becomes excessively high in terms of area penalty as well as lock time. A simple, yet effective means of reducing the phase lock time is achieved by resetting the counters in the frequency divider at the first positive edge of the reference input after the digital PLL has acquired frequency lock to within its resolution.

Chapter 7

Low-Power, High-Performance Digital Arithmetic

7.1. Introduction

In the previous chapters, it has been shown how to leverage digital technology to enhance the performance and facilitate high level integration of phase locked-loops. In Chapter 5, a large digital block (the modified sigma-delta modulator) is used to randomize spurs generated by fractional-N PLLs and also to widen the closed loop bandwidth of the PLL. This large digital block operates at a relatively low frequency (~10MHz). In Chapter 6, a high-speed digital block is used to obtain a digital

representation of the output frequency. This estimate is then compared to the desired output frequency (also a digital word).

In this chapter, new dynamic and static logic families are proposed. These logic families can be used to reduce the power consumption of the PLL architectures reported in chapters 5 and 6. First, the logic families are described in detail. This is followed by an evaluation of how much of the PLL's power is reduced in comparison with a standard cell implementation.

7.1.1 Introduction of Low-Power Design

Perhaps one of the most pervasive trends in advanced digital circuit design is low-power design. Low-power design is especially important in portable applications in order to prolong battery lifetime. Reducing power consumption also helps in reducing heat dissipation in microchips. This simplifies chip heat removal and hence reduces packaging costs. Power consumption in any digital design is dominated by dynamic power consumption, which is given as

$$P_{dyn} = \alpha_{0\to1} \cdot C_L \cdot V_{dd} \cdot V_{swing} \cdot f \tag{7.1}$$

where C_L is the load capacitance, V_{dd} is the supply voltage, V_{swing} is the output logic swing, f is the operating frequency, and $\alpha_{0\to1}$ is the $0\to1$ data toggling at the output. One important method of improving the performance of a system is to use dynamic logic. Dynamic logic, however, generally suffers from high susceptibility to noise and high power consumption.

For moderate performance applications, a static logic family may be used. In these types of applications, minimizing leakage power is often more important than dynamic power. One effective method of minimizing the leakage power is to minimize the supply voltage. In doing so, however, the performance of the circuit techniques must be guaranteed at very low supply voltages.

7.2 Dynamic Differential Logic Families

In this section, a new class of dynamic differential logic, swing limited logic (SLL), is presented [63]. It is shown that this class of circuits provides very high performance at power-delay products an order of magnitude less than conventional circuit techniques at nominal supply voltages. First, the choice of dynamic differential logic is justified. The requirements for reliable operation of SLL are also detailed. This follows a detailed description of two implementations of SLL, Short-Circuit Current Logic (SC^2L) and Clock-Pulse Controlled Logic (CPCL). The performance of SLL is supported by experimental results.

7.2.1 Swing Limited Dynamic Differential logic

7.2.1.1 Conventional Dynamic Logic

Dynamic logic is usually preferred over static logic in high-performance applications due to its speed advantage [51]. There are two classes of dynamic logic: single-ended and differential dynamic logic. A simple single-ended dynamic gate is shown in Fig. 7.1(a). During evaluation, if all inputs are low, internal node, X, is left floating at logic high. The voltage at this node may drop due to leakage current or crosstalk noise from adjacent lines. If the voltage on this node becomes too low, it can cause incorrect switching of the output inverter. One solution to this problem is to insert a feedback keeper device, MP2, shown in Fig. 7.1(a). This, however, comes at the expense of reduced performance.

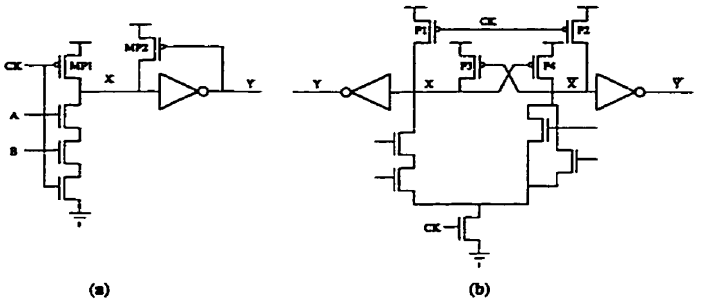


Figure 7.1. Dynamic Logic (a) single-ended, (b) differential

A basic dynamic differential gate is shown in Fig. 7.1(b). One advantage of this configuration is that it overcomes the non-inverting characteristic of single-ended dynamic logic. Cross-coupled PMOS transistors also help noise immunity. Consider the case when Y is logic low, this means that X is logic high. Unlike the single-ended case, this node is not floating, since P3 is on. This added noise immunity comes at no extra delay cost.

One disadvantage of dynamic logic, however, is that it exhibits higher switching activity (α in Equation 1) than static logic. Furthermore, dynamic differential logic (DDL) has a switching activity of 1, rendering it incapable of exploiting signal correlation to decrease power consumption. This may render DDL to be more power consuming than static CMOS for digital signal processing (DSP) applications. For this reason, low-power design of dynamic differential logic is essential to keep it a viable circuit technique for high-performance and low-power requirements.

7.2.1.2 Noise Analysis of Swing Limited Dynamic Logic

One method of satisfying these two conflicting requirements is to reduce the logic swing of dynamic differential logic. This however, reduces the noise margin to an already noise susceptible circuit. For this reason, a method of quantifying the minimum reliable logic swing must be developed.

For equally loaded differential outputs, the two strongest factors in varying the output voltage are leakage current and crosstalk. The effect of leakage current may be reduced by using cross-coupled PMOS transistors P3 and P4, as shown in Fig. 7.1(b). Another method of reducing the leakage current is to increase channel length of the pull-down NMOS transistors. This, however, comes at the expense of reducing the performance of the gate.

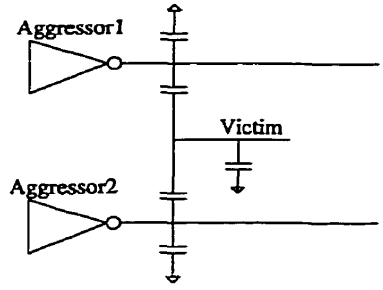


Figure 7.2. Configuration for crosstalk model

To evaluate the effect of crosstalk, a 0.35μm CMOS technology and a simple configuration shown in Fig. 7.2 have been assumed. In this configuration, there are two aggressor nodes adjacent to one victim node. It is assumed that both aggressors have the same driving strength and that the victim is floating.

There are several factors that affect crosstalk [54]. The most important factors, for this study, are

- voltage swing of the aggressor,
- aggressor net adjacency, and
- victim to aggressor net spacing.

Fig. 7.3 illustrates the effect of varying both the voltage swing and the net spacing between the victim and the aggressor. The nets have been assumed to be adjacent for 200μm, which is a fair assumption for local interconnects.

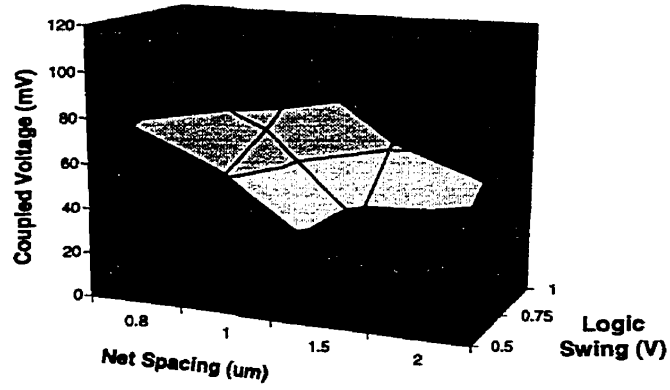


Figure 7.3. Coupled voltage dependence on voltage swing and net spacing

In SLL design, it is especially important that differential signals have common mode rejection of noise. A simple way to achieve this is to route both signals together so that if one line picks up any noise, there would be a high probability that its complementary wire would also pick up the same noise. The coupling capacitance between the two lines, however, can cause slow switching on the differential signals, since they switch in opposite directions of one another. This delay also depends on the net spacing between the differential wires and the voltage swing on the wires. Fig. 7.4 illustrates the effect of varying both the voltage swing and net spacing between the differential wires on the delay of the differential signal.

To decide an optimum point for the logic swing, a weighted function which takes into account both the normalized delay variation and the coupled voltage can be expressed as

$$F = \Delta V_{coupled} + t_{d,normalized} \tag{7.2}$$

Minimizing this equation, gives the optimum point at a voltage swing of 0.7V. Further increase in voltage swing may be necessary in the case of longer wires or taking into account supply noise.

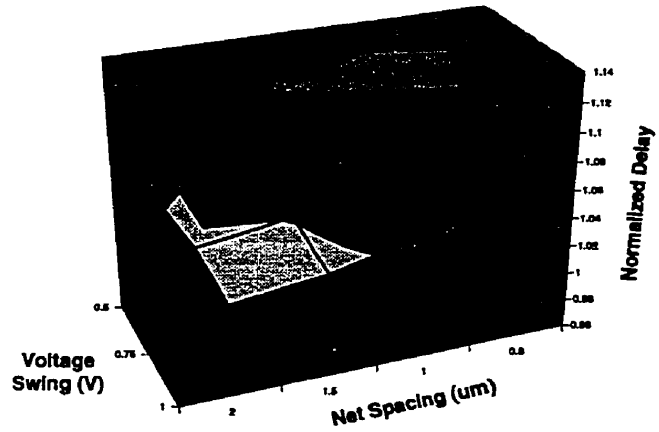


Figure 7.4. Normalized delay dependence on voltage swing and net spacing of differential signals

7.2.2. Short-Circuit Current Logic

7.2.2.1 Basic Operation

The first proposed dynamic differential logic family, called Short-Circuit Current Logic (SC^2L) [55,67] is shown in Fig. 7.5. Its basic operation is now explained. When CK is high, the CMOS inverter at the bottom (MP1, MN1) discharges node B to 0. This turns on MP2 and MP3. MP4 and MP5 act as pass gates and are turned off for this duration of the clock pulse. The circuit is effectively in precharge mode. During the evaluation mode, the CK input starts to go low. For a brief amount of time, both MP1 and MN1 are turned on. First, the pass gate PMOS transistors are enabled. Secondly, the CMOS inverter dissipates short-circuit current. This discharges one output through the N-logic tree according to the input logic. The amount of voltage drop across the output is determined by the amount of short-circuit current dissipation, which in turn, depends on the W/L ratio of the pull-down inverter as well as the clock slew rate [53].

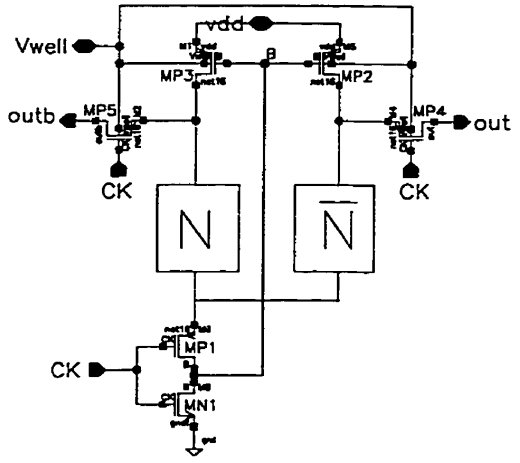


Figure 7.5. Short-Circuit Current Logic (SC^2L) Gate

One important point to note is that the maximum voltage at node B is V_{DD}-$V_{TN,body}$. This means that the body terminals of MP2 and MP3 need to be biased to a boosted supply voltage to effectively turn them off and the clock swing can be reduced to V_{DD}-$V_{TN,body}$. MP4 and MP5 now must have their body terminals biased to V_{well} since they are driven by a reduced swing clock signal. Reduced clock swing helps in reducing energy consumption and effects of clock feedthrough effect[53], which are serious in reduced swing dynamic logic.

7.2.2.3 Modified SC^2L Gate

The basic SC^2L gate suffers from variation of voltage swing in the presence of variable output loads. One solution entails placing a diode from VDD to the output nodes in order to limit the voltage swing. Although turn on time of the diode is slow, it can be used to hold the output voltages to fixed values at periods of low activity. Fixing the voltage swing also permits the use of charge sharing as in Charge Recycling Differential Logic (CRDL) [52]. This results in a new gate, Quasi-static SC^2L (QSC^2L), shown in Fig. 7.6.

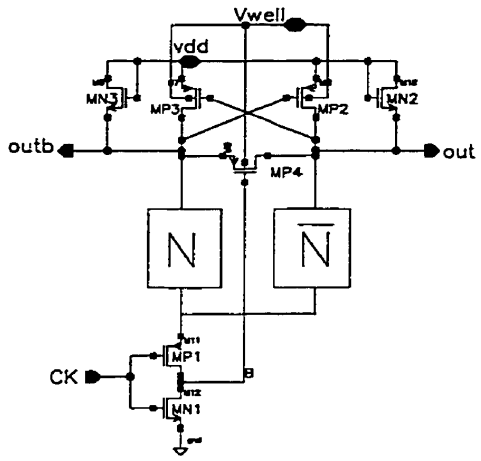


Figure 7.6. Quasi-static SC^2L (QSC^2L) Gate

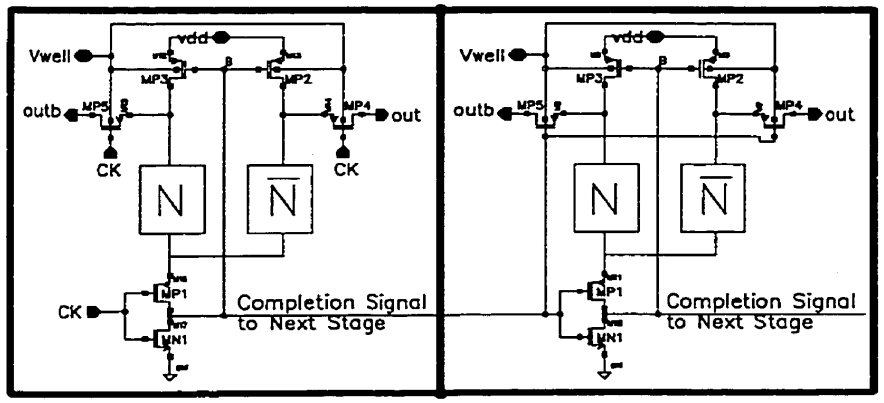


Figure 7.7. SC^2L Asynchronous Pipelining Strategy

7.2.2.4 SC^2L Gate Pipelining

One important issue in high-speed dynamic logic design is that of clocking. Distributing the clock signal to every single gate significantly increases the system's clock power. Fig. 7.7 demonstrates an efficient way to pipeline SC^2L gates. In SC^2L, the clock is applied only to the first pipeline stage. Subsequent stages receive a completion signal. The completion signal is generated for free. Since node B represents the inversion of the clock signal and it does not fully switch until the output

has completely settled, it may be used as a completion signal for the next stage. The disadvantage of this scheme is that it places a larger load on node B, which may slow down circuit operation.

7.2.2.5 SC^{2}L Simulation Results

In this section, SC^{2}L is evaluated in terms of its delay, power, robustness and is compared to other logic families. All gates were implemented in a 0.35μm CMOS technology.

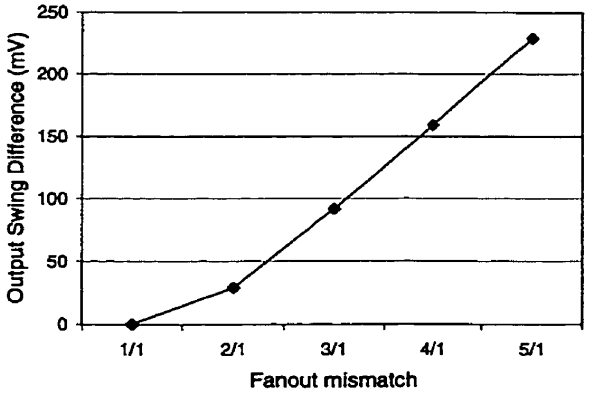


Figure 7.8. Output voltage swing sensitivity to load mismatch

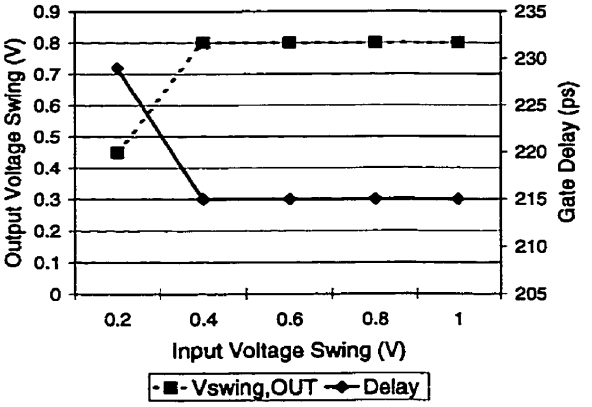


Figure 7.9. Output voltage swing vs. input voltage swing

Since logic swing is limited by restricting the evaluation time, the output voltage swings may vary according to loading conditions. A SC^2L inverter/buffer was used for this characterization with nominal operation conditions of V_{dd}=3.3V, f_{CLK}=500MHz, V_{swing}=0.8V. The output voltage swing sensitivity to fan-out mismatch is shown in Fig. 7.8. Another important characterization, the output voltage swing and gate's delay sensitivity to input voltage swing, is shown in Fig. 7.9. As the figure shows, an input voltage swing between 0.4V to 1V causes very little change in the gate's performance. This shows that the gate has good regenerative properties as long as noise injected into the output lines is less than 400mV (0.8V-0.4V).

A full-adder (FA) is used as a benchmark circuit to compare SC^2L to other logic styles. Using a fanout of 3 and V_{dd}=3.3V, the results shown in Table 7.1 have been obtained. As the table reveals, QSC^2L has 37% less energy than SC^2L, due to charge recycling. SC^2L and QSC^2L are compared to other logic families (normalized to 100MHz) in Fig. 7.10. This figure shows that both SC^2L and QSC^2L exhibit an order of magnitude less power-delay product than other logic families.

Table 7.1. Results of FA implementation in SC^2L and QSC^2L

	SC^2L (FA)	**QSC^2L (FA)**
Logic Swing	0.7V	0.7V
Frequency	900MHz	900MHz
Power	0.106mW	0.062mW
Energy (uW/MHz)	0.118pJ	0.069pJ
Power-Delay Product	29.15fJ	18.14fJ

7.2.2.6 SC^2L Experimental Results

Fig. 7.11 shows a chip micrograph of two 8-bit asynchronous pipelined SC^2L-based CRAs. Both CRA's were optimized for 0.7V logic swing and 700MHz operation. In order to ease testing, an on-chip oscillator (CKGEN) and output frequency dividers

(DIV2) have been used. Current-mode logic (CML) has been used in the DIV2 circuits to support low-swing inputs. The output's signal (SUM) frequency is divided by 4 to ease testing.

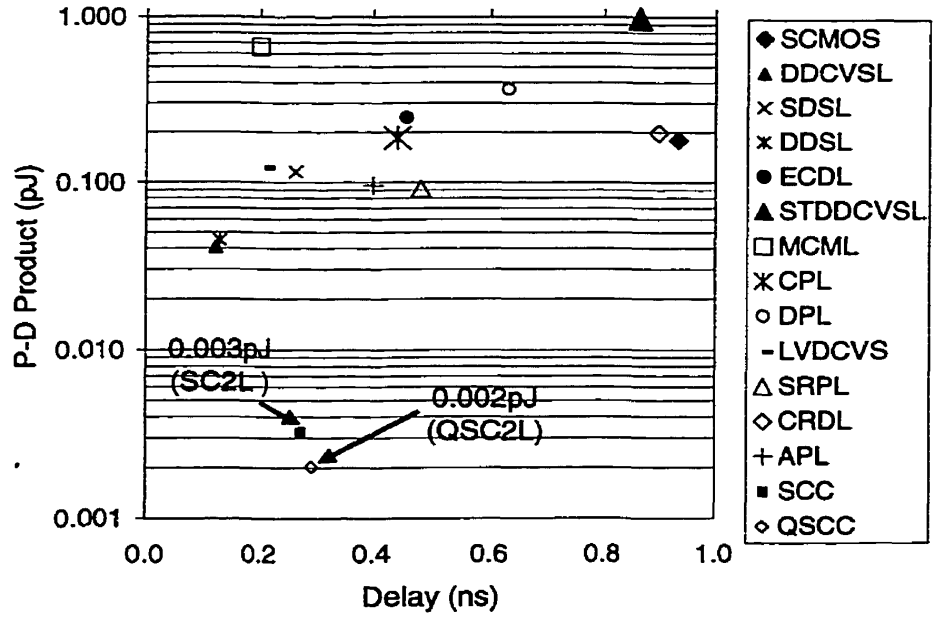


Figure 7.10. Comparison of SC^2L with other logic families [56]

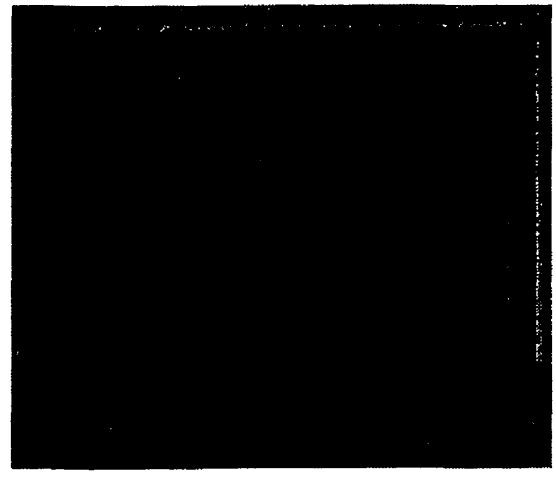


Figure 7.11. Chip micrograph of two SC^2L carry ripple adders

In order to test the functionality of the pipelined CRA, an input vector of A[7:0]=1,

B[7:0]=0, and C_{in} alternates between 1 and 0 at one-half the clock frequency, which means that the internal clock rate is running at 8x the output signals' rate. The measured output of the adder is shown in Fig. 7.12. Since the waveforms display a 90.9MHz operating frequency, the internal clock rate is 727MHz. The measured delay between Sum[7] and Sum[3] is 800ps, which translates to an internal delay of 67ps per FA. The results of the simulated and the measured SC^2L FA implementations are compared in Table 2.

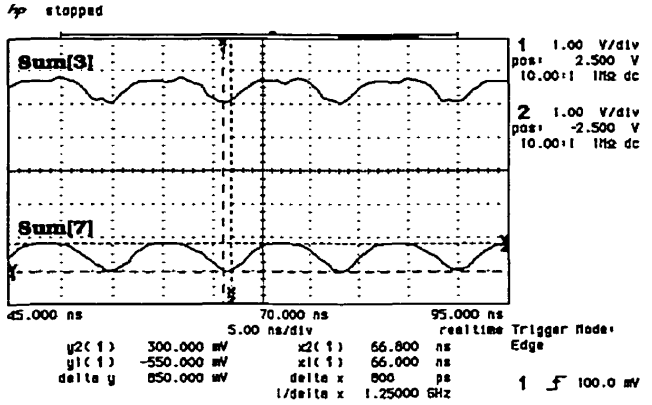


Figure 7.12. Waveform output of test results from 8bit CRA (Sum[3] and Sum[7] output bits)

Table 7.2. Simulated & measured results of FA SC^2L

SC^2L characteristics	**Simulated**	**Measured**
Logic Swing	0.7V	0.85V
Frequency	700MHz	727MHz
Delay	80ps	67ps
Power	0.275mW	0.293mW
Energy (uW/MHz)	0.304pJ	0.403pJ
Power-Delay Product	22fJ	19.6fJ

7.2.3 CPCL Logic Family

7.2.3.1 Basic Operation

One weak point of SC^2L is that it has low driving capability due to the short amount of time the short-circuit current is available. Another swing limited dynamic differential logic family, Clock Pulse Control Logic (CPCL), is proposed to solve this difficulty [63]. Fig. 7.13 shows the CPCL gate. This circuit creates a pulse from the CK signal by a high-pass passive RC filter, effectively differentiating the clock signal. When the CK signal is low, the circuit is in the precharge phase. When the CK signal goes high, the coupling capacitor, C, allows a certain amount of charge to enter node A. This charge is trapped at that node when the CK signal finishes switching to high. Transistor MN2 acts as a resistor. Effectively, MN2 and C act as a passive high-pass RC filter. The RC time constant may be adjusted according to the desired load drive and voltage swing.

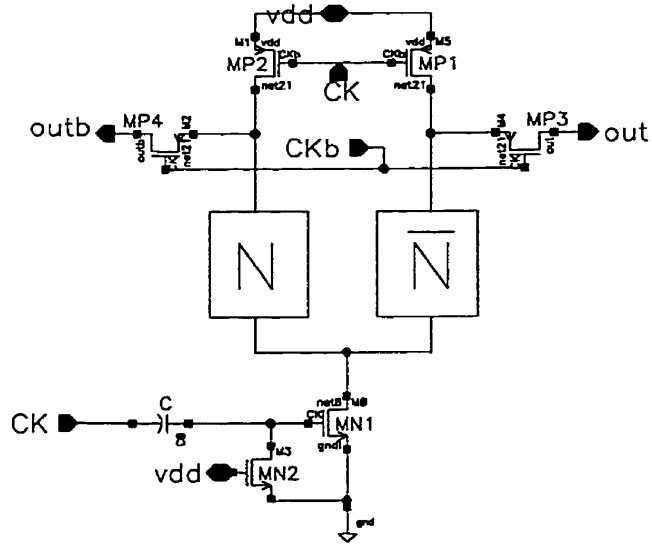


Figure 7.13. Basic CPCL Gate

Unlike SC^2L, CPCL requires both the clock signal and its complement. The implementation of coupling capacitor, C, in a digital CMOS technology entails the use

of a separate N-Well. This incurs heavy area penalties. Since a separate coupling capacitor is required per gate, it is not practical to use CPCL to implement large logic blocks. It is, however, useful in implementing bus drivers and high fanout gates, since its driving capability far exceeds that of SC^2L.

7.2.3.2 CPCL Characterization

CPCL is now evaluated in terms of its driving capability and its sensitivity to load mismatches. In all cases, nominal conditions of V_{dd}=3.3V, f_{CLK}=500MHz, voltage swing = 1V are used. Fig. 7.14 shows the output voltage swing sensitivity to load mismatch.

One of the main reasons for using CPCL is its superior driving capability (as compared to SC^2L). Fig. 7.16 shows a CPCL inverter's delay as a function of load capacitance for a constant voltage swing. The delay of a static CMOS (SCMOS) inverter is shown in the figure for purpose of comparison. The CPCL curve exhibits lower slope due to its reduced logic swing.

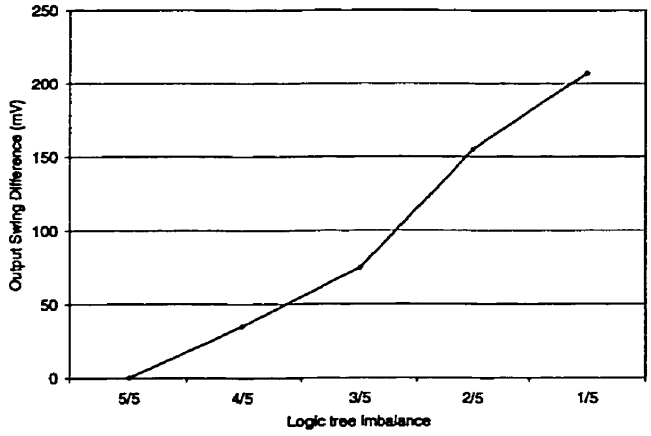


Figure 7.14. Output voltage swing sensitivity to load mismatch for CPCL logic

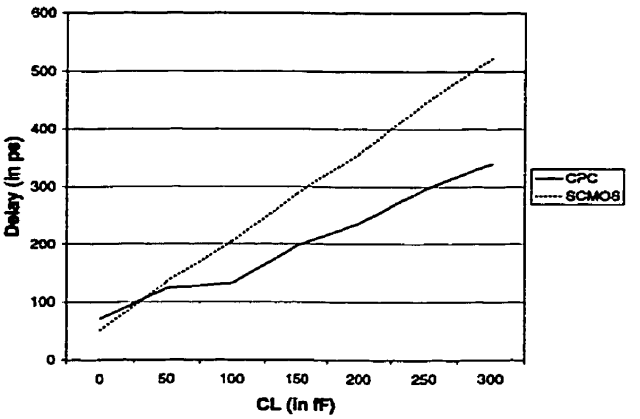


Figure 7.15. Delay versus load capacitance for CPCL logic

7.3 Differential Static Logic Family

In this section, a low-voltage high-performance static logic family is characterized and evaluated. This logic family is to be used as a building block for implementing the logic portion of the ΣΔ digital modulator.

7.3.1 LVDSL Family Operating Principle

The basic Low-Voltage Differential Static Logic (LVDSL) gate [64,65], shown in Fig. 7.16, is based on the DCVS logic family. This circuit will be referred to as LVDSL-1. The operation of the gate is best described through an example. Suppose initially OUT=1, $\overline{OUT}$=0, IN=1, and $\overline{IN}$=0. When the inputs toggle (IN=1→0 and $\overline{IN}$=0→1), node 1 will be pulled down to 0V through MN4. This would turn on pull-up transistor MP1 and thus charging $\overline{OUT}$ to VDD. This, in turn, would turn on transistor MN1, which would act as a diode. Node 2 would now be pulled up to VDD-$V_{T,MN1}$ and thus turning off MP2. Now that MP2 is turned off the OUT node is allowed to be pulled down through MN2 and MN4; however, since MN2 also acts as a diode, the OUT node is only pulled down to $V_{T,MN2}$. Also note that since the low voltage level is always a threshold drop above zero volts, the gate input to MN3 will be at V_T.

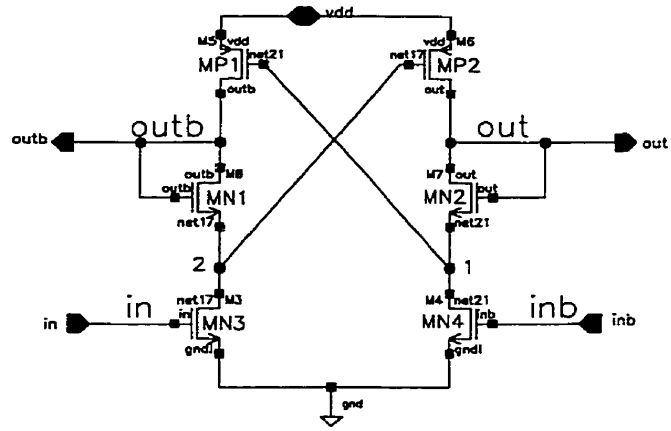


Figure 7.16. A schematic of a LVDSL-1 inverter/buffer

Analysis of this circuit indicates that its operation can outperform that of DCVS for two reasons. Firstly, as in differential split-level logic (DSL) [71], transistors MN1 and MN2 help isolate the pull-up devices from the NMOS differential tree and hence the pull-up devices may be sized up to enhance speed. Secondly, internal nodes 1 and 2 never exceed VDD-V_T, which means that the evaluation transistors MN3 and MN4 are always gate overdriven (operate in the linear region) [59]. Transistors operating in this region switch faster since they do not have to first enter saturation region.

The circuit, however, does suffer from the drawback of subthreshold leakage current. Close examination of the circuit indicates either transistors MN3, MP2 or transistors MN4, MP1 are weakly turned on. In either case, this causes leakage current in both branches of the differential logic tree, which is significant for nominal supply voltages. One effective means of minimizing the leakage current is to aggressively scale the supply voltage. The leakage current can be further eliminated by using power down circuitry as shown in Fig. 7.17. This circuit is referred to as LVDSL-2. Transistors MN5 – MN7 eliminate leakage current due to weakly turned on NMOS devices. When the SLEEP signal is enabled, cross-coupled transistors MN5 and MN6 pull down the low-level signal (either OUT or $\overline{OUT}$) from V_T to 0V. Differential mode of operation is assured by transistor MN7. Leakage currents due to MP1 or MP2

are reduced by applying a high bias voltage (greater than VDD) to the body terminal of these devices. This has the effect of increasing the effective threshold voltage of these transistors and thus biasing the PMOS transistors closer to the cut-off mode of operation [59].

LVDSL-1 and LVDSL-2 are now evaluated in terms of delay, power, energy, and leakage current. They are also compared to other static logic families. All gates were implemented in a 0.35μm CMOS technology.

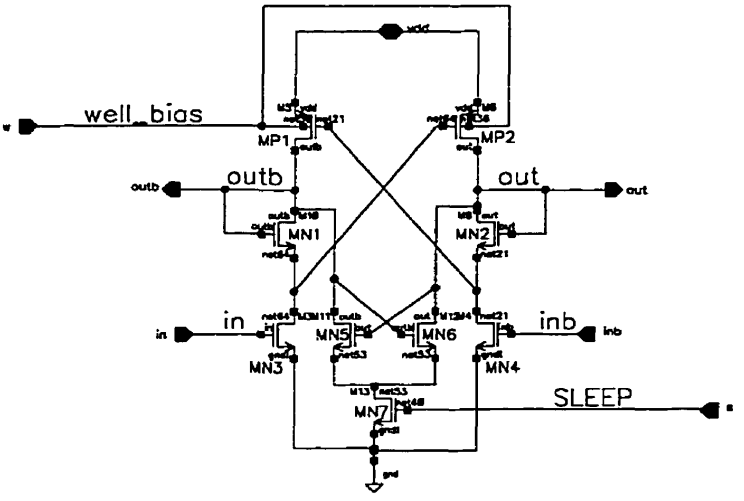


Figure 7.17. A Schematic of a LVDSL-2 inverter/buffer

7.3.2 LVDSL Characterization

First, a LVDSL-1 inverter/buffer is characterized and compared to complementary CMOS inverter. Fig. 7.18 shows how the power-delay product scales with supply voltage. These measurements have been taken at 200MHz and the driving load was a fanout of one. The figure shows two interesting facts. LVDSL-1 was functional at a supply voltage of only 0.8V, while a conventional CMOS inverter failed to operate at this low voltage. Another point to note is the sharp rise in energy (power-delay product) consumption for voltages over 1.5V. These two facts restrict the feasible operating supply voltage for LVDSL between 0.8V and 1.5V.

LVDSL-1 is now characterized in terms of its driving capability. Once again, complementary CMOS has been used as a reference circuit. Fig. 7.19 shows the driving capabilities of LVDSL-1 and complementary CMOS for loads ranging from 0fF to 100fF. These measurements were taken at an operating frequency of 200MHz @ VDD=1V. For loads less than 20fF, complementary CMOS outperforms LVDSL-1; however, for higher loads, LVDSL-1 clearly outperforms complementary CMOS in terms of delay and power consumption. This is due to the fact that in LVDSL, the pull-up PMOS's are isolated from the NMOS tree by the diode transistors, and hence pull-up and pull-down operations may be performed more efficiently.

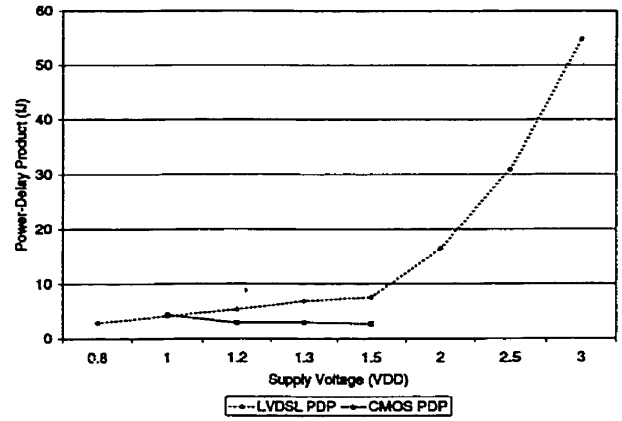


Figure 7.18. Energy (power-delay product) of a LVDSL-1 inverter/buffer

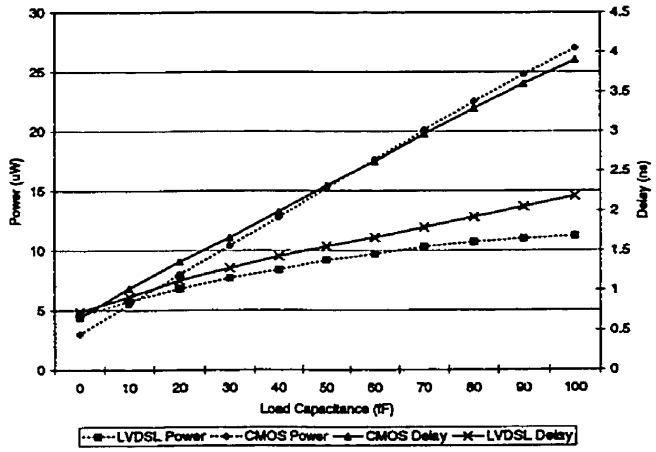


Figure 7.19. Power and Delay of LVDSL-1 versus output capacitance

The leakage current in LVDSL-1 is also characterized as a function of supply voltage and shown in Fig. 7.20. Four different cases are illustrated. Leakage current of LVDSL-1 is graphed. Two variations of LVDSL-1 are shown; the body terminals of MP1 and MP2 are biased to two different boosted voltages as in LVDSL-2; however, the pull-down circuitry (MN5-MN7) of LVDSL-2 has been omitted to investigate the effectiveness of body biasing alone. Lastly, the leakage current in a CMOS inverter is provided as a reference. Clearly, body biasing alone does help, but it seems there is an upper limit to the reduction in leakage current. Since this upper limit is far from the ideal (CMOS inverter), the effectiveness of LVDSL-2 must be investigated.

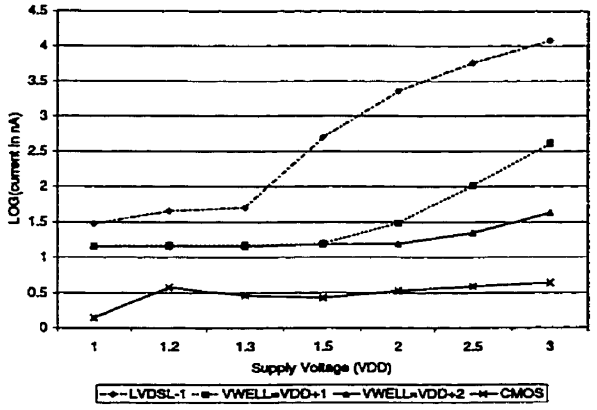


Figure 7.20. Leakage current of LVDSL-1 inverter/buffer

LVDSL-2 is first characterized in terms of leakage current in order to determine its effectiveness. This is shown in Fig. 7.21. Clearly, LVDSL-2 fulfills its task of reducing the leakage current to comparable levels with respect to a CMOS inverter. Fig. 7.21 also reveals that biasing the body terminals of MP1 and MP2 beyond 1V above VDD does not yield much improvement in leakage current reduction for supply voltages less than 2V. Fig. 7.22 compares LVDSL-1 and LVDSL-2 in terms of dynamic power and delay. Clearly, the overhead circuitry associated with LVDSL-2 has minimal impact on its performance.

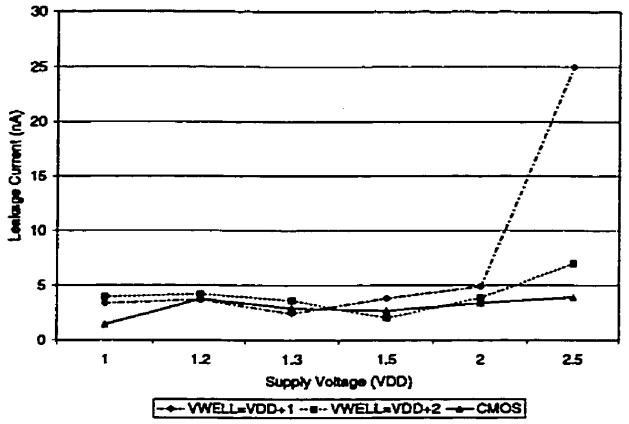


Figure 7.21. Leakage current of LVDSL-2 inverter/buffer

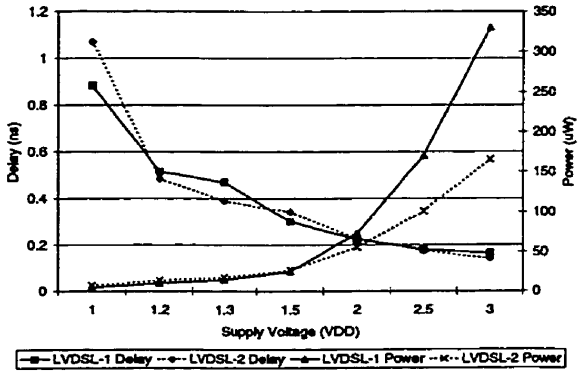


Figure 7.22. Power and Delay of LVDSL-1 and LVDSL-2

7.3.3 LVDSL Evaluation

In order to evaluate the effectiveness of LVDSL it is necessary to compare it with currently used logic styles. Such logic styles include complementary CMOS, DPL, SRPL, CPL, and DCVS. A full-adder implementation is used to compare the different logic styles. The logic tree used for the full-adder is shown in Fig. 7.23. This tree was used in LVDSL, SRPL, CPL, and DCVS. The logic configuration used for DPL and CMOS can be found in [60] and [59], respectively. A fanout of 1, uniformly

distributed input vectors, and a clock frequency of 200MHz are assumed. For a supply voltage of 1.2V, Table 5.4 shows the delay, power, and energy of various full-adder implementation normalized with respect to LVDSL-2. Clearly, for its operating range, LVDSL performs the best in terms of delay, power, and energy.

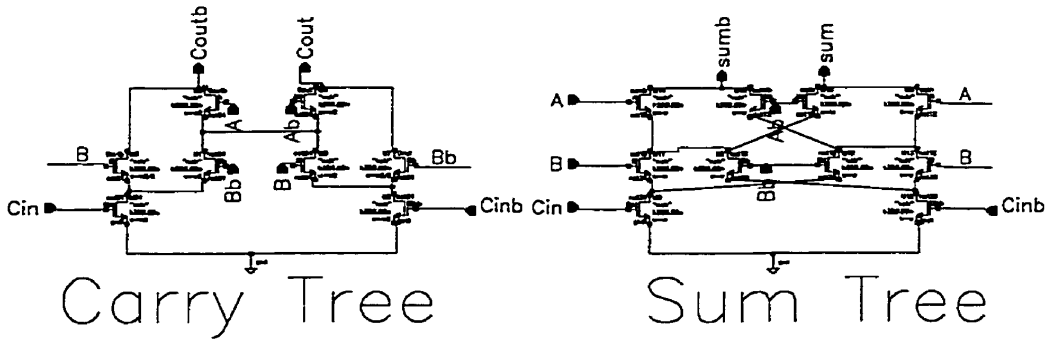


Figure 7.23. Full-adder differential logic tree

Table 7.3. Full-adder Performance at V_{DD}=1.2V Normalized to LVDSL2

	LVDSL2	CMOS	DPL	SRPL	CPL	DCVS
Delay	1	5.71	3.03	2.19	3.11	5
Power	1	3.62	0.96	0.90	0.76	1.30
Energy	1	20.70	2.91	1.97	2.36	6.52

LVDSL is also evaluated in terms of area. Fig. 7.24 shows the layout two D-type flip-flops (DFFs). Fig. 7.24(a) shows the layout of a DFF implemented in conventional CMOS, while Fig. 724(b) shows the layout of a DFF implemented in LVDSL. As the figure reveals, the LVDSL DFF occupies 10% less area than a conventional DFF, although LVDSL requires differential inputs and outputs.

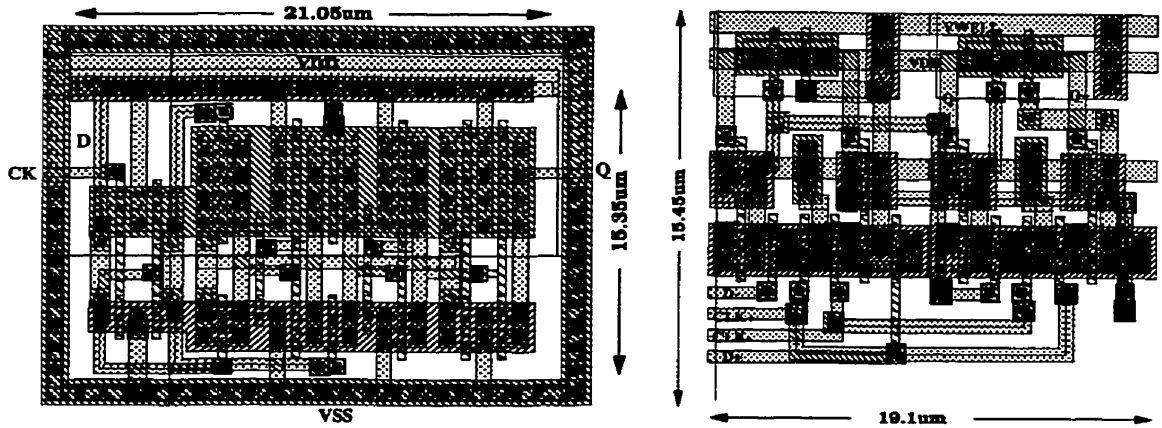


Figure 7.24. D-type flip-flop implemented in (a) conventional CMOS and (b) LVDSL

7.4 Low-Power Digital Logic in PLL Architectures

7.4.1 Fast Lock ΣΔ PLL Implementation

LVDSL has been used to implement the digital ΣΔ as well as the digital control circuitry in the ΣΔ PLL architecture proposed in chapter 5. Fig. 7.25 below compares the LVDSL implementation with the conventional CMOS implementation. The delay on both the LVDSL and the conventional CMOS were set to be equal. As the figure reveals, LVDSL is capable of reducing the power product of the digital modulator by an additional 35% even if pipelining and voltage scaling are used in the conventional CMOS implementation. Since the delays are set to be equal and both operate at the same frequency (13MHz), it can be said that the LVDSL implementation exhibits 35% less energy-delay product than the conventional CMOS implementation. Note that LVDSL exhibits power savings despite the fact that differential signals are used, which means that the switching activity of the LVDSL implementation is double that of the conventional CMOS implementation.

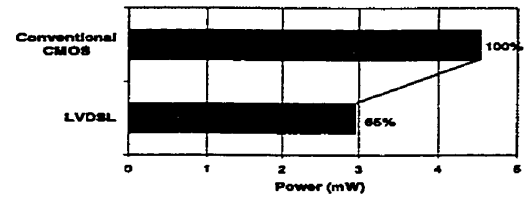
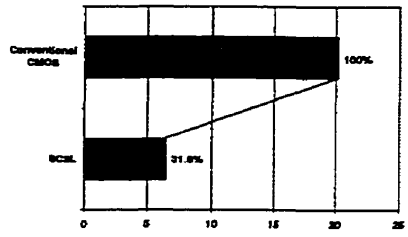


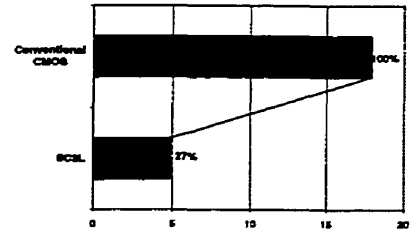
Figure 7.25. Comparison of LVDSL and conventional CMOS for ΣΔ modulator implementation

7.4.2 Fast Lock Digital PLL Implementation

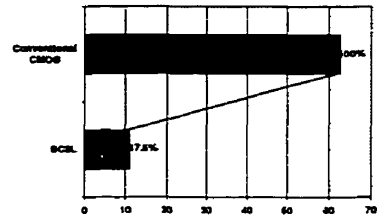
SC^2L and CPCL have been used to implement the digital logic in the fast lock digital PLL implementation. CPCL has been used in driving heavily loaded lines such as driving the output of the counter to the shift register as well as the accumulator. The adder in the accumulator was implemented as a carry ripple adder. The three implementations listed in chapter 6 (IEEE 802.11 wireless LAN implementation, DECT handset implementation, and DECT base station implementation) are all re-implemented using SC^2L. The results of these implementations are shown in Fig. 7.26. Recall that in the DECT base station implementation, eight pipelined counters were used.



(a)



(b)



(c)

Figure 7.26. SC^2L implementations of fast lock digital PLL for (a) IEEE 802.11, (b) DECT handset, and (c) DECT base station

7.6 Conclusions

It has been demonstrated that SLL has great potential for high-speed low-power applications. Two implementations of this new class of dynamic logic gates, SC^2L and

CPCL, have been shown to have complementary qualities. SC^2L can be used to implement dense datapaths with short interconnects, while CPCL consumes large area, but can drive large loads efficiently. Full-adder implementations of SC^2L and QSC^2L have been shown to exhibit an order of magnitude less power-delay product with respect to other logic families. Furthermore, asynchronous pipelining can be used to connect SC^2L gates together without introducing any special circuitry. An 8-bit CRA based on this pipelining strategy has been fabricated, tested and found to operate up to 727MHz while dissipating 2.34mW of power.

A static differential logic family, LVDSL, has also been described. LVDSL has been demonstrated to work at very low supply voltages while achieving lower power-delay products than other static logic families. A full-adder implementation of LVDSL has been shown to exhibit lower power, delay, and energy than most other logic families. A D-type flip-flop has also been demonstrated to achieve lower area than a D-type flip-flop implemented in conventional CMOS.

It has been demonstrated that when the proposed logic families are used to implement the digital portions of the PLLs proposed in chapters 5 and 6, power savings ranging from 35% to 82.5% are possible. This comes at no extra delay penalty.

Chapter 8

Summary and Conclusions

The design and analysis of agile frequency synthesizers have been detailed in this thesis. The frequency synthesizer designs have all been based on the phase-locked loop (PLL) structure. This led to two new frequency synthesizer structures. One is based on a $\Sigma\Delta$ PLL, while the other is based on digitizing portions of the PLL.

In the theoretical analysis of PLLs, spurious performance and lock time were analyzed. Closed form expressions predicting the spur location and strength have been developed using Fourier Series analysis. These expressions have been developed for both the uncompensated fractional-N PLL as well as the analog compensated fractional-N PLL. The effects of varying the compensation pulse parameters, for the case of the analog compensated fractional-N PLL, have been investigated. The closed form expressions were then validated through simulations. The lock time performance of PLLs has also been analyzed in detail. Previously, only linear models for PLLs have

been used in order to predict the locking behaviour of PLLs. In this study, a nonlinear time-domain based approach has been adopted in order to better understand the frequency locking phenomenon in PLLs. Bounds on expressions of frequency lock time have been developed. Phase locking phenomena have also been investigated. It has been determined that there seems to be an optimal trajectory in which the phase lock time of the PLL is minimized.

In the design of the first PLL, a conventional MASH ΣΔ fractional-N PLL was developed. The design was optimized for GSM and GPRS wireless communications standards. It has been demonstrated that sigma-delta modulation has succeeded in significantly reducing the spurs, but not completely eliminating them. The spur levels, however, meet the GSM and GPRS specifications. The power consumption of the ΣΔ PLL has been dominated by the divider. In future designs, lower power can be achieved by using less current in the prescaler. The effect of using a constant seed has been investigated. A constant seed did not succeed in eliminating the spurs as anticipated. To completely eliminate the spurs generated by the fractional-N divider, different architectures must be explored.

In order to improve upon the MASH ΣΔ PLL design, a wideband phase-locked loop (PLL) modulator for wireless applications is reported. This modulator is based on PLL fractional-N frequency synthesis techniques along with sigma-delta modulation to randomize fractional-N spurs. A modified sigma-delta function allows for suppression of sigma-delta noise at lower frequencies, and hence allows for wider loop bandwidth. Also, sigma-delta quantization noise is reduced by using fractional division ratios. Low-power and low-area algorithmic techniques are used in the modified sigma-delta modulator in order to make it a feasible option. The wide bandwidth of 200KHz also makes the proposed ΣΔ PLL suitable for closed loop modulation purposes.

In chapter 6, a simple yet effective method of reducing the lock time has been demonstrated. It has been shown that the lock time and frequency resolution trade-off

can be performed in the digital domain, as opposed to the analog domain. The advantage of this is that in the digital domain, the trade-off deals with the area and power penalties incurred for extra performance gains. The PLL's lock time and frequency resolution are, therefore, independent of the loop filter parameters, which enables a drastic reduction in loop filter sizes. This enables a fully integrated solution. The digital PLL architecture has been shown to be effective for low frequency resolution standards such as cordless and wireless LAN standards. For high frequency resolution standards, such as GSM, the architecture can only be used as a frequency aid circuit. The cost of fine resolution becomes excessively high in terms of area penalty as well as lock time. A simple, yet effective means of reducing the phase lock time is achieved by resetting the counters in the frequency divider at the first positive edge of the reference input after the digital PLL has acquired frequency lock to within its resolution.

Finally, power reduction on the two previously designed PLLs has been attempted by changing the underlying digital logic families used to implement the digital portions of the PLLs. A new class of logic families, called swing limited logic (SLL), has been demonstrated that it has great potential for high-speed low-power applications. Two implementations of this new class of dynamic logic gates, SC^2L and CPCL, have been shown to have complementary qualities. SC^2L can be used to implement dense datapaths with short interconnects, while CPCL consumes large area, but can drive large loads efficiently. Full-adder implementations of SC^2L has been shown to exhibit an order of magnitude less power-delay product with respect to other logic families.

A static differential logic family, LVDSL, has also been described. LVDSL has been demonstrated to work at very low supply voltages while achieving lower power-delay products than other static logic families. A full-adder implementation of LVDSL has been shown to exhibit lower power, delay, and energy than most other logic families. A D-type flip-flop has also been demonstrated to achieve lower area than a D-type flip-flop implemented in conventional CMOS.

It has been demonstrated that when the proposed logic families are used to implement the digital portions of the PLLs proposed in chapters 5 and 6, power savings ranging from 35% to 82.5% are possible. This comes as no extra delay penalty.

Appendix A

1.2 GHz ΣΔ Fractional-N Synthesizer Design Specification

1. Introduction

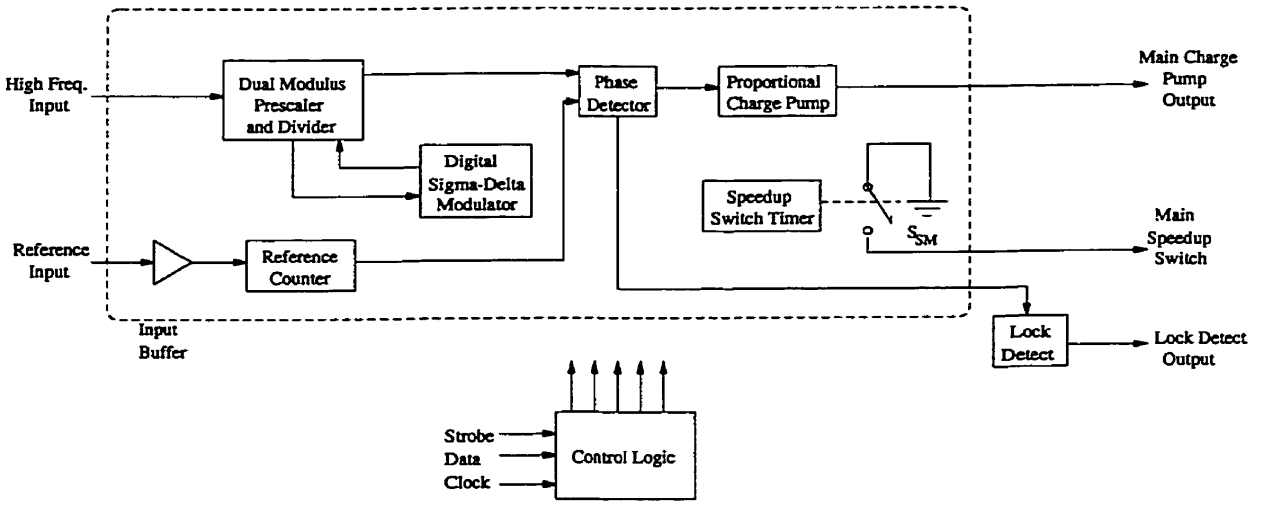
The PYRAMID Integrated Circuit is a single channel synthesizer, a main high frequency fractional channel. The main channel operates up to 1.2GHz. This part is designed primarily as a proof of concept of the sigma-delta (ΣΔ) fractional N synthesizer concept.

2. Features

1. 0.35 Micron BiCMOS

2. Fractional Main Loop to 1.2 GHz maximum frequency
3. Main Loop fractional division ratio:
4. Multifunction Lock detect output
5. Direct Access Sigma-Delta input/output
6. 3 Volt Charge Pumps
7. Ultra-Fast Lock time speedup mode for Main Loop.
8. 24 pin CSP package
9. Programmable charge pump DC current levels
10. Prescaler: 8/9/16/17
11. Sigma-Delta fractional compensation method maintains better than 30 dB of additional suppression across the entire band.

3. Block Diagram



4. Electrical Interface

4.1 Pin Assignments

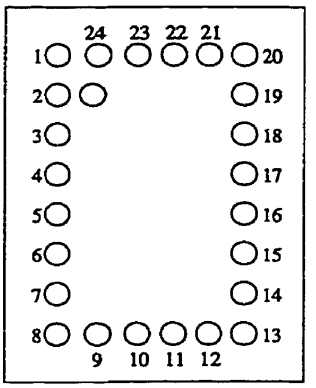


Figure A.1. 24 Pin Chip Scale Packaging (top view)

4.2 Terminal Functions

Table A.1. Chip pin list

CSP Pin#	Symbol	Function	Description
1	GCPM	--	Main Charge Pump Ground
2	PHP	O	Main Charge Pump Output
3	VCPM	I	Main Charge Pump DC Supply
4	GPSM	--	Main Prescaler Ground
5	RFMN	I	Main Channel RF input (Differential +)
6	/RFMN	I	Main Channel RF input (Differential -)
7	VPSM	I	Main Prescaler DC Supply
8	STRB	I	Serial Interface Strobe
9	DATA	I	Serial Interface Data
10	CLK	I	Serial Interface Clock
11	GDIG	--	Digital Ground
12	LDET	O	Lock Detect Output
13	VDIG	I	Digital DC Supply
14	GREF	-	Reference Buffer Ground
15	REFI	I	Reference Buffer Input

16	SD_BIT0	I	ΣΔ Bypass Input bit 0
17	SD_BIT1	I	ΣΔ Bypass Input bit 1
18	SD_OUT1	O	ΣΔ Output Bit 1
19	VCXO	I	Main Reference Buffer DC Supply
20	VDD_SD	I	ΣΔ DC Supply
21	SD_OUT2	O	ΣΔ Output Bit 2
22	SD_BIT2	I	ΣΔ Bypass Input bit 2
23	RSET	--	Charge Pump Current Setting Resistor
24	SWM	O	Speedup Switch – Main

5. Specifications

5.1 Absolute Maximum Ratings

Table A.2. Absolute Maximum Ratings

Parameter	Symbol	Value		Unit
		Min.	Max	
Supply Voltage	V_{CC}	-0.3	+6.0	V
Voltage applied to any other pin	V_{IN}	-0.3	V_{CC}+0.3	V
Maximum RF Level to RF pin			+8	dBm
Power Dissipation $T_A = 25°C$	P_D		1000	mW
Output Current (continuous)	I_{OUT}		250	mA
AC Input Voltage			V_{CC}	V_{PP}
DC Current to any I/O pin	I_{MAX}	-10	+10	mA
Storage Temperature	T_{STG}	-55	+125	°C
Lead Temperature (soldering, 10sec.)	T_L		+300	°C
Ambient Operating Temperature	T_{AMB}	-20	+75	°C

5.2 DC Electrical Characteristics

Table A.3. Recommended DC Operating Conditions

Recommended Operating Conditions						
Parameter	Symbol	Condition	Value			Unit
			Min.	Typ.	Max.	
Power Supply	V_{CC}		2.6	3.0	3.3	V
Operating Temperature	T_A		-20		+75	°C

Table A.4. DC Current Characteristics

Current Characteristics						
Parameter	Symbol	Condition	Value			Unit
			Min.	Typ.	Max.	
Regulated Voltage	V_{set}	R_{set}=20KΩ		1.0		V
External Resistor to Ground	R_{set}		16	20	24	KΩ
Supply Current [1]	I_{CC}					
Supply Current [2], Sleep mode						
Comments: 1) Average Current 2) All Digital input pins and other control signals are set to zero volts for this measurement.						

Table A.5. Reference Channel Characteristics

Reference Channel T_A=25°C (note 3.4), V_{CC}=2.85 V unless otherwise specified						
Parameter	Symbol	Condition	Values			Unit
			Min.	Typ.	Max.	
Operating Frequency	F_{REFI}	Comment 1	3.25	13	20	MHz
Input Resistance	R_{REFI}		15	20	24	KΩ
Input Capacitance	C_{REFI}				5	PF
Input Voltage	V_{REFI}	Z_{SOURCE}=1 KΩ	0.5	1.0	2.0	Vpp
Input Duty Cycle			45	50	55	%
Comments: 1) Max and min limits are for design purposes only. 13MHz is the nominal input frequency						

Table A.6. Reference Clock Buffered Output Description

Reference Clock Buffered Output T_A=25°C (note 3.4), V_{CC}=2.85 V unless otherwise specified						
Parameter	Symbol	Condition	Values			Unit
			Min.	Typ.	Max.	
Operating Frequency	F_{REFO}		3.25	13	20	MHz
Output Impedance	Z_{REFO}				300	Ω
Output Level	V_{REFO}	Comments 1,3,4,5,6	V_{REFI}		1.5	Vpp
Harmonic Suppression		Comments 1,3,6		-20		dB
Spurious Outputs		Comments 1,3,6			30	dBc
Reverse Isolation		Comments 2,3		-20		dB

Comments:

1) Z_{LOAD}=27pF||3.3K @13MHz, Z_{SOURCE}=50Ω, with 0.7 Vpp since input.
2) Z_{SOURCE}=50Ω, Z_{LOAD}=300Ω @ 13MHz
3) Measured with 1000pF DC blocking capacitors at the input and output
4) Reference Buffered output is not inverted with respect to the input.
5) Reference Buffered output does not have to be linear with respect to input.
6) A shunt tank circuit consisting of a 150 pF capacitor in parallel with a 1uH inductor is placed between the output and the load. The capacitor is of the ceramic 0402 or 0603 type and the inductor will have typical Q>35@50MHz, for example: TOKO 1008CS-102XKBC

5.3 Digital Characteristics

Table A.7. Clock, data, and strobe signal level specifications

CLK, DATA, STROBE Signal Levels T_A=25°C (note 3.4), V_{CC}=2.85 V unless otherwise specified						
Parameter	Symbol	Condition	Values			Unit
			Min.	Typ	Max.	
CLK, DATA, STROBE						
Input Logic Voltage	V_{IH}	Input Logic High	0.8×V_{CC}		V_{CC}+0.3	V
	V_{IL}	Input Logic Low	-0.3		0.2×V_{CC}	V
Input Logic Current	I_{IH}	V_{IH}	-5		+5	uA
	I_{IL}	V_{IL}	-5		+5	uA

5.4 Timing

T_A=25°C , V_{CC}=3.0 unless otherwise specified.

Table A.8. Digital Timing specification.

	Parameter (See figure 4)	**Min**	**Typ.**	**Max**	**Unit**
f_{CLOCK}	frequency, CLOCK		13	13	MHz
t_F	Fall time, CLOCK			8	ns
t_R	Rise time, CLOCK			8	ns
t_{SU_CLOCK}	Set up time, STROBE low before CLOCK↑	10			ns
	Clock Duty Cycle	30	50	70	%

t_{SU_DATA}	Setup time, DATA valid before CLOCK↑	20	ns
t_{H_DATA}	Hold time, DATA valid after CLOCK↑	20	ns
t_{W_STROBE}	Pulse width, STROBE high	20	ns
$t_{SU\_STROBE1}$	Setup time, CLOCK low before STROBE high	20	ns
$t_{SU\_STROBE2}$	Setup time, STROBE↑ before CLOCK↑	20	ns

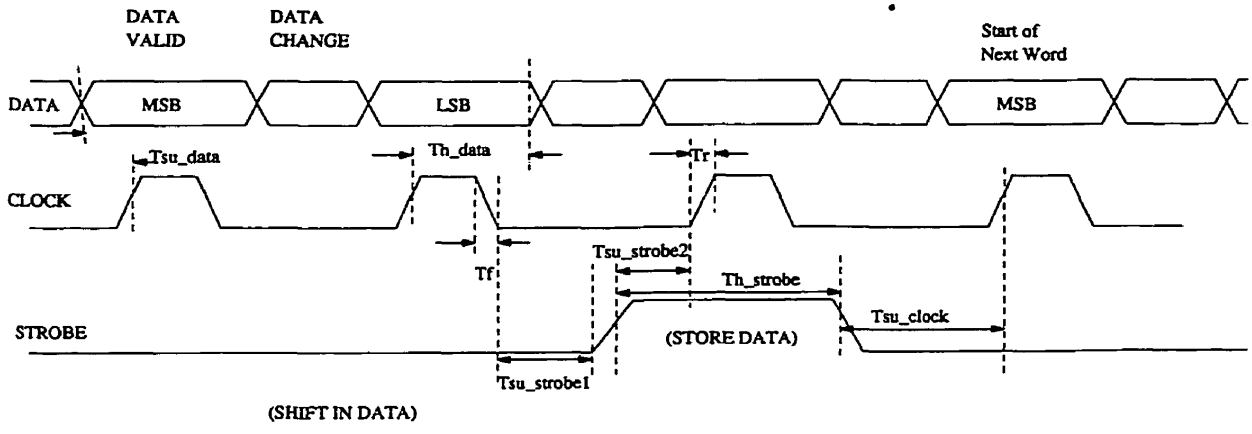


Figure 6.2. Digital Timing Specification

6.1 Serial Programming Description

The serial control hardware input consists of a standard 3-wire serial port interface bus, comprised of DATA, CLK and STROBE signals. These signals are used to input 24 bit control words to the chip. The control words contain information for programming the synthesizer, according to the protocol described in this document.

Figure A.2 describes the timing diagram for the serial input. When STROBE is low, data present on the DATA line is shifted into the integrated circuits (IC) serial register on the positive edges of the CLK signal. When STROBE goes high, data present in the serial register is latched and remains stable for any further clock cycles as long as STROBE remains high. Depending upon address bits formatted into each control word, the other data in the control word is routed into different working registers in the IC. Data is entered into the serial register in MSB first format.

At power up, the IC working Registers shall be in a default state (according to the tables described in this document) and shall not change until new control data is clocked in on the serial bus.

Data Definition:

MSB																							LSB
FIRST IN																							LAST IN
BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

A WORD – Sigma Delta Input

BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	A	FN																						
DEF	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
FN	Fractional division ratio. Default set to 2/65 fractional ratio. Prgram 23 bits from 24 bits (LSB not programmed and set to 0, A=FN/2)																							
A	A word select																							

B WORD – Main Loop Update

BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	ADDRESS				Not used				SED				NMU									CTL		SU
DEF	1	0	0	0	X	X	X	X	0	0	1	1	0	0	1	0	0	1	0	1	1	1	0	1

ADDRESS	B word select
SED	Seed of Sigma Delta. This number is decoded into a 16 bits number (2^{16}) and added to FN.
NMU	Main divider division ratio (Default=75 -> A=3, B=9 when the prescaler division ratio is 8)
CTL	Mode control for digital compensation scheme. 00=external bypass, 01=accumulator, 10=2nd order sigma-delta, 11=3rd order sigma-delta
SU	Main loop speed up mode. "0"=disable, "1"=enable

C WORD – Main Loop Setup

BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	ADDRESS				Not used						Rd		PRD		P1	Sd	Pb	CPM					Mp	PE
DEF	1	0	0	1	X	X	X	X	X	X	0	0	0	0	1	0	1	1	1	0	0	X	1	1

ADDRESS	C word select
CPM	Main Charge pump Current ratio setting. Per Table 1.
Mp	Main Prescaler Setting: 1=8/9, 0=16/17
Rd	Main Reference Divider Setting: (0,0)=div-by-1, (0,1) or (1,0)=div-by-2, (1,1)=div-by-4
P1	Main Charge Pump DC Power "0"=Power Down, "1"=Power Up
Sd	Speedup charge control "0"=software control, "1"=STRB pin control (see Comment 1)
PE	Main Prescaler DC Power "0"=Power Down, "1"=Power Up
Pb	Reference Input Buffer Amplifier Power up/down. "1"=Power up, "0"=Power down
PRD	Main PFD Reset Delay (00=6ns, 01=8ns, 10=10ns, 11=12ns)

D WORD – Main Loop Setup 2

BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	ADDRESS					te	tg	Hm	tm															
DEF	1	0	1	0	X	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0

ADDRESS	D word select
tm	Counter value for speedup mode (default=100usec -> tm=1300 for 13MHz reference)
hm	Hold main loop in speedup mode forever if hm="1", otherwise use timer ™ if hm="0"
te	Toggle Enable. "1"=Enable toggle mode, "0"=Use Main PFD lock detector LD
tg	Toggle Mode. If "te" bit is set to "Toggle mode", and the "tg" bit is set to "1" then LD pin puts out a logic high state, if "0" then LD pin puts out a logic low state. If the "te" bit is not set to "Toggle Mode," then the "tg" bit is ignored.

H WORD – Test mode word

BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	ADDRESS				Test mode																			
DEF	1	1	1	0	X	X	X	X	X	X	X	X	X	1	1	X	X	X	X	0	0	X	X	X

ADDRESS	H word select
H[4]	If H[4]=1, external seed value to sigma-delta is written through the DATA pin. If H[4]=0, external seed value from DATA pin is inhibited.
H[3]	If H[3]=1, reset the sigma-delta each time a new channel is selected. If H[3]=0, reset sigma-delta only on power up.

Table A.9. Charage pump current ratio settings

Charge Pump Current Ratio ($I_{SET}=V_{SET}/R_{SET}$) bias current for charge pumps					
BIT		I_{PHP}	BIT		I_{PHPSU}
6	5		4	3	
0	0	$16xI_{SET}$	0	0	$64xI_{SET}$
0	1	$12xI_{SET}$	0	1	$48xI_{SET}$
1	0	$8xI_{SET}$	1	0	$32xI_{SET}$
1	1	$4xI_{SET}$	1	1	$16xI_{SET}$

7. Notes

This section contains comments about any conditions, parameters or requirements that are applicable to PYRAMID and may not be described in the specifications.

7.1 General

1. **Speedup Mode Additional Requirements.** When leaving the speedup mode, it is *strongly desired* that the loop should not break the phase lock during the transition from speedup mode to normal mode. The transient response as a result of switching from speedup to normal mode can cause the speedup mode to be ineffective.
2. **Reprogramming Serial Registers.** In high reliability systems, user may want to periodically refresh the serial control registers, without changing the content of these registers. If a user continuously reprograms a control word with the same frequency/control data (i.e. there is no change in frequency), when there shall be no transient response in phase lock during the reprogramming events.

3. **Lock Time, Main Channel.** A goal of the PYRAMID is to meet 140uSec (to within 100 Hz) lock time for General Packet Radio Service (GPRS) timing requirements. This lock time should be attained using the loop filters suggested in Appendix B.
4. **RMS Phase Error** shall be calculated when necessary, using the following formula:

$$\phi_{RMS} = \frac{180}{\pi}\sqrt{2\times \int_{10}^{0.5\times BW} L(f)df} \qquad \textbf{(degrees)}$$

Where: BW= The GSM channel bandwidth (200KHz)

L(f)=The SSB phase noise (in dBc/Hz) vs. frequency offset from the carrier (f).

7.2 Operational

1. Charge pump Polarity reversal feature.

   Main Charge pump shall have only a positive VCO tuning voltage slope.

7.1 Specifications

1. **RF input sensitivity for the Main channel.** The "RF input sensitivity" line item for the Main channel assumes that the RFMN pin is AC coupled through an external 1000pF ceramic 0402 multilayer chip capacitor.
2. **Specification Guaranteed** over temperature by simulation only when this note is referenced.
3. **Fractional and Reference Spurious.** Fractional and Reference spurious are dependent upon external loop filter component characteristics, tolerances and PCB layout, which are selected by the IC user, rather than the IC designer, these specifications are guaranteed only by simulation using commerically available lumped element external component models.

8. Pyramid Test Modes

Table A.10. PYRAMID Test Modes Table

#	Test Name	Test Description	Test Vector I[9:0]
1	NORMAL MODE	Regular operating mode	000xx 01000
2	RESET MODE	Software RESET	1xxxx xxxxx
3	LD	Directs internal LD net to LD_OUT pin	0xxxx 01000
4	M_IN	Directs Main Prescalar Output to TEST PIN	000xx x0000
5	M_REF_DIV	Directs Main Ref Divided Output to the TEST PIN	0xxxx x0001
6	M_CNTB	Directs Main B Counter (6 bits) Output to TEST PIN	0xxxx x0010
8	M_CNTA	Directs Main A Counter (3 bits) Output to TEST PIN	0xxxx x0011
9	M_LD	Directs Main LD output to TEST PIN	0xxxx x0100
10	M_UP	Directs Main PFD UP output to TEST PIN	0xxxx x0101
11	M_DN	Directs Main PFD DN output to TEST PIN	0xxxx x0110
12	M_SPU_EN	Directs MAIN Speed Up enable signal to TEST PIN	0xxxx x0111
13	TEST_PIN_HIGH	Sets the TEST PIN to high to set voltage levels during DC test	0xxxx x1010
14	TEST_PIN_LOW	Sets the TEST PIN to low to set voltage levels during DC test	0xxxx x1011
15	REF_IN	Directs Ref Output to the TEST PIN	0xxxx x1001
16	MSPU_PULSE	Directs Main Loop Speedup Enable Pulse to the TEST PIN	0xxxx x11x0
17	M_INSEL	Allows override of Main prescalar and uses DATA pin signal injected into main counter	010xx xxxxx
18	REF_SEL	Allows override of Main Ref clock by DATA	011xx xxxxx

		pin signal.	
19	M_LDSEL	Allows input from the DATA pin to test Main LKDET circuit which includes the pulse filter and the delay counter to show that it's in lock.	0xxxx 11000
20	CPUMP_NORM_BOTH	Allows all pumps to be in their normal operating state and in both up and down pumping mode. This allows ability to test current matching	0xx11 11110
21	CPUMP_NORM_UP	Allows all pumps to be in their normal operating state and up pump mode.	0xx01 11110
22	CPUMP_NORM_DN	Allows all pumps to be in their normal operating state and down pump mode	0xx10 11110
23	CPUMP_NORM_Z	Allows all pumps to be in their normal operating state and high Impedance. This allows ability to test CPUMP leakage.	0xx00 11110
24	CPUMP_SPU_BOTH	Allows main pump to be in speed up operating state and in both up and down pumping mode. This allows ability to test current matching	0xx11 11111
25	CPUMP_SPU_UP	Allows main pump to be in speed up operating state and up pump mode.	0xx01 11111
26	CPUMP_SPU_DN	Allows main pump to be in speed up operating state and down pump mode	0xx10 11111
27	CPUMP_SPU_Z	Allows main pump to be in speed up operating state and high Impedance. This allows ability to test CPUMP leakage.	0xx00 11111
28	TST_SD_BIT0	Allows bit0 of sigma-delta output to be routed to the TEST PIN	0xxxx 01101
	Bits other than I[9:0] are used in the following test modes		
50	PFD_R_SEL	Allows bypassing main loop PFD reference	I<10> = 1 (enable)

		signal by bit I<10> and using DATA pin input as the reference signal for main loop PFD	
52	SD_OUT_SEL	Allows selection of either external data (through pins SD_BIT0, SD_BIT1, and SD_BIT2) or an n^{th} order sigma-delta output (where n=1,2,3)	B<2:1> = 00:bypass SD by SD_BP 01: accumulator 10: second order SD 11: third order SD
53	SD_RESET	Allows the sigma-delta modulator to be reset either whenever channel switches, or only on power-up.	H<3> = 0 (reset on power-up only)
54	SD_EXT_SEED	Allows an external dynamic seed to be fed into the sigma-delta.	H<4> = 1 (enable)

9. Test Configurations

The purpose of this appendix is to define test conditions under which certain parameters are to be measured.

Test Condition 1: Current, Spurious and Phase Noise

Setup:

VCC=3.0 volts, temperature=25C

Reference input frequency: 13MHz source per specification

Reference Buffer enabled: output load per Figure B.2

Rset=20K

Main Channel:

Enabled, Locked Normal Mode Charge Pump.

Center frequency 975.4MHz, f_{PFD}=13MHz

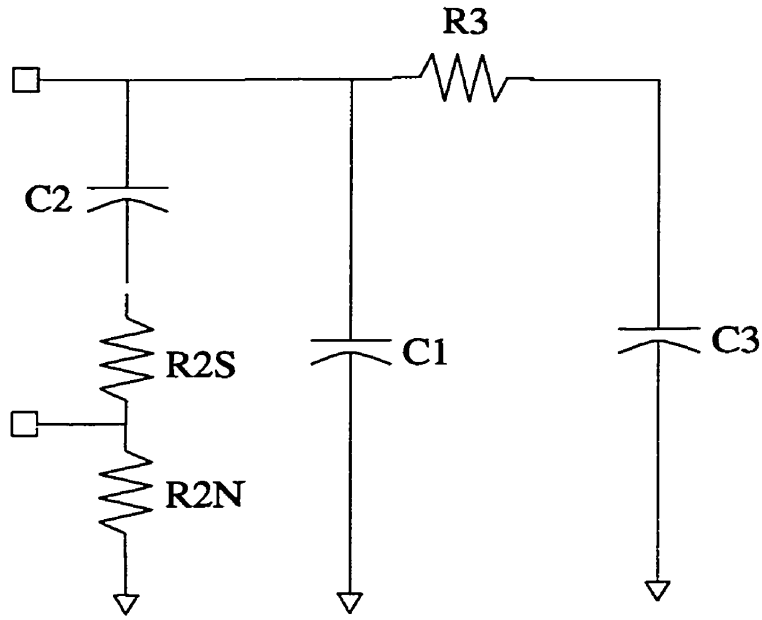
Fractional number: 2/65, Division ratio (NMU) = 75

Normal Mode Charge Pump Current Setting: 4X (0.2mA)

Loop Filter:

VCO Gain: 40MHz/V

Suggested VCO: MuRata MQE 961-554


R3
C2
R2S
C1
C3
R2N

Test Condition 2: Lock Time

Setup:

VCC=3.0 volts, temperature=25C

Reference input frequency: 13MHz source per specification

Reference Buffer enabled: output load per Figure B.2

Rset=20K

Main Channel:

Enabled, Locked Normal Mode Charge Pump.

Start Frequency: 890MHz (NMU=68, FN=30)

Stop Frequency: 915MHz (NMU=70, FN=25)

F_{PFD} = 13MHz

Fractional number (FN) = 30 to 25

Normal Mode Charge Pump Current Setting: 4X (0.2mA)

Speedup Mode Charge Pump Current Setting: 16X (0.8mA)

Speedup Mode time: 100uSec

Loop Filter/VCO: Same as Configuration 1.

Test Condition 3: Current, Spurious and Phase Noise

Setup:

VCC=3.0 volts, temperature=25C

Reference input frequency: 13MHz source per specification

Reference Buffer enabled: output load per Figure B.2

Rset=20K

Main Channel:

Enabled, Locked Normal Mode Charge Pump.

Center frequency 975.4MHz, f_{PFD}=3.25MHz

Fractional number: 8/65, Division ratio (NMU) = 300

Normal Mode Charge Pump Current Setting: 4X (0.2mA)

Loop Filter:

VCO Gain: 40MHz/V

Suggested VCO: MuRata MQE 961-554

Test Condition 4: Lock Time

Setup:

VCC=3.0 volts, temperature=25C

Reference input frequency: 13MHz source per specification

Reference Buffer enabled: output load per Figure B.2

Rset=20K

Main Channel:

Enabled, Locked Normal Mode Charge Pump.

Start Frequency: 890MHz (NMU=273, FN=55)

Stop Frequency: 915MHz (NMU=281, FN=35)

F_{PFD} = 3.25MHz

Fractional number = 0

Normal Mode Charge Pump Current Setting: 4X (0.2mA)

Speedup Mode Charge Pump Current Setting: 16X (0.8mA)

Speedup Mode time: 100uSec

Loop Filter/VCO: Same as Configuration 1.

10. Fractional Channel Number

The purpose of this appendix is to list the valid list of binary values for the fractional values used in PYRAMID.

Numerator of fraction (denominator is 65)	Binary Equivalent (24 bits)
1	000000111111000000111111
2	000001111110000001111110
3	000010111101000010111101
4	000011111100000011111100
5	000100111011000100111011
6	000101111010000101111010
7	000110111001000110111001
8	000111111000000111111000
9	001000110111001000110111
10	001001110110001001110110
11	001010110101001010110101
12	001011110100001011110100
13	001100110011001100110011
14	001101110010001101110010
15	001110110001001110110001
16	001111110000001111110000
17	010000101111010000101111
18	010001101110010001101110
19	010010101101010010101101
20	010011101100010011101100
21	010100101011010100101011
22	010101101010010101101010
23	010110101001010110101001
24	010111101000010111101000
25	011000100111011000100111
26	011001100110011001100110
27	011010100101011010100101
28	011011100100011011100100
29	011100100011011100100011
30	011101100010011101100010
31	011110100001011110100001
32	011111100000011111100000
33	100000011111100000011111
34	100001011110100001011110
35	100010011101100010011101
36	100011011100100011011100

37	100100011011100100011011
38	100101011010100101011010
39	100110011001100110011001
40	100111011000100111011000
41	101000010111101000010111
42	101001010110101001010110
43	101010010101101010010101
44	101011010100101011010100
45	101100010011101100010011
46	101101010010101101010010
47	101110010001101110010001
48	101111010000101111010000
49	110000001111110000001111
50	110001001110110001001110
51	110010001101110010001101
52	110011001100110011001100
53	110100001011110100001011
54	110101001010110101001010
55	110110001001110110001001
56	110111001000110111001000
57	111000000111111000000111
58	111001000110111001000110
59	111010000101111010000101
60	111011000100111011000100
61	111100000011111100000011
62	111101000010111101000010
63	111110000001111110000001
64	111111000000111111000000

11. Frequency Plan of Pyramid

The purpose of this appendix is to give a sample of possible operating frequencies for PYRAMID assuming a 13MHz input reference frequency.

Prescaler Division Ratio	**Reference Division Ratio**	**F_{MIN}**	**F_{MAX}**
8/9	1	728MHz	>1.2GHz
	2	364MHz	>1.2GHz
	4	182MHz	>1.2GHz
16/17	1	Cannot be used (out of range)	
	2	Cannot be used (out of range)	
	4	780MHz	>1.2GHz

12. Operating Modes of the Digital ΣΔ Modulator

Digital Sigma-Delta Modulation techniques can be applied to fractional-N PLLs in order to digitally compensate for fractional spurs. The advantage of this technique over analog compensation is that it avoids the limitation of reduction of fractional spurs due to device mismatches. Indeed, digital sigma-delta modulation, when applied to fractional-N PLLs, has been shown to be able to achieve lower spurious levels for the same PLL loop parameters.

The Digital Sigma-Delta Modulator (DSDM) built for the PYRAMID project was constructed in such a way as to be as flexible as possible. It was pipelined to achieve a maximum operating frequency of 65MHz. This high speed can alternatively be traded-off for lower power consumption by reducing the power supply of the DSDM. Two control bits (which can be programmed through the serial interface) can choose between a first, second, or third order DSDM, according to Table E.1 listed below. A first order DSDM is simply an accumulator with the carry out signal added to the division ratio (NMU). This is the same as a conventional uncompensated fractional-N PLL. Second order sigma-delta achieves spur randomization, at the expense of an increase of phase noise by 20dB/decade (at the VCO output). Third order sigma-delta achieves even more randomization of fractional-N spurs, at the expense of higher increase of phase noise, a 40dB/decade increase of phase noise (at the VCO output). Note that an external bypass option is inserted, where the entire sigma-delta modulator is bypassed, and instead, an external 3-bit value is fed into the adder preceding the digital divider.

Table A.11 Operating modes of the Digital Sigma-Delta Modulator (DSDM)

Control Bits	Operation
00	External Bypass
01	First-order Sigma-Delta
10	Second-order Sigma-Delta
11	Third-order Sigma-Delta

The output of second and third order sigma-delta modulators differ from the first order sigma-delta in that their outputs may be positively or negatively valued. The maximum output range of each modulator order is listed in Table E.2. In order to efficiently code the output of each modulator order, the binary codes listed in Table E.3 were used. Note that if an external bypass value is to be used instead of the sigma-delta modulator output, it still must adhere to the codes listed in Table E.3.

Table A.12. Output Range of the DSDM

Sigma-Delta Modulator Order	Output Range	
	Minimum	Maximum
First Order	0	1
Second Order	-1	2
Third Order	-4	3

Table A.13. Binary Coding Scheme used in DSDM

Value	Binary Code
4	100
3	011
2	010
1	001
0	000
-1	111
-2	110
-3	101

Two modes exist for resetting the sigma-delta. In the first mode, the sigma-delta is reset only on power-on. In the second mode, the sigma-delta is reset whenever there is a new channel selection. The impact of either option on the output spectrum of PYRAMID still remains to be studied when the device is fabricated.

List of Contributions

1. Articles published/accepted/submitted to refereed journals

[1] A. M. Fahim and M. I. Elmasry, "A Fast Lock Digital Phase-Locked Loop Architecture for Wireless Applications," submitted to *IEEE Transactions on Circuits and Systems II*, Aug. 2000.

[2] A. M. Fahim and M. I. Elmasry, "A Wideband Sigma-Delta Phase-Locked Loop Modulator for Wireless Applications," submitted to *IEEE Transactions on Circuits and Systems II*, Aug. 2000.

[3] A. M. Fahim and M. I. Elmasry, "Low-Power High-Performance Dynamic Differential Logic Families," submitted to *IEEE Journal of Solid-State Circuits*.

[4] A. Bellaouar, M. S. Obrecht, A. M. Fahim, and M. I. Elmasry, "Low-Power Direct Digital Synthesis for Wireless Communications," *IEEE Journal of Solid-State Circuits*, pp. 385-390, March 2000.

2. Articles published/accepted/submitted to refereed conferences

[5] A. M. Fahim and M. I. Elmasry, "A Low-Voltage Energy Scalable Static Differential Logic (ES^2DL) Family," *European Solid-State Circuits Conference*, pp. 114-117, September 21-23, 1999.

[6] A. M. Fahim and M. I. Elmasry, "SC^2L; A Low-Power High-Performance Dynamic Differential Logic Family," *International Symposium on Low Power Electronics and Design*, pp. 88-90, August 16-17, 1999.

[7] A. Bellaouar, M. S. Obrecht, A. M. Fahim, and M. I. Elmasry, "A Low-Power Direct Digital Frequency Synthesizer for Wireless Applications," *IEEE Custom Integrated Circuits Conference*, pp. 593-597, May 15-17, 1999.

[8] A. M. Fahim and M. I. Elmasry, "A Low-Voltage Differential Static Logic (LVDSL) Family," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 230-233, May 30 – June 2, 1999.

[9] A. M. Fahim and M. I. Elmasry, "Low-Power Frequency Synthesizer Design Methodology for Wireless Applications," *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 115-119, 1998 (printed in 1999).

3. Non-refereed contributions

[10] A. M. Fahim and M. I. Elmasry, "SC^2L; A Low-Power High-Performance Dynamic Differential Logic Family," *Micronet Annual Workshop 2000*, pp. 88-89, April 26-27, 2000.

[11] A. M. Fahim and M. I. Elmasry, "A Low-Power CMOS Frequency Synthesizer Design Methodology for Wireless Systems," *Micronet Annual Workshop 1998*, pp. 45-46, April 5-7, 1998.

[12] A. M. Fahim and M. I. Elmasry, "A Low-Power PLL-based Frequency Synthesizer for Wireless Applications," University of Waterloo, VLSI Report #UW/VLSI 96-01, April 1996.

[13] A. M. Fahim and M. I. Elmasry, "A Low-Power PLL-based Frequency Synthesizer for Wireless Applications," University of Waterloo, VLSI Report #UW/VLSI 96-01, April 1996.

Bibliography

[1] R. Best, *Phase-locked Loops: Theory, Design, and Applications*, New York: McGraw-Hill, Inc., 1993.

[2] U. Rohde, *Digital PLL Frequency Synthesizers: Theory and Design*, Englewood Cliffs, N.J.: Prentice Hall, 1983.

[3] D. Wolaver, *Phase-locked Loop Circuit Design*, Englewood Cliffs, N.J.: Prentice-Hall, 1991.

[4] Analog Devices. *ADF4026/ADF4207/ADF4208: Dual PLL Frequency Synthesizer*, 1999, Datasheet.

[5] Philsar, *PS-XX00 Series: Fractional-N Synthesizers*, Philsar, 1999, Product Datasheet.

[6] A. Kajiwara and M. Nakagawa, "A New PLL Frequency Synthesizer with High Speed Switching," *IEEE Trans. On Vehicular Technology*, vol. 41, no. 4, pp. 407-413, Nov. 1992.

[7] T. Aytur and J. Khoury, "Advantages of Dual-Loop Frequency Synthesizers for GSM Applications," *IEEE Int'l Symposium on Circuits and Systems*, pp. 17-20, 1997.

[8] C. Yuen, C. Wong, and K. Tsang, "1uS Fast Acquisition 2.4GHz Low Voltage Frequency Synthesizer for FH-CDMA Mobile Communications," *Asia Pacific Microwave Conference, paper# 3P01-1, 1997.*

[9] B. Zhang and P. Allen, "Feed-Forward Compensated High Switching Speed Digital Phase-Locked Loop Frequency Synthesizer," *ISCAS*, vol. 4, pp. 371-374, 1999.

[10] R. Dulk, "Digital Fast Acquisition Method for Phase-Lock Loops," *Electronics Letters*, vol. 24, no. 17, pp. 1079-1080, 18th August 1988.

[11] H. Sato, K. Kato, and T. Sase, "A Fast Pull-In PLL IC Using Two-Mode Pull-In Technique," *Electronics and Communications in Japan*, Part 2, vol. 75, no. 3, pp. 41-50, 1992.

[12] J. Lee and B. Kim, "A 250MHz Low Jitter Adaptive Bandwidth PLL," *ISSCC,* pp. 346-347,477, 1999.

[13] H. Shirahama, K. Taniguchi, and K. Nakashi, "A New Very Fast Pull-in PLL System with Anti-Pseudo-Lock Function," *Symposium on VLSI Circuits*, pp. 75-76, 1993.

[14] A. Djemouai, M. Sawan, and M. Slamani, "A 200MHz Frequency-Locked Loop Based on New Frequency-to-Voltage Converters Approach," *ISCAS*, vol. 2, pp. 89-92, 1999.

[15] B. Goldberg, *Digital Techniques in Frequency Synthesis*, New York, NY: McGraw-Hill, 1996.

[16] .*Matlab 5: Owner's Manual*, Mathworks, Natick, Massachusetts, 1999.

[17] M. Manassewitsch, *Frequency Synthesizers Theory and Design*, New York, NY: John Wiley & Sons, 1980.

[18] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, "Quantization and dither: a theoretical survey," *J. Audio Eng. Soc.*, vol. 40, no. 5, pp. 355-375, May 1992.

[19] A. Hill and J. Surber, "The PLL dead zone and how to avoid it," *RF Design*, pp. 131-134, March 1992.

[20] W. Lindsey and C. Chie, "A Survey of Digital Phase-Locked Loops," *Proceedings of the IEEE*, vol. 69, April 1981.

[21] Motorola Inc., *The Technique of Direct Programming by Using a Two-Modulus Prescaler*, 1981, Application Note AN827.

[22] P. Larsson, "High-Speed Architecture for a Programmable Frequency Divider and a Dual-Modulus Prescaler," *IEEE J. of Solid-State Circuits*, vol. 31, no. 5, pp. 744-748, May 1996.

[23] W. Rhee, "Design of high-performance CMOS charge-pumps in phase-locked loops," *IEEE Int'l Smposium on Circuits and Systems*, vol. II, pp. 545-458, 1999.

[24] S. Norsworthy, R. Schreier, G. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE Press: New York, 1997.

[25] T. Riley, M. Copeland, and T. Kwasniewski, "Delta-sigma modulation in fractional-N frequency synthesis," *IEEE J. Solid-State Circuits*, vol. 28, no. 5, pp. 553-559, May 1993.

[26] L. Williams, III and B. Wooley, "Third-order cascaded sigma-delta modulators," *IEEE Transactions on Circuits and Systems*, vol. CAS-38, pp. 489-498, May 1991.

[27] R. Schneiderman, *Future Talk: The Changing Wireless Game*, IEEE Press: New York, 1997.

[28] ISO/IEC JTC1/SC29 WG11/602, Recommendation H.262, 1993.

[29] J. Zhang et. al, "MPEG-2 Video Serivces for Wireless ATM Networks," *IEEE J. on Selected Areas in Communications*, vol. 15, no. 1, pp. 119-128, January 1997.

[30] T. Ojanpera and R. Prasad, *Wideband CMDA for Third Generation Mobile Communications*, Artech House: Boston, 1998.

[31] K. Park, *Personal and Wireless Communications: Digital Technology and Standards*, Kluwer Academic Publishers: Boston, 1996.

[32] M. Catedra and J. Perez-Arriaga, *Cell Planning for Wireless Communications*, Artech House: Boston, 1999.

[33] C. Wuensch, "Fractional-N Synthesizers meet HSCSD and GPRS demands," *Microwaves & RF*, pp. 105-121, Aug. 1999.

[34] J. Crawford, *Frequency Synthesizer Design Handbook*, Artech House: Norwood, MA, 1994.

[35] M. Curtin and P. O'Brien, "Phase-Locked Loops for High-Frequency Receivers and Transmitters – Part 2", *Analog Dialogue 33-5*, pp. 13-22, 1999.

[36] W. Chen and J. Wu, "A 2V 1.6 GHz BJT Phase-Locked Loop," *IEEE Custom Integrated Circuits Conference*, pp. 563-566, 1998.

[37] F. Gardner, "Charge-pump phase-lock loops," *IEEE Transactions on Communications*, vol. COM-28, pp. 1849-1858, Nov. 1980.

[38] B. Miller and R. Conley, "A Multiple Modulator Fractional Divider," *IEEE Trans. Instrum. Meas.*, vol. 40, pp. 578-583, June 1991.

[39] M. Perrott, T. Tewksbury, and C. Sodini, "A 27-mW CMOS fractional-N synthesizer using digital compensation for 2.5-Mb/s GFSK modulation," *IEEE J. of Solid-State Circuits*, vol. 32, no. 12, pp. 2048-2060, Dec. 1997.

[40] W. Rhee, A. Ali, and B. Song, "A 1.1GHz CMOS Fractional-N Frequency Synthesizer with a 3b 3rd-order $\Sigma\Delta$ Modulator," *Int'l Solid-State Circuits Conference*, pp. 198-199, 2000.

[41] J. Cranickx and M. Steyaert, "A 1.75-GHz/3-V dual modulus divide-by-128/129 prescaler in 0.7μm CMOS," *IEEE J. of Solid-State Circuits*, vol. 31, pp. 890-897, July 1996.

[42] I. Koren, *Computer Arithmetic Algorithms*, Prentice Hall: Englewood Cliffs: New Jersey, 1993.

[43] K. Hwang, *Computer Arithmetic: Principles, Architecture, and Design*, John Wiley & Sons, New York, 1979.

[44] T. Akiyama, H. Suzuki, S. Sato, and K. Nonaka, "A 1.8GHz Low power and Low voltage silicon dual modulus prescaler," *IEEE Symposium on VLSI Circuits*, pp. 95-96, 1987.

[45] M. Mizuno, H. Suzuki, M. Ogawa, K. Sato, and H. Ichikawa, "A 3mW 1.0GHz Silicon ECL Dual-Modulus Prescaler IC," *Int'l Solid-State Circuits Conference*, pp. 90-91, 1992.

[46] T. Seneff, L. McKay, K. Sakamoto and N. Tracht, "A Sub-1mA 1.5-GHz Silicon Bipolar Dual Modulus Prescaler," *IEEE J. of Solid-State Circuits*, vol. 29, no. 10, pp. 1206-1211, Oct. 1994.

[47] M. Perrot, "Techniques for High Data Rate Modulation and Low Power Operation of Fractional-N Frequency Synthesizers," PhD Dissertation, Massachusetts Institute of Technology, MA, 1997.

[48] D. Thomas and P. Moorby, *The Verilog Hardware Description Language.* Kluwer Academic Publishers: Boston, 1998.

[49] D. FitzPatrick and I. Miller, *Analog Behavioral Modeling with the Verilog-A Language*, Kluwer Academic Publishers: Boston, 1998.

[50] Cadence On-line Tutorial, *SpectreSVerilog Reference Manual*, 1999.

[51] R. Gu, K. Sharaf, and M. I. Elmasry, *High-Performance Digital VLSI Circuit Design*, Kluwer Academic Publishers: Boston, 1996.

[52] B. Kong, J. Choi, S. Lee, and K. Lee, "Charge Recycling Differential Logic (CRDL) for Low Power Application," *J. of Solid-State Circuits*, vol. 31, no. 9, pp. 1267-1276, September 1996.

[53] B. G. Streetman, *Solid State Electronic Devices*, Prentice-Hall: Englewood Cliffs, NJ, 1995.

[54] P. Strivastava, A. Pua, and L. Welch, "Issues in the Design of Domino Logic Circuits," *IEEE Great Lakes Symposium*, pp. 108-112, 1998.

[55] A. Fahim and M. Elmasry, "SC^2L; A Low-Power High-Performance Dynamic Differential Logic Family," *IEEE Int'l Symposium on Low Power Electronics Design*, pp. 88-90, 1999.

[56] P. Ng, P. Balsara, and D. Steiss, "Performance of CMOS Differential Circuits", *J. of Solid-State Circuits*, vol. 31, no. 6, pp. 841-846, June 1996.

[57] K. Hwang, *Computer Arithmetic: Principles, Architecture, and Design*, John Wiley and Sons, 1979.

[58] A. D. Booth, "A signed binary multiplication technique," Quart. J. Mech. Applied Math, no. 4, part 2, 1951.

[59] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Reading, Massachusetts: Addison-Wesley Publishing Co., 1993.

[60] M. Suzuki, et. al., "A 1.5-ns 32-b CMOS ALU in Double Pass-Transistor Logic," *IEEE J. of Solid-State Circuits*, vol. SC-28, no. 11, pp. 1145-1151, November 1993.

[61] A. M. Fahim and M. I. Elmasry, "A Fast Lock Digital Phase-Locked Loop Architecture for Wireless Applications," submitted to *IEEE Transactions on Circuits and Systems II*, Aug. 2000.

[62] A. M. Fahim and M. I. Elmasry, "A Wideband Sigma-Delta Phase-Locked Loop Modulator for Wireless Applications," submitted to *IEEE Transactions on Circuits and Systems II*, Aug. 2000.

[63] A. M. Fahim and M. I. Elmasry, "Low-Power High-Performance Dynamic Differential Logic Families," submitted to *IEEE Transactions on VLSI, Low-Power Electronics Special Issue*, December 2000.

[64] A. M. Fahim and M. I. Elmasry, "A Low-Voltage Energy Scalable Static Differential Logic (ES^2DL) Family," *European Solid-State Circuits Conference*, pp. 114-117, September 21-23, 1999.

[65] A. M. Fahim and M. I. Elmasry, "A Low-Voltage Differential Static Logic (LVDSL) Family," *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 230-233, May 30 – June 2, 1999.

[66] A. M. Fahim and M. I. Elmasry, "Low-Power Frequency Synthesizer Design Methodology for Wireless Applications," *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 115-119, 1998 (printed in 1999).

[67] A. M. Fahim and M. I. Elmasry, "SC^2L; A Low-Power High-Performance Dynamic Differential Logic Family," *Micronet Annual Workshop 2000*, pp. 88-89, April 26-27, 2000.

[68] A. M. Fahim and M. I. Elmasry, "A Low-Power CMOS Frequency Synthesizer Design Methodology for Wireless Systems," *Micronet Annual Workshop 1998*, pp. 45-46, April 5-7, 1998.

[69] A. M. Fahim and M. I. Elmasry, "A Low-Power PLL-based Frequency Synthesizer for Wireless Applications," University of Waterloo, VLSI Report #UW/VLSI 96-01, April 1996.

[70] H. Hsu, *Signals and Systems*, McGraw Hill: New York, 1995.

[71] L. Pfennings, et. al, "Differential Split-Level CMOS Logic for Subnanosecond Speeds," *J. of Solid-State Circuits*, vol. Sc-20, no. 5, pp. 1050-1055, Oct. 1985.

[72] N. Filiol, T. Riley, C. Plett, and M. Copeland, "An agile ISM band frequency synthesizer with built-in GMSK data modulation," *IEEE J. of Solid-State Circuits*, vol. 28, pp. 553-559, May 1993.

[73] T. Weigandt, *Low-Phase-Noise, Low-Timing-Jitter Design Techniques for Delay Cell Based VCOs and Frequency Synthesizers*, Ph.D. Dissertation, University of California at Berkeley, 1998.

[74] T. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press: Cambridge, UK, 1998.