

Copyright © 1990, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**HIGH SPEED CLOCK RECOVERY IN VLSI
USING HYBRID ANALOG/DIGITAL TECHNIQUES**

by

Beomsup Kim

Memorandum No. UCB/ERL M90/50

6 June 1990

**HIGH SPEED CLOCK RECOVERY IN VLSI
USING HYBRID ANALOG/DIGITAL TECHNIQUES**

by

Beomsup Kim

Memorandum No. UCB/ERL M90/50

6 June 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

High Speed Clock Recovery in VLSI Using Hybrid Analog/Digital Techniques

Ph.D.

Beomsup Kim

Department of EECS

Chairman of Committee

Abstract

The increasing popularity of high speed data transmission systems such as local area networks (e.g. Ethernet, Token Ring etc.), disk drive systems, and optical communication systems (e.g. FDDI etc.) along with constant evolution of better IC technologies puts several stringent requirements on the clock recovery circuits. Current implementations for these applications usually make use of an analog PLL implemented in bipolar technologies, often using an emitter coupled multivibrator. These techniques are limited in their ability to implement more sophisticated algorithms at high speed, and are not well-suited to implementation in CMOS technology needed to achieve higher level of integration and lower power dissipation.

The purpose of this research project is to explore an alternative architecture which overcomes the limitations of the previous approaches and to investigate the circuit design issues associated with the new architecture. Based on this research, the *hybrid analog/digital clock recovery technique* which possesses the merits of analog PLL's and digital PLL's is of potential interest in high speed data transmission and shows several advantages such as fast acquisition, low jitter, low error rate, and capability of implementation of sophisticated algorithms.

A prototype chip was fabricated to evaluate the performance of the hybrid analog/digital clock recovery circuit in 2 μ m CMOS process. This chip operates at a maximum data rate of 33 MHz from a single 5-V power supply and achieves very fast phase acquisition, a decode window of 94% of full window width, effective sampling jitter of 100ps RMS, and an effective input sampling rate of 1 GHz. The ring oscillator in the analog PLL shows a 62 ppm/deg C temperature coefficient (TC) and 4.5% supply sensitivity of free-running frequency. The total power dissipation is about 600 mW and the active area is 30,000 mil² (20 mm² in a 2 μ m single-poly double-metal n-well CMOS process).

ACKNOWLEDGEMENT

I would like to express my deepest appreciations to the thoughtful guidance and the continuous support that I have been privileged to receive from my advisor, Professor Paul R. Gray. His advice is always valuable. I also would like to thank Professor Rober R. Meyer for his help and suggestions on this thesis.

The exchange of ideas with fellow students at UC Berkeley is particularly fruitful. Among them, Yui-Min Lin, Cormac Conroy, Gani Jusuf, Greg Uehara, David Helman, Steve Lewis, Sehat Sutarja come into mind. David Helman designed and layed out 1 μ m version of the chip and Ken Lutz provided me his assistance in the lab set up and PC board design.

Special thanks goes to Cormac Conroy for his continuous proofreading this thesis and many valuable suggestions. I would also like to thank Yiu-Min Lin for sharing many fruitful discussion with me. To all of the IC group students I have worked with, I appreciate all the help.

Finally, I dedicate this thesis to my dear wife Seung Hee Choi. Her love, endurance and sacrifice she has been filling me with every single day in the monotonicity of the Albany village life enable me to finish my work. Also, I thank my and my wife's family for their encouragement and support.

This research is sponsored by the National Science Foundation under grant MIP-8801013, California MICRO program, Level One Communication, Xerox Corporation, and Texas Instruments. Prototype fabrication was done by MOSIS.

Table of Contents

Chapter 1 - Introduction	1
1.1 Background and Motivation	1
1.2 Thesis Organization	2
Chapter 2 - Clock Recovery in Digital Communication Systems	3
2.1 Introduction	3
2.2 Functional Objectives of Clock Recovery Systems	3
2.3 Clock Recovery System Performance Parameters	5
2.3.1 Input Characteristics	5
2.3.1.1 Line Coding	5
2.3.1.2 Intersymbol Interference	7
2.3.1.3 Input Phase Jitter	8
2.3.2 System Performance Parameters	10
2.3.2.1 Clock Phase Jitter	10
2.3.2.2 Acquisition Speed	11
2.3.2.3 Static Phase Offset and Stability	12
Chapter 3 - Clock Recovery Architecture Review and Comparison	14
3.1 Introduction	14
3.2 Open Loop Clock Recovery Techniques	15
3.2.1 Spectral Line Methods	15
3.3 Closed Loop Clock Recovery Techniques	17
3.3.1 Continuous-Time Clock Recovery Methods	18
3.3.1.1 Early-Late Methods	18
3.3.1.2 Transition-Detection Based Methods	21
3.3.2 Discrete-Time Clock Recovery Methods	24
3.3.2.1 MMSE Clock Recovery Methods	25
3.3.2.2 Baud-Rate Clock Recovery Methods	26
3.4 Comparison of Clock Recovery Techniques	27
Chapter 4 - Hybrid Analog/Digital PLL	29
4.1 Overview	29
4.2 Architecture	32
4.2.1 Analog Phase-Locked Loop	34
4.2.2 Parallel Phase Sampler	39

4.2.3	Transition Detector and Digital Phase-Locked Loop	40
4.2.4	Characteristics of Hybrid Analog/Digital Clock Recovery	46
4.3	Circuit Design Issues	48
4.3.1	Differential Delay Cell	49
4.3.2	Replica Biasing Circuit	51
4.3.3	Harmonic Oscillation	53
4.3.4	Dual-Latch Data Path	55
4.3.5	Pipelining in Digital Phase-Locked Loop	56
4.3.6	Phase Quantization Effects	60
4.4	Design of Hybrid Clock Recovery Circuit	61
4.4.1	Summary of Circuit Design Issues	61
4.4.2	Possible Extensions for Increased Performance	62
4.4.3	Projected Performance in Scaled Technologies	63
Chapter 5	Jitter Analysis	66
5.1	Introduction	66
5.2	Jitter in Ring Oscillator	67
5.2.1	Fast-Slewing Saturated Ring Oscillator	69
5.2.2	Slow-Slewing Saturated Ring Oscillator	74
5.2.3	Non-Saturated Ring Oscillator	79
5.2.4	Comparison of Ring Oscillator Jitter Performances	81
5.3	PLL Phase Noise Accumulation	82
5.4	DLL Phase Jitter	86
5.5	Example: PLL Noise Analysis with Slow-Slewing Saturated Ring Oscillator	89
5.6	PLL Phase Jitter System Simulation	96
Chapter 6	Optimal K Sequence for the DPLL	100
6.1	Introduction	100
6.2	Problem Definition	100
6.3	Optimization Based on Minimum Mean Square Criterion	103
6.4	Optimization Based on Minimum Mean Sum of Squares Criterion	105
Chapter 7	Experimental Results	111
7.1	Introduction	111
7.2	Analog PLL and Parallel Phase Sampler	111
7.3	Hybrid Analog/Digital Clock Recovery	115
Chapter 8	Conclusion	120

8.1 Summary of Research Results	120
References	121

Chapter 1

Introduction

1.1 Background and Motivation

In modern data transmission systems such as local area networks, disk drive systems, telecommunication networks, and optical communication systems, information is transmitted or received in the form of either baseband or passband signals containing sequences of digital data symbols. In these applications, usually only data signals are transmitted by the transmitter and separate clock signals used to synchronize the data are not transmitted to save the expense of interconnection and hardware. Therefore, the receiver should incorporate a circuit block, called the *clock recovery circuit*, to recover the clock information from the received data, and synchronize the data with the recovered clock.

Since the demands of the data transmission system grow fast, many alternatives are studied to achieve high performance clock recovery circuits for several different applications ranging from relatively slow digital subscriber loops to high speed optical data communications. One of the most challenging areas for the clock recovery circuits is high speed data transmission applications such as the disk drive read channel, the local area networks (Ethernet, Token Ring etc.), and optical communications (FDDI etc.).

In these applications, clock recovery from received data must be performed with stringent requirements on small static phase offset, low sensitivity of decode error to phase jitter, and various programming capabilities. Currently most high speed clock recovery circuits make use of an analog PLL implemented usually in bipolar technology, using either an emitter coupled multivibrator or a starved ring oscillator

VCO,[H.LEE] and utilize techniques such as zero-phase start and PLL time constant gear-shifting to achieve fast acquisition at the start of preamble. But these techniques are limited in their ability to implement more sophisticated algorithms at high speed, and also are not well-suited to the CMOS implementation needed to achieve higher levels of integration and lower power dissipation in high speed data transmission systems.

This dissertation describes a new architecture based on an analog PLL and a digital PLL, namely the *hybrid analog/digital clock recovery circuit*, which was developed to allow the use of CMOS technology to achieve high speed data recovery, fast phase acquisition and large input jitter tolerance and also allow implementation of sophisticated algorithms at high speed such as digital zero-phase start, digital gear-shifting, programmable window width/offset, and programmable pulse pair compensation. The large phase jumps normally associated with digital PLL's are avoided.

1.2 Thesis Organization

Chapter 2 introduces the functional objectives of clock recovery systems and their performance issues. In Chapter 3, several conventional clock recovery methods are reviewed and compared. Chapter 4 describes the overall architecture of the hybrid analog/digital clock recovery system, investigates the circuit design issues and addresses their solutions. One of key performance issues, PLL jitter, is studied in depth in chapter 5. Chapter 6 derives the optimal solution of the digital PLL control parameter sequences based on a minimum mean square error criterion. In chapter 7, experimental results from the 2 μ m CMOS prototype hybrid analog/digital clock recovery circuit are presented. Finally, chapter 8 gives a summary of the research results and conclusions.

Chapter 2

Clock Recovery in Digital Communication Systems

2.1 Introduction

In this chapter, firstly, the objectives of clock recovery circuits are introduced. Next, the problems and the performance issues related to clock recovery circuits are reviewed. This chapter forms the basis for the following chapters.

2.2 Functional Objectives of Clock Recovery Systems

In most data transmission applications in local area networks, disk drive systems, telecommunications and optical communications, information is transmitted or received in the form of either modulated or unmodulated sequences of digital data bits. In these applications, usually only data signals are transmitted by the transmitter and separate clock signals are not transmitted in order to save the expense of interconnect and system hardware. Therefore, the receiver should incorporate some circuits to recover the clock signals from the received data signals, and synchronize the data with the recovered clock. These circuits are usually called *clock recovery circuits* or *timing recovery circuits*.

Figure 2.1 shows the clock recovery function. Here the clock recovery circuit takes the data bit stream and generates clock pulses with rising edges in the center of each received data bit. The data decision circuit synchronizes the data bit stream using the rising edge of the recovered clock pulses. Since initially, the receiver does not know the phase and the frequency of the incoming data pulses, the clock recovery circuit needs a time period, called the *preamble period*, to acquire the phase and frequency. Also, in steady state, the clock recovery circuit must maintain the frequency

and phase locking in the presence of input jitter and missing pulses (data "0"'s). In the following sections, these issues are explained in detail.

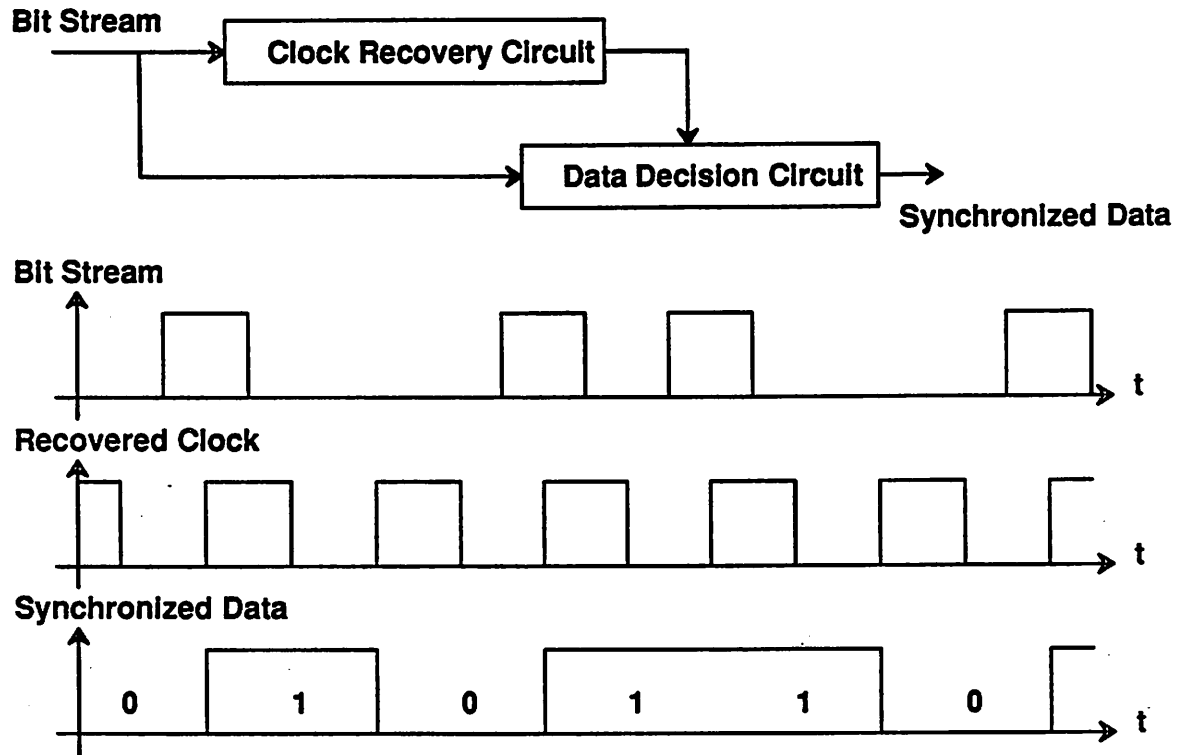


Figure 2.1 Function of Clock Recovery Circuits

Figure 2.2 shows the block diagram of typical data transmission receiver with a clock recovery circuit. Here, the preamp circuit detects the low level electric signals from the communication channel (e.g disk drive heads, coaxial cables, and twisted pair wires), and amplifies them. The following AGC circuit amplifies the outputs of the preamp circuit to a proper voltage level. The equalization filter reshapes the signal degraded by the communication channel to the original pulse shape. The pulses coming from the equalization filter are used to derive clock information and synchronize the data with the recovered clock signals generated from the clock recovery circuit. Currently, most high speed clock recovery circuits make use of an analog PLL

implemented in bipolar technology. But to achieve higher level of integration and lower power dissipation for future data transmission receiver applications, the use of CMOS technology is preferred.

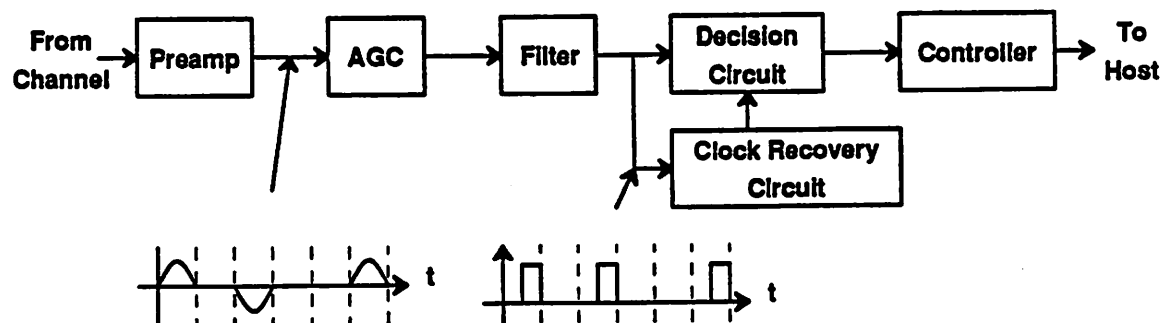


Figure 2.2 Typical Data Transmission System Receiver

2.3 Clock Recovery System Performance Parameters

In this section, some of the important performance parameters of clock recovery circuits are presented. The subject is divided into two parts; (i) problems associated with the input signal source, (ii) system level issues.

2.3.1 Input Characteristics

In this section, various line coding methods for the input signal are described, and the intersymbol interference and the phase jitter associated with the input signal are discussed.

2.3.1.1 Line Coding

Line coding techniques have been extensively studied since the emergence of local area networks and the ongoing evolution of the telecommunication network to digital service. Line coding is usually used in broadband communication systems such

as baseband local area networks (Ethernet, token ring, optical networks and so forth), disk drive read and write channel, and short loop digital PBX connections for terminals, hosts, and digital phones. In such systems, the transmission media cause only moderate distortion of the pulse because the bandwidth of the channel is wider than the spectrum of the signals. The important requirements of the line coding are spectral shaping, reduction of dc wander, bandwidth efficiency, and signal synchronization capabilities.

The easiest way to shape the spectrum of the transmitted signal in a binary signaling scheme is a suitable choice of the signal waveforms. Figure 2.3 shows the signal waveforms for *nonreturn-to-zero* (NRZ) code, *return-to-zero* (RZ) code, *Manchester* (or *biphase*) code, and *Miller* (or *delay modulation*) code scheme.

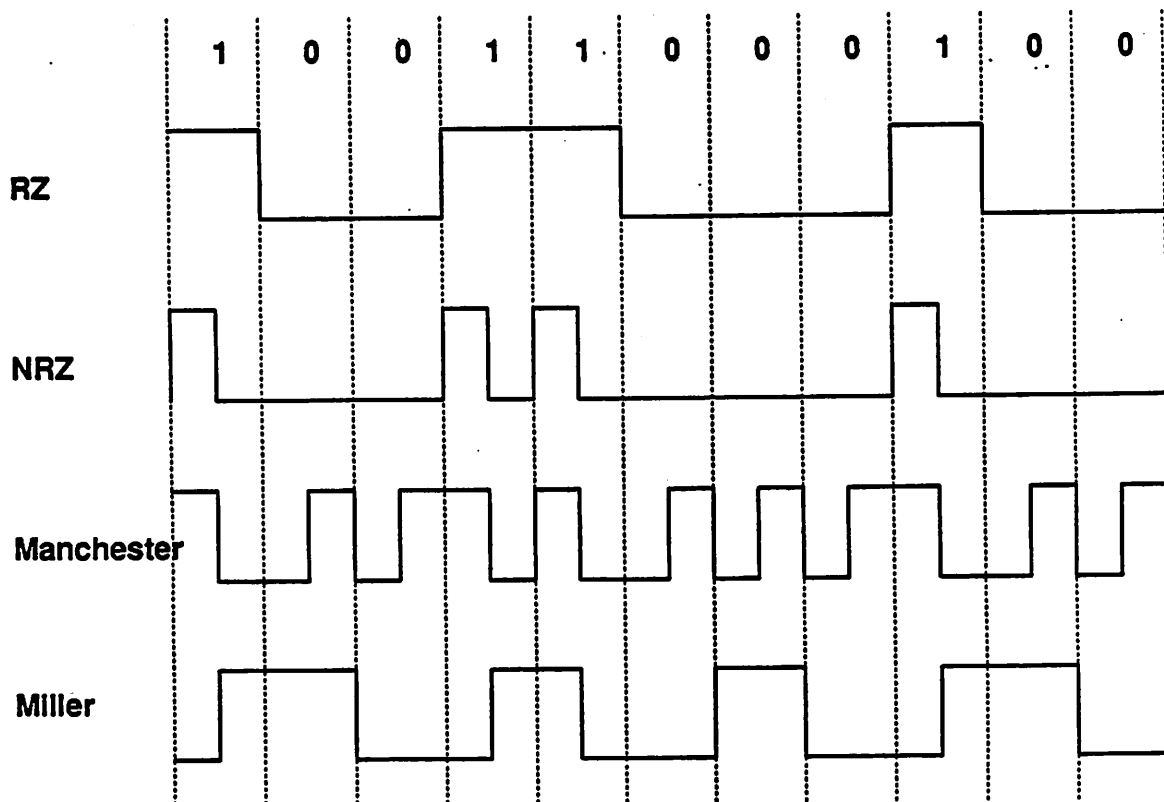


Figure 2.3 Pulse Shapes of Various Line Coding Schemes

NRZ is the simplest code, it also makes efficient use of bandwidth. The main limitations of NRZ are the presence of a dc component and the lack of synchronization capability. RZ is basically the same as NRZ except it occupies more bandwidth. Manchester code uses more bandwidth to ensure a predictable transition in the middle of each interval to remove the dc component. Also the receiver can use this information for clock recovery and error detection. In the case of Miller code the code can be represented through a modulation with memory to deliberately achieve a higher bandwidth efficiency and the minimum dc component.

2.3.1.2 Intersymbol Interference

Even in broadband transmission, dispersion causes degradation of the pulse, called *intersymbol interference* (ISI). For example as shown in figure 2.4, in the case where the transmission line can be represented by a simple RC network, the transmitted signals experience trailing type dispersion and the shapes are degraded.

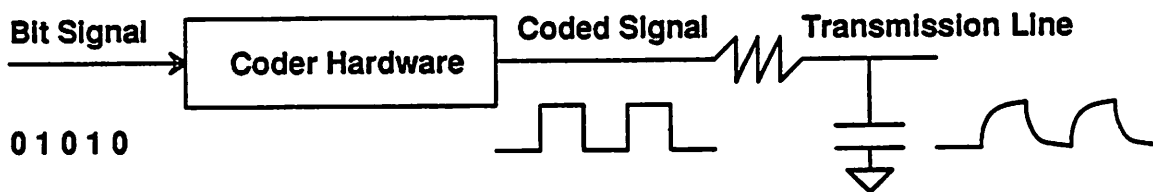


Figure 2.4 Signal Degradation Caused by Dispersion

To quantify the degradation of the signal, a graphical illustration called an *eye diagram* is used. Figure 2.5 shows the eye diagrams for the ideal pulse and degraded pulse. In an eye diagram, as the degradation gets worse, the center of the eye becomes narrower and the probability of transmission error increases. For a channel with limited bandwidth, since the signal degradation is very severe, pulse shaping often using

Nyquist pulses [LEE], is required to reduce the intersymbol interference. In broadband transmission systems, while the intersymbol interference is not as severe, jitter caused by signal noise on the degraded pulses is the major obstacles to error-free transmission.

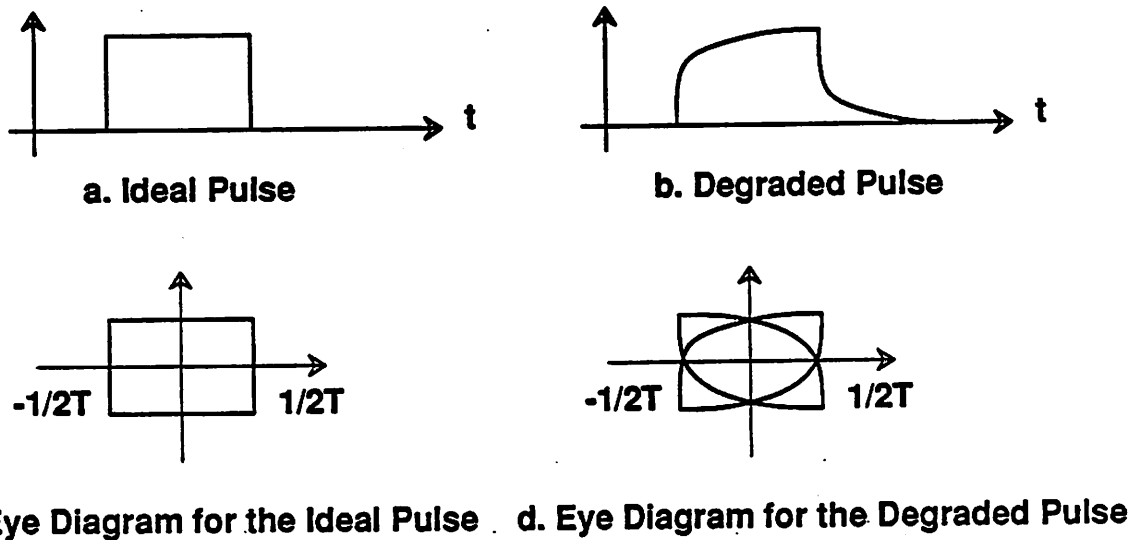


Figure 2.5 Effect of Pulse Degradation on Eye Diagram

2.3.1.3 Input Phase Jitter

The received signal always contains noise generated from a number of noise sources such as thermal noise, crosstalk, shot noise and so forth. The noise affects the shape of signals and pulse position. In general, an ideal clock recovery circuit takes a sequence of time positions at which a rising or falling edge of a pulse crosses some threshold voltage and averages the time positions to derive the real input pulse position. This averaging process makes the clock recovery circuit insensitive to position variation of the edge for input signal edges, referred to as *jitter*. However there is a limited amount of input jitter that a real clock recovery circuit can tolerate. Therefore insensitivity to input jitter is a very critical requirement for clock recovery circuits and

decision circuits. Figure 2.6 shows the input jitter and eye diagrams when the input signal has jitter.

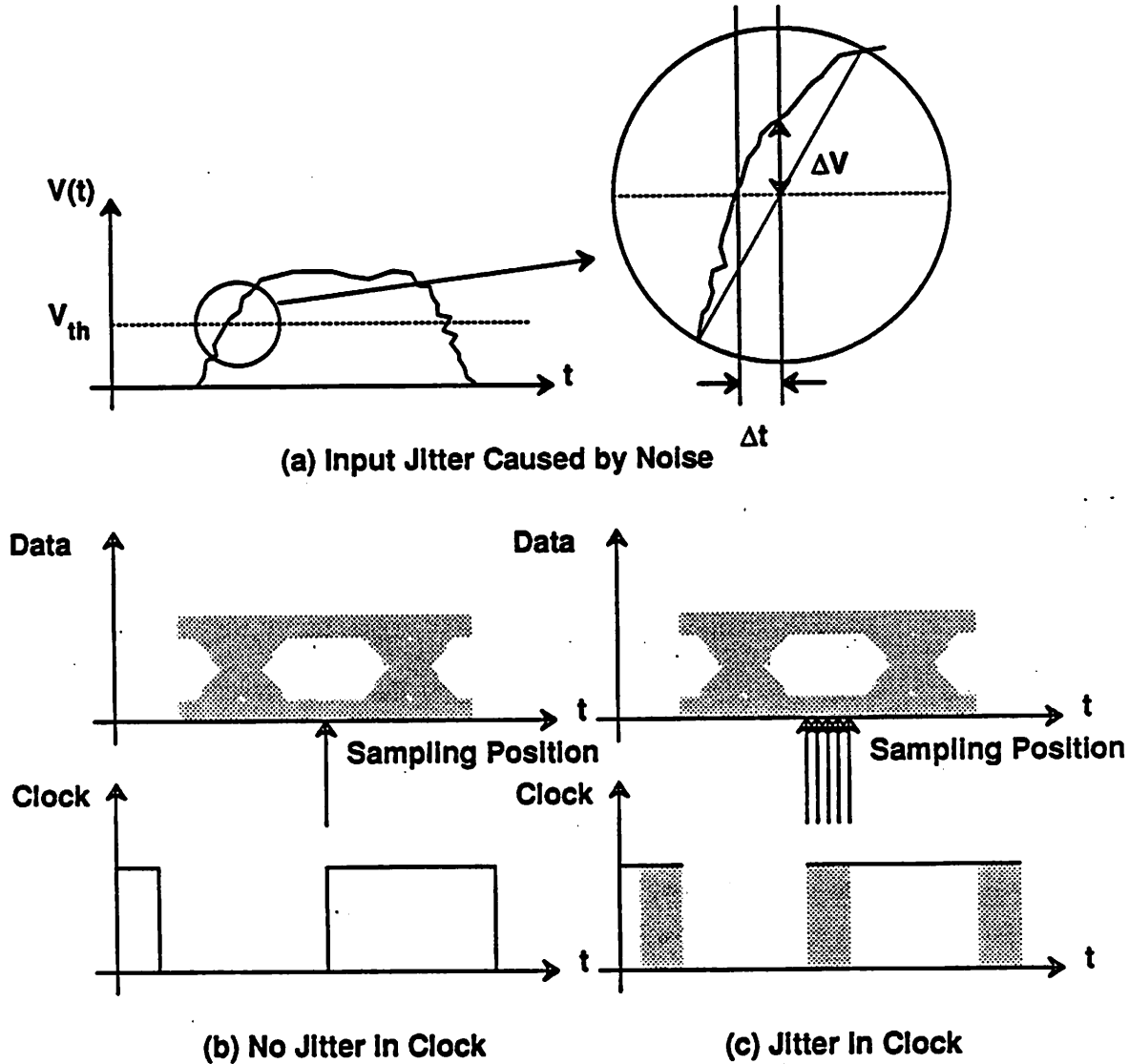


Figure 2.6 Input Jitter and Eye Diagram

Here in the figure 2.6.(a), the time position of rising edge crossing the threshold voltage is shifted to the left from the ideal position because the noise changes the voltage near the threshold voltage. Also the center opening of the eye diagram is narrower as the noise gets worse (figure 2.6.(b)). This indicates that the probability of

data transmission error increases because the signal-to-noise ratio decreases with narrowing of the eye opening. Note that here the clock is assumed to have no jitter. In a real situation, clock recovery circuits generate clocks containing some jitter. Therefore the sampling point also moves around the center of the eye and the error probability also increases (figure 2.6.(c)).

2.3.2 System Performance Parameters

In this section, the problems of clock recovery circuits are described. The first sub-section discusses clock jitter, the second deals with acquisition speed, and the last one describes static phase offset and stability.

2.3.2.1 Clock Phase Jitter

Similar to the input signals, clock signals also contain jitter caused by several noise sources such as thermal noise, shot noise, power supply noise, input jitter and so forth. As shown previously in figure 2.6, if the clock signal has jitter, the sampling moment of the data is wandering which results in large data transmission errors through the channel. This phenomenon gets worse as the clock edge moves further away from the center of the eye. Here, one bit interval (from the figure, one pulse interval) is called a *decoding window*. The jitter aspect of the performance of clock recovery circuits can be characterized by the bit error rate as a function of decoding window as shown in figure 2.7.

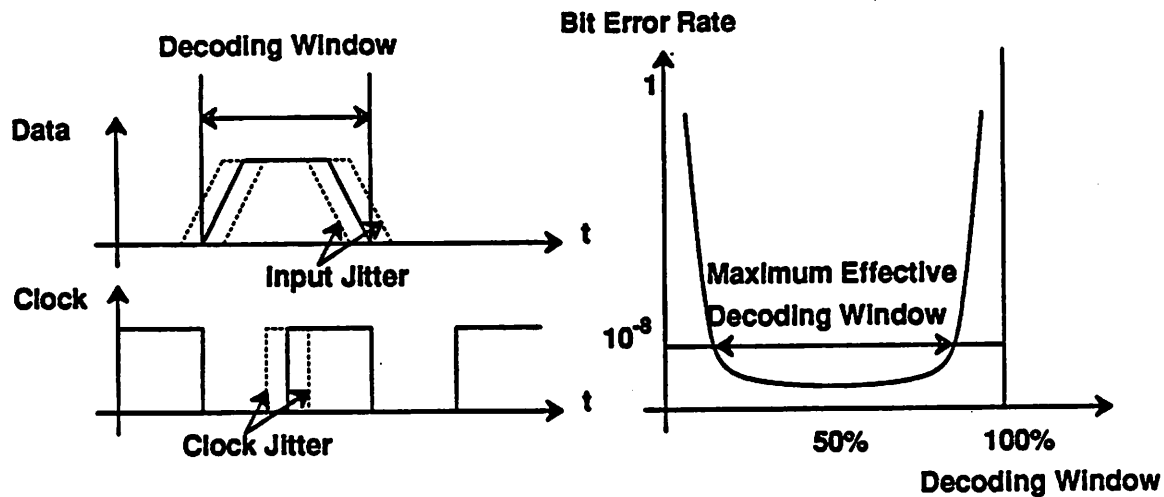


Figure 2.7 Bit Error Rate vs Decoding Window

In the vicinity of the wall of the decoding window, the bit error rate (the probability of false decision) approaches 1, and in the center region of the decoding window, the bit error rate is minimum. Here the distance between two crossing points with the some probability threshold is defined as the *maximum effective decoding window* . In typical disk drive applications, a window width of 90% or more is required at the 10^{-8} bit error rate. In digital transmission channels, the threshold is usually 10^{-8} , but a smaller window width is allowed because of smaller input jitter.

2.3.2.2 Acquisition Speed

In most data transmission channels, the clock information should be derived from the incoming data bit stream. When the clock recovery circuit begins reading a new data record, the clock recovery circuit does not know the frequency and the phase of the incoming data bit stream. Therefore, it is necessary to have some "grace period" to acquire the phase and the frequency and to synchronize the data. For this period, called *preamble period*, the transmitter usually send some regular sequence of pulses, for example 1 0 1 0 1 0 ..., in order to help the clock recovery circuit acquire the phase

and frequency of the pulses. This signal pattern is called a preamble sequence. Figure 2.8 shows the preamble sequence and the clock synchronization. The acquisition speed of the clock recovery circuit determines the length of the preamble sequence to be sent by the transmitter. Therefore to reduce the time spent for this preamble, the clock recovery circuit should be capable of fast acquisition.

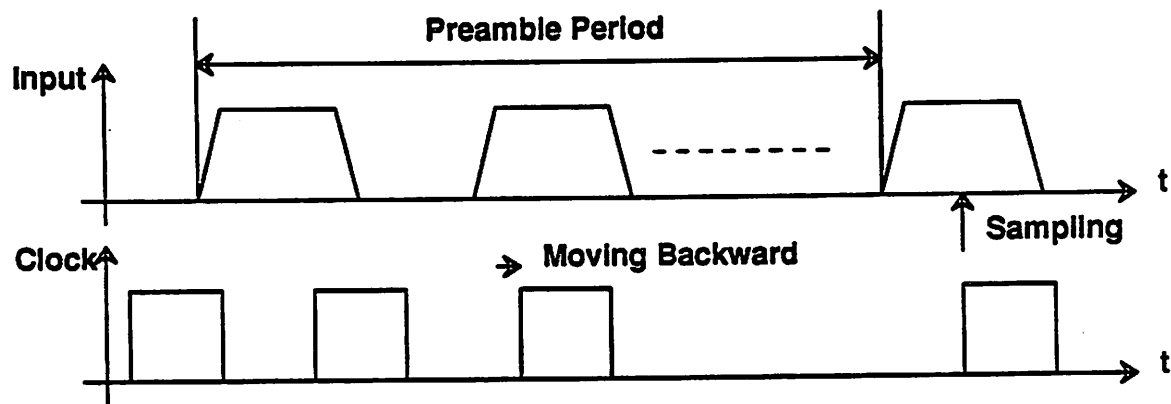


Figure 2.8 Definition of Acquisition Speed

2.3.2.3 Static Phase Offset and Stability

Due to circuit non-ideality and offset, the derived clock does not have its rising edge in the center of the eye but is shifted to the right or left a little bit. This shift is called *static phase offset* and represents another source of data transmission error. Therefore this static phase offset should be small enough. Figure 2.9 shows the static phase offset.

High performance clock recovery circuits usually incorporate a phase-locked loop. The loop may be unstable if a certain stability criteria are not met (this will be described in detail in later chapters) because the loop can be thought of as a negative feedback loop. If the loop is not stable, the position of the clock signal will oscillate and the PLL will fail to acquire the phase and the frequency of the incoming data bit

stream. Thus, stability consideration of the clock recovery circuit is another important factor to achieve a high performance system.

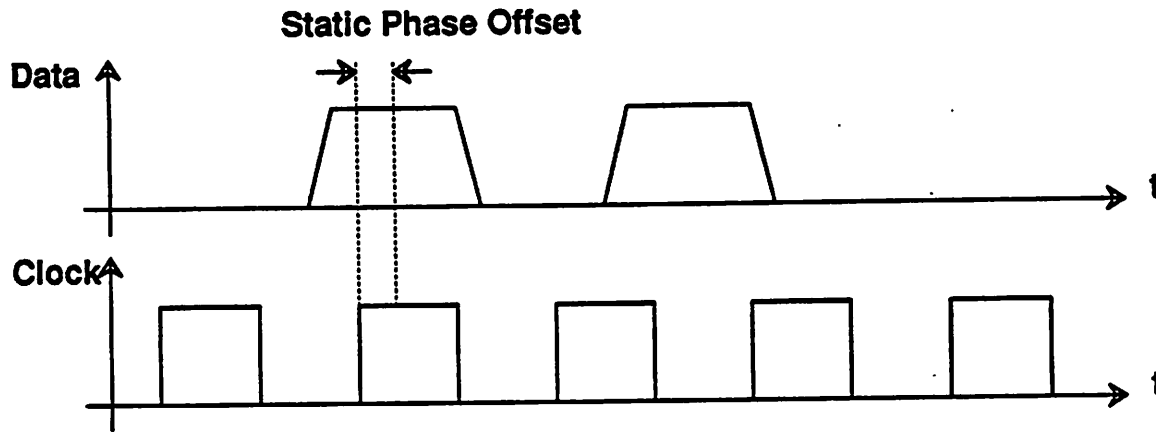


Figure 2.9 Static Phase Offset

Chapter 3

Clock Recovery Architecture Review and Comparison

3.1 Introduction

Clock recovery techniques have been studied since the appearance of data communications. Many ideas have been suggested to implement clock recovery circuits for several different applications. In general, clock recovery techniques can be divided into two broad classes depending on the application. The first category is *open loop clock recovery*, in which the incoming signal is passed through a nonlinear device and a frequency shaping circuit, and the resulting signal is used to sample the incoming signal to extract the carried information. This method does not have a feedback loop, therefore, it is simple and is used in high frequency clock recovery applications. The spectral line method is an example of open loop clock recovery. The second class of the techniques is *closed loop clock recovery* with a feedback loop, in which symbol timing error is estimated and the error is fed back to a voltage controlled oscillator (VCO). The VCO is then controlled to adjust the sampling instant to the middle of the symbol interval.

The closed loop clock recovery method is characterized by several factors; one of the principal factors being the method used to determine the phase error. On this basis, closed loop clock recovery techniques may be divided into two further categories. The first one may be referred to as *continuous-time clock recovery*, and the second *discrete-time clock recovery*. The continuous-time clock recovery method generates continuous-time phase error signals from a multiplier or some logic circuits, and these signals are directly used to control a VCO. In contrast, the discrete-time clock recovery technique samples the input signals with a VCO clock creating phase

errors in the discrete-time domain, and controls the VCO after doing subsequent signal processing in the discrete-time domain. Continuous-time clock recovery circuits are usually used in broadband data communication systems such as local area networks (e.g. Ethernet, Token Rings), optical communication networks, and disk drive read channels which do not suffer large signal degradation. Discrete-time clock recovery is used in low speed applications for band-limited data communication systems through telephone lines or narrow-band communication channels. Examples of the latter are digital subscriber loops and integrated services digital networks (ISDN).

In this chapter, closed loop clock recovery techniques are emphasized. Open loop clock recovery techniques are treated, briefly in section 3.2. Section 3.3 describes closed loop clock recovery techniques. Finally, in section 3.4, some comparisons among these techniques are presented.

3.2 Open Loop Clock Recovery Techniques

There are not many distinct open loop clock recovery techniques since the performance is relatively poor compared to closed loop clock recovery circuits. However, for very high frequency applications, this method is frequently used because of its simplicity.

3.2.1 Spectral Line Methods

One of the oldest methods, still used in microwave frequency range clock recovery, is the class of *spectral line methods*. This method is the simplest method of overall clock recovery and based on the clock tone present in the power spectrum of the incoming data bit stream.

We assume that the digital information is transmitted by weighted pulses, each with identical shape and spaced uniformly by T secs. If the impulse response of each

pulse is represented by $p(t)$, then the incoming baseband data pulse stream may be represented as

$$R(t) = \sum_{k=-\infty}^{\infty} A_k p(t-kT) \quad (2.1)$$

where A_k are the transmitted symbols. In binary transmission scheme, A_k can be either "-1" or "1". When the mean-value of the data symbol is non-zero, the baseband pulse stream contain a spectral line at the baud rate. To determine whether a spectral line at the baud rate exists from the data pulse stream, $R(t)$ can be represented by the following equation.

$$R(t) = E[A_k] \sum_{m=-\infty}^{\infty} p(t-mT) + \sum_{m=-\infty}^{\infty} (A_m - E[A_k]) p(t-mT) \quad (2.2)$$

The first term is independent of the data A_k , is periodic with period T, and can be thought of as a deterministic signal. Its periodicity implies a fundamental at the baud rate $2\pi/T$. This fundamental frequency can be extracted with a bandpass filter. The second term is zero-mean and random, and results in jitter on the recovered clock tone. This method is called *linear spectral line method*. However, in most case, the mean-value of data symbols is zero, there is no strong clock tone in the incoming data bit stream. But often the second and higher moments of the incoming signal are non-zero and periodic. Therefore, to derive a clock tone, the incoming signal is passed by a nonlinear component (such as squarer) and filtered by a bandpass filter. These methods are called *nonlinear spectral line methods*. To illustrate this, a squarer device is used as an example. If the correlation function of the data symbols is expressed as (2.3),

$$E[A_m A_n] = \sigma_A^2 \delta_{m-n} \quad (2.3)$$

the expected value of the output of the squarer device can be expressed as (2.4).

$$E[R(t)^2] = \sigma_A^2 \sum_{m=-\infty}^{\infty} p(t-mT)^2 \quad (2.4)$$

This expectation value can be considered a deterministic component of the output of squarer device and it is periodic with period T . The fundamental of this signal will be extracted by the bandpass filter. Figure 3.1 shows the block diagram of this approach.

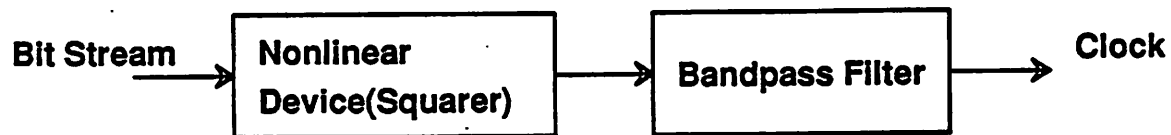


Figure 3.1 Nonlinear Spectral Line Method

The primary advantage of this approach is in its simplicity. Since it does not require any complex circuitry inside, it can be used in high speed clock recovery circuits. In microwave frequency range clock recovery, this method can be easily implemented by nonlinear device and a bandpass filter. However, since the passband of the bandpass filter is not usually realized accurately enough, the noise rejection is some times poor, therefore the phase jitter caused by this noise is usually considerable. Also, the entire clock recovery depends on the simple bandpass filter: there is no good way to control the static phase offset and the acquisition time. As a result, it require a long preamble period and displays large phase errors due to the nonlinear device and the bandpass filter.

3.3 Closed Loop Clock Recovery Techniques

As mentioned, the closed loop clock recovery methods are divided into two categories based on how phase errors are processed. Section 3.3.1 describes continuous-time clock recovery techniques, and section 3.3.2 explains discrete-time clock recovery techniques.

3.3.1 Continuous-Time Clock Recovery Methods

Almost all clock recovery circuits in this category use a PLL. Based on different PLL topologies, several approaches are presented. Here, two examples, *early-late methods*, and *transition-detection based methods* are illustrated in section 3.3.1.1 and 3.3.1.2 respectively.

3.3.1.1 Early-Late Methods

Early-late method is very popular for several broadband transmission applications. Figure 3.2 shows the block diagram for an example. In this diagram, the input signal is integrated by a pair of gated integrators, each performing its integration over a time interval of $T/2$. Integration by the early gate occurs in the $T/2$ preceding the nominal location of data transitions and the output is held by the following sample and hold circuit for the next $T/2$. While the late gate integrates during the $T/2$ immediately following the transitions and the following sample and hold circuits hold the output for the next $T/2$. If timing error is zero, the data transitions fall exactly on the boundary between early and late gates.

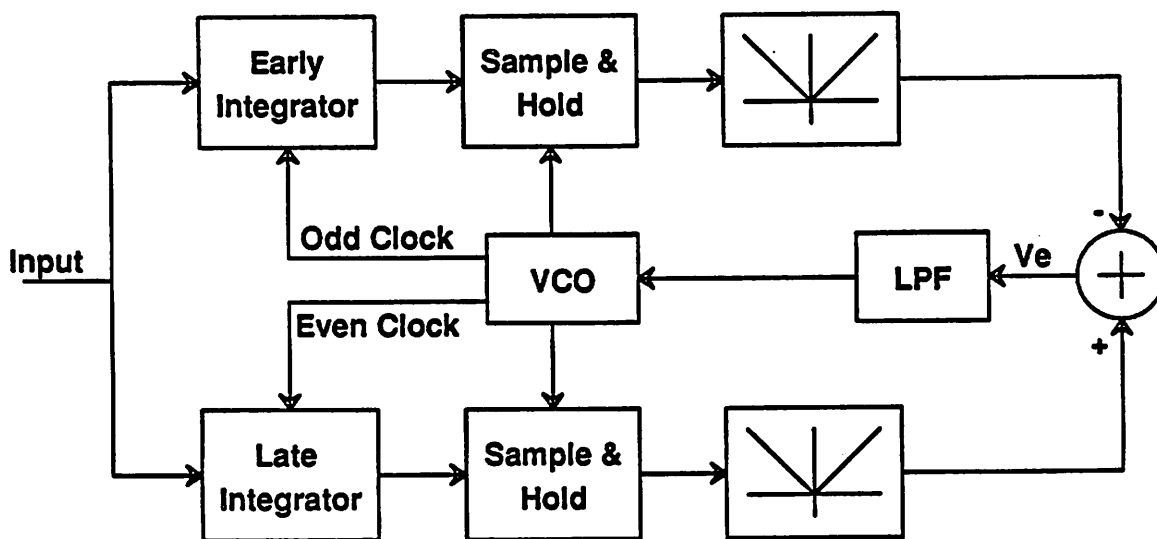


Figure 3.2 Early-Late Method

If the input leads the VCO clock, the rectified output of early gate sample and hold circuits will generate smaller voltage than that of late gate sample and hold circuits. Therefore the lowpass filtered output of the error voltage will increase and the VCO clock will go faster and in phase. Figure 3.3 shows the waveforms for this case.

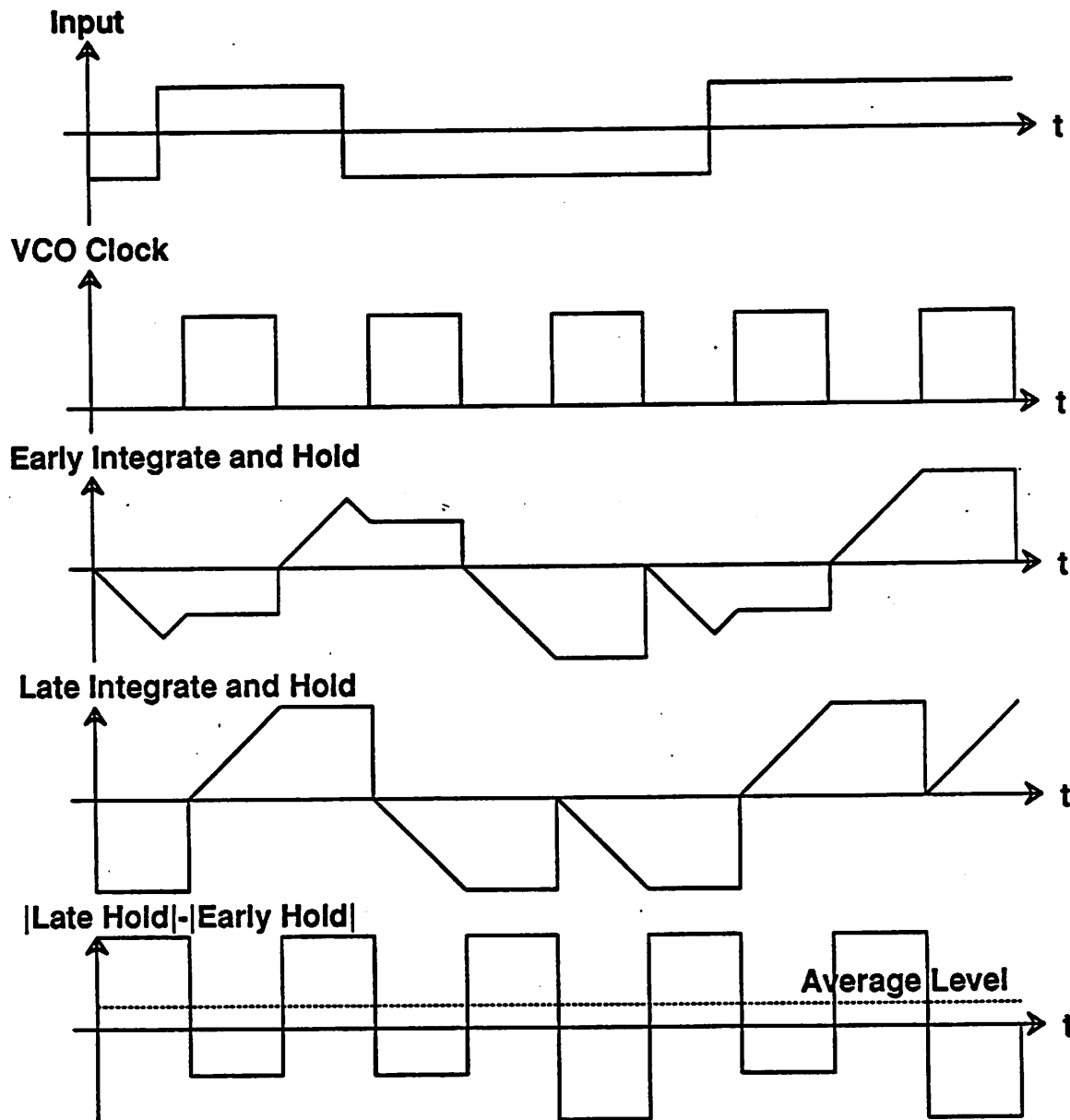


Figure 3.3 Clock Waveforms for Early-Late Method

In figure 3.3, the VCO clock rising edge lags behind that of the input signal. The early gate integrates the input signal for the early-half period of the clock signal, and holds it for the remainder of the clock period. On the other hand, the late gate integrates the input signal for the late-half period of the clock signal and hold it for the next early-half period. If the absolute values of the late gate hold signals are taken for the early-half period of the clock signals, and the negative of the absolute values of the late gate hold signals are taken for the late-half period of the clock signals, the resultant signals are represented by figure 3.3. The averaged value of these signals generated from a lowpass filter gives the control signal to adjust the phase of the VCO clock signals.

As a summary, when the transition occurs, the output of each rectified sample and hold circuit shows the different voltage level. For some time, these differences are accumulated through the lowpass filter which results in shifting of VCO clock pulse to the left by increasing the frequency of VCO. But if no transition occurs, no voltage difference is generated from the subtractor, therefore there will be no change of the VCO frequency. Basically, this technique can be used in narrowband transmission systems if the integrators are removed. But it requires 2 X higher sampling than baud-rate sampling techniques which are widely used in narrowband transmission systems.

As mentioned earlier, this method is popular for rectangular pulses. Since this method relies on the shape of the pulses, a large static phase offset can be introduced if the pulse shape is not symmetric. In high speed applications, the two integrators are required to settle very fast, which limits the operating speed of the clock recovery circuit. Also the acquisition time is not easily controlled.

3.3.1.2 Transition-Detection Based Methods

The most widely used technique for clock recovery in high performance wide-band data transmission application is the *transition-detection based method*. There are many variations of these methods depending on the exact implementation of each PLL loop component. However, the basic configuration is the same for almost all implementations and shown in figure 3.4.

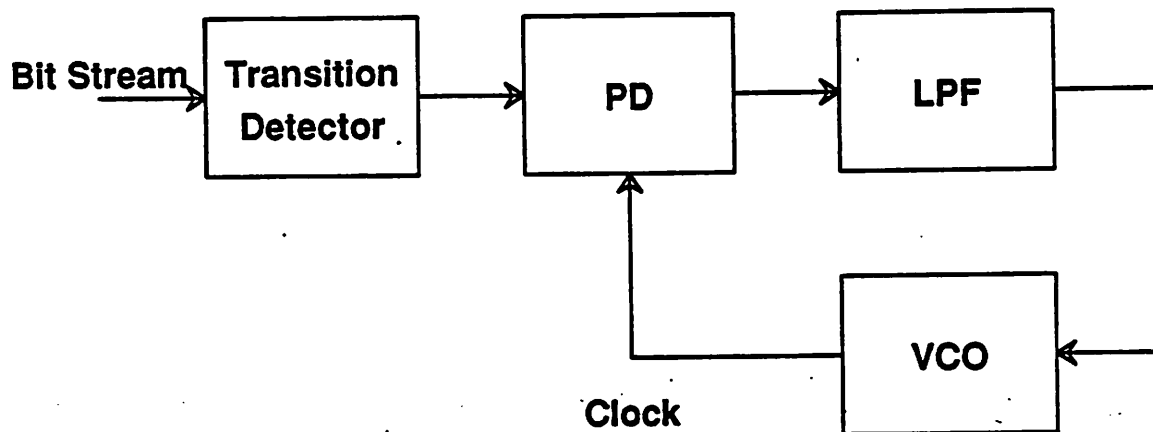


Figure 3.4 Transition-Detection Based Clock Recovery Method

Usually, the input is transmitted using NRZ line coding. Since the NRZ signal does not have a frequency component at the baud rate, it is necessary to change the NRZ signal to another form of signal which has a frequency component at the baud rate. The transition detector in figure 3.4 does this function. It detects the transition edge and generates a pulse corresponding to the transition edge (both low-to-high and high-to-low transition). In effect, it translates the NRZ signal to RZ signal which has a strong frequency component at the baud rate. Then, the phase detector compares the RZ signals with the VCO clocks. For example, a simple multiplier can be used as a phase detector. In this case, the VCO clock is shifted by 90 degrees to the left when the phase detector generates output signals with the zero dc components (when the

PLL is locked into input signal). If the VCO clock is shifted by more than 90 degree to the left, the output signals with a negative dc component are generated from the phase detector and the VCO clock is adjusted to the right. The lowpass filter extracts the dc component from the phase detector output. Figure 3.5 shows the output waveforms of each block.

This method has many merits. First, there is no slow circuit in the loop, therefore it can be used for high speed clock recovery. Also, since the clock recovery circuit can use the pulse transition edge to derive the clock information without relying on the pulse shape, static phase offset can be reduced. However, the static phase offset still depends on the precision matching on the loop components and the path delays. Also, fast phase acquisition requires complex analog design and many external components. Conventional digital PLL's can solve those problems, however these are not practically usable because of large phase jumps normally associated with them. Currently, many clock recovery circuits for the high performance applications are built using analog PLL based method with zero-phase start and gearshifting techniques.

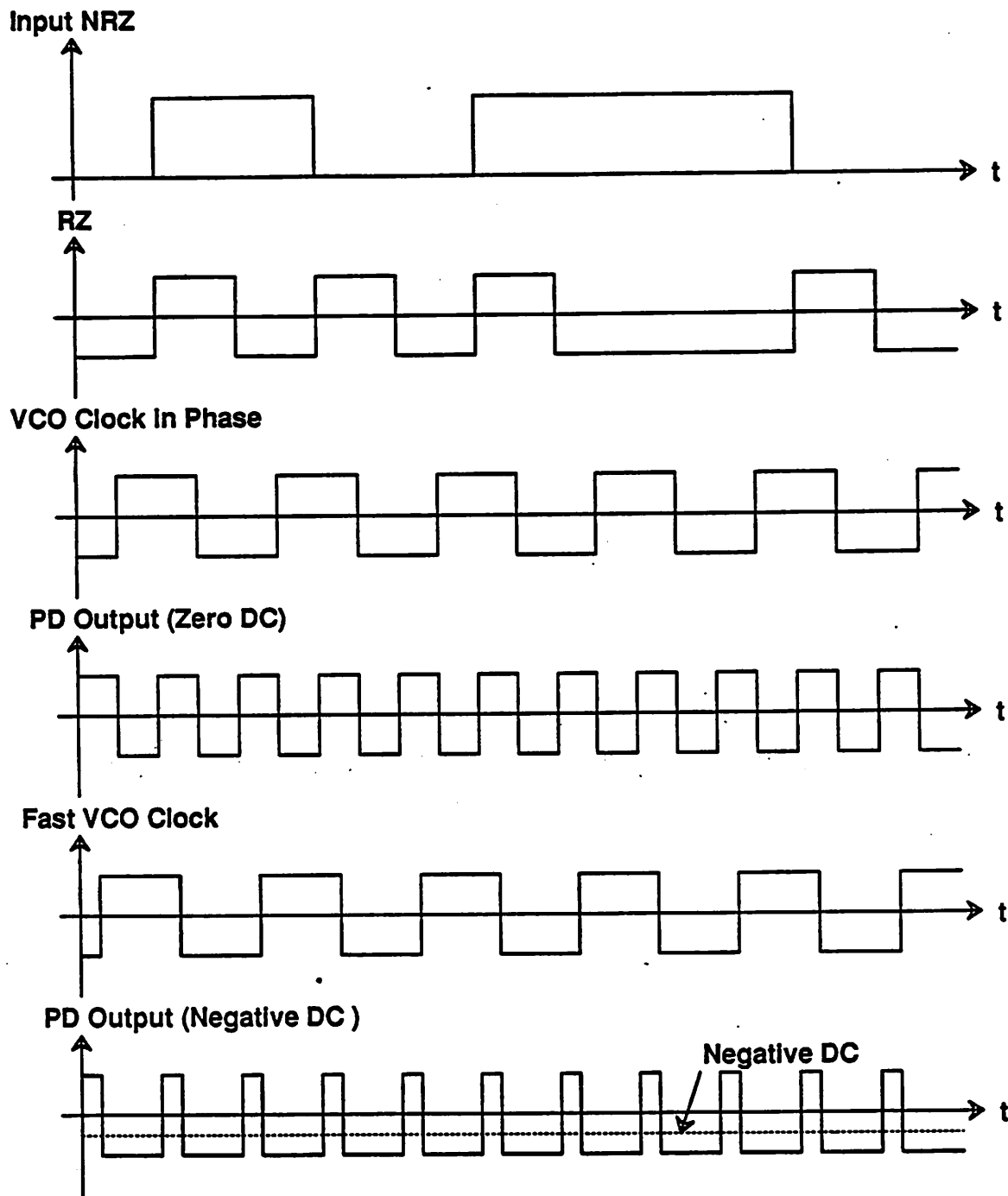


Figure 3.5 VCO Clock and Phase Detector Output Waveforms

3.3.2 Discrete-Time Clock Recovery Methods

In narrowband data transmission system such as a digital subscriber loop through a telephone line, the received data waveforms are degraded very much due to pulse dispersion, noise and echo. Therefore, sophisticated signal processing is involved and it is not feasible to use the simple clock recovery circuits describes earlier in section and 3.2 and 3.3.1. For example, in the full-duplex digital subscriber loop transceiver shown figure 3.6, the clock recovery circuit does not take the incoming data directly, but takes it after receive filter, sampler and echo canceler. Thus the signal being sampled is not the square pulse that is usually available in the broadband transmission system, but dispersed pulses (if ideal, Nyquist pulses). The *minimum mean-square error* (MMSE) and the *baud-rate sampling* techniques are the good candidates for such system.

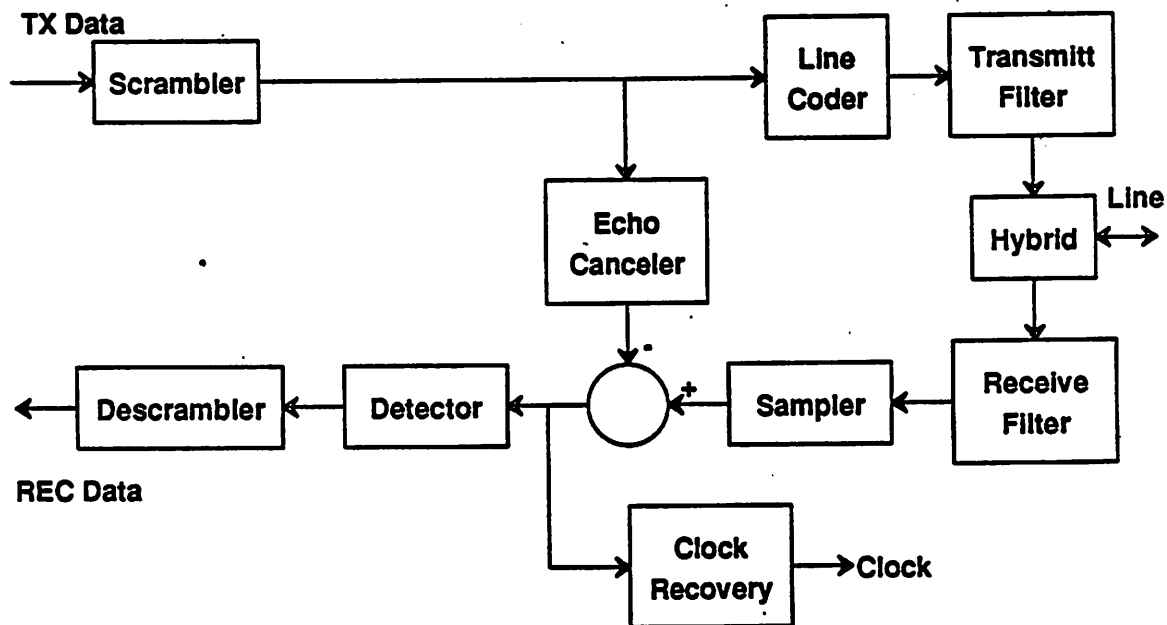


Figure 3.6 A Full-Duplex Digital Subscriber Loop Transceiver

3.3.2.1 MMSE Clock Recovery

In figure 3.7, the received signal is sampled at times $kT + \tau_k$. Here, the symbol interval is T , and the timing error is represented by τ_k . If the output of the receive front end filter is $Q_k(\tau_k)$, and the correct data symbol is A_k , the MMSE clock recovery circuit adjusts τ_k to minimize the expected square error between the input to the slicer and the correct symbol,

$$E[|E_k(\tau_k)|^2] = E[|Q_k(\tau_k) - A_k|^2] \quad (3.5)$$

Since the input to the slicer $Q_k(\tau_k)$ is a complicated nonlinear function of the timing phase τ_k , the expected squared error is minimized by adjusting the timing phase in the direction opposite the derivative of the expected value of the squared error. Then the new timing phase is

$$\tau_{k+1} = \tau_k - \alpha \operatorname{Re} \left\{ E_k^*(\tau_k) \frac{\partial Q_k(\tau_k)}{\partial \tau_k} \right\} \quad (3.6)$$

To realize the derivative function using the discrete-time filter, the derivative is approximated by (3.7).

$$\frac{\partial Q_k(\tau_k)}{\partial \tau_k} = (Q_{k+1}(\tau_k) - Q_{k-1}(\tau_k)) / T \quad (3.7)$$

Then, (3.6) becomes (3.8).

$$\tau_{k+1} = \tau_k - \alpha' \operatorname{Re} \left\{ [Q_k(\tau_k) - \hat{A}_k] [Q_{k+1}(\tau_k) - Q_{k-1}(\tau_k)] \right\} \quad (3.8)$$

where the α' is $\frac{\alpha}{T}$ and the \hat{A}_k is the slicer output replacing the actual symbol A_k in the decision-directed tracking period. In the training period, the actual known symbols A_k are available and are used to aid phase acquisition.

One problem of this method is that since $E_k(\tau_k)$ is not linear function of τ_k , the stochastic algorithm on which this method relies is not guaranteed to converge to the

optimal timing phase. Therefore, in practice, some other techniques should be used in the acquisition time to get reasonably good timing phase. Also, this method requires Nyquist sampling which is 2 X higher than the baud-rate sampling because of the discrete-time differentiator realization. The next clock recovery circuit, *baud-rate sampling techniques* allow baud-rate sampling.

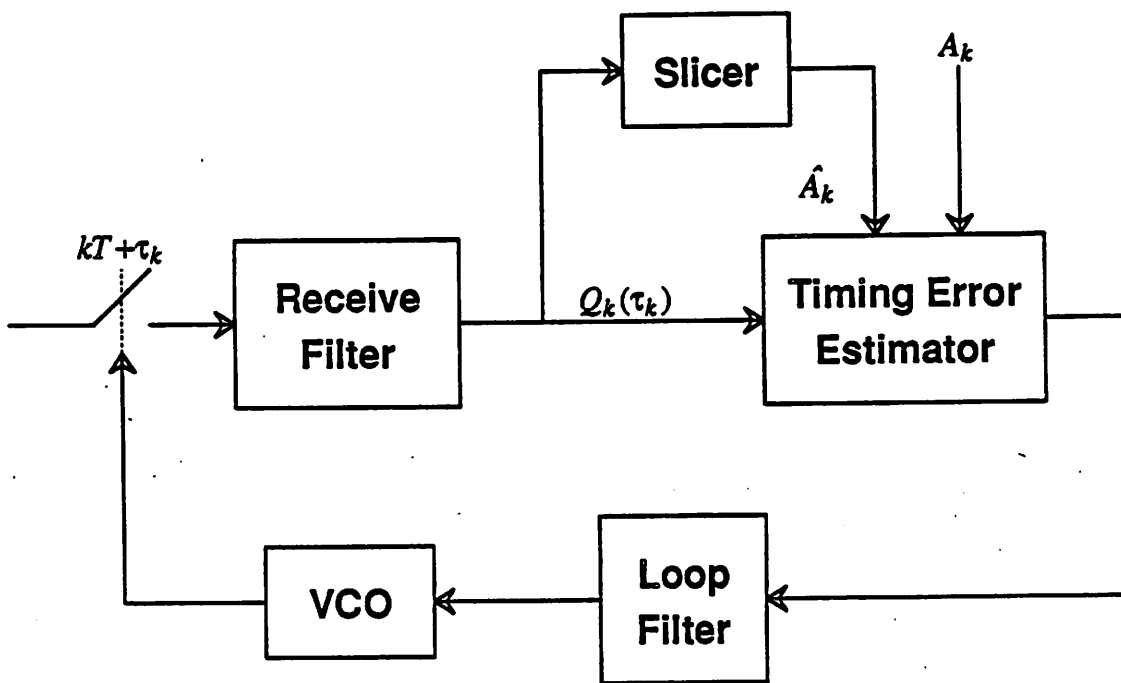


Figure 3.7 MMSE Clock Recovery Techniques

3.3.2.2 Baud-Rate Clock Recovery

There are several baud-rate clock recovery techniques using discrete-time clock recovery methods. One of the techniques derived from MMSE techniques are as follows. If the timing function to update the timing phase is defined as (3.9).

$$f(\tau_k) = -\text{Re} \left\{ E [Q_k(\tau_k) - \hat{A}_k] [Q_{k+1}(\tau_k) - Q_{k-1}(\tau_k)] \right\} \quad (3.9)$$

Then, $f(\tau_k)$ becomes (3.10) under the assumption that $Q_k(\tau_k)$ is a wide sense

stationary (WSS) random process.

$$f(\tau_k) = \text{Re} \left\{ E[\hat{A}_k Q_{k+1}(\tau_k)] - E[\hat{A}_k Q_{k-1}(\tau_k)] \right\} \quad (3.10)$$

For example, if the continuous time PAM signal at time $kT + \tau_k$ is (3.11).

$$Q_k(\tau_k) = \sum_{m=-\infty}^{\infty} A_m p((k-m)T + \tau_k) + N_k \quad (3.11)$$

where the A_k and N_k are WSS random processes and N_k is zero mean and independent of A_k . Then, the timing function is (3.12).

$$f(\tau_k) = E[|A_k|^2][p(\tau_k + T) - p(\tau_k - T)]. \quad (3.12)$$

If $p(t)$ is symmetric about τ_k , the timing function will be zero at $f(0) = 0$.

This method is very attractive for the narrowband clock recovery circuits because the nature of the baud-rate sampling helps the design of the echo canceler and equalization filter which are commonly used in the digital subscriber loops.

3.4 Comparison of Clock Recovery Techniques

In this section, qualitative comparisons among the clock recovery methods described above are made based on applications and performance. First, the clock recovery methods are divided into several categories. Then, application areas are defined. Next, the clock recovery circuits are divided based on the operating speed, acquisition speed, input jitter tolerance, complexity and some examples are given. Table 3.1 summarized these. Here, SLM, ELM, TDM, MMSEM, and BRM represents Spectral Line Methods, Early-Late Method, Transition-Detection based Method, Minimum-Mean Square Estimation Methods, and Baud-Rate clock recovery Method, respectively.

Table 3.1 Comparison of Clock Recovery Methods

Name	SLM	ELM	TDM	MMSEM	BRM
Method	Open	Closed	Closed	Closed	Closed
Phase Error	Continuos	Continuous	Continuous	Discrete	Discrete
Operating Range	Broadband	Broadband	Broadband	Narrowband	Narrowband
Acquisition Speed	Slow	Slow	Fast	Slow	Slow
Jitter Tolerance	Low	High	High	Low	Low
Complexity	Simple	Simple	Simple	Complex	Complex
Examples	Microwave	LAN	LAN, Disk	Modem, ISDN	Modem, ISDN

Chapter 4

Hybrid Analog/Digital PLL

4.1 Overview

As mentioned earlier, many alternatives have been studied to achieve high performance clock recovery circuits for several different applications ranging from relatively slow digital subscriber loops to high speed optical data communications. However, one of the most challenging areas for high speed clock recovery circuits is the disk drive read channel for the following reasons.

1. When compared with other communication channels, the bit stream coming from the magnetic medium has large amount of jitter coming from inter-track modulation, pulse dispersion, offset, and power supply and thermal noise. Therefore, the circuit should tolerate large input jitter.
2. Since the preamble time for the disk drive increases the disk searching time, the clock recovery circuit should possess fast acquisition time in order to reduce the preamble time.
3. The data bit rate in the disk drive read channel is approximately from 15 MBits/sec to 60 MBits/sec. Therefore the clock recovery circuits should operate at these high speed.
4. The high performance disk read channel requires several complex functional blocks which can handle such features as the multiple frequency tracking, window shift and shrink, and pulse pair compensation etc. which makes the design of the clock recovery circuit difficult.

Figure 4.1 shows the typical disk drive read channel using a pulse detection scheme. Here, the disk heads reads the data stored in magnetic medium through

magnetic flux changes and sends small electric signals to the preamplifier. The output of the preamplifier is amplified through an AGC to a proper voltage level and filtered by a pulse slimming filter. The peak detector detects the peaks of the signal and generates pulses corresponding to these peaks. Then, the pulses coming from the peak detector are used to derive the clock information and to detect data through a clock recovery circuit called a *data separator*.

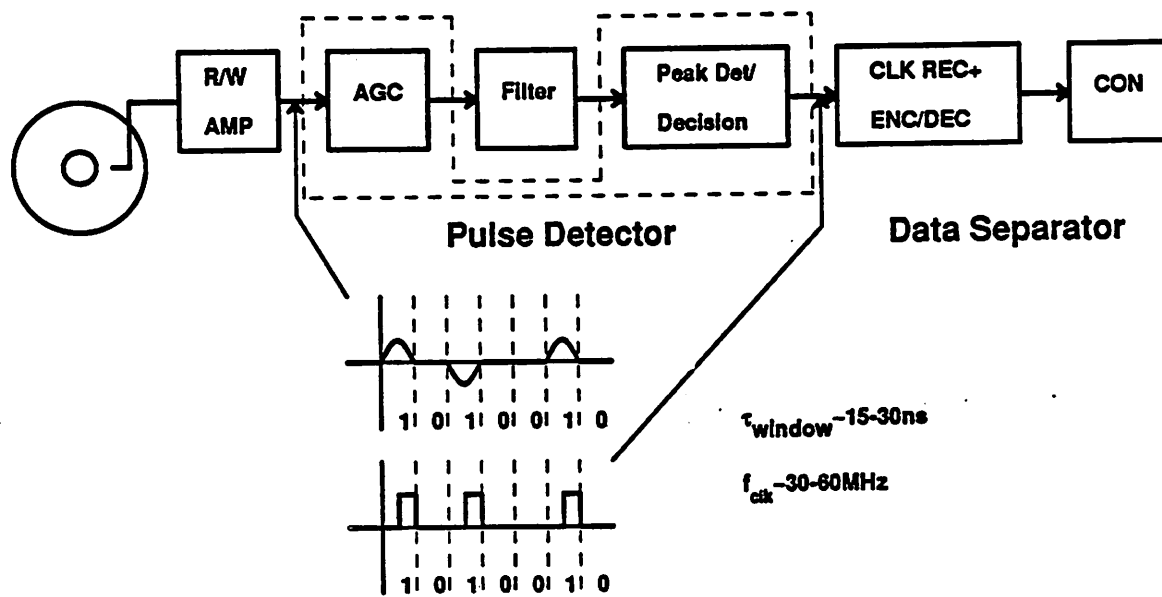


Figure 4.1 Typical Disk Drive Read Channel

There are several requirements for this data separator. When the disk drive begins reading a new data record, the clock recovery circuit must align its local clock with the incoming data bit stream as rapidly as possible. Techniques of zero phase start and variable loop time constant are used to speed up the acquisition time, but acquisition time of 20-40 bit cycles is required for the current implementations. Since the acquisition time is waste time, it is desirable to reduce it. In steady state, the clock recovery circuit must decode data correctly in the presence of a large amount of jitter

and other circuit impairments. Therefore, static phase offset should be reduced and the decoding window width should be increased. For high performance disk drives such as zonal disk, the disk drive is required to track multiple frequency data bit streams. Also for system level tests and diagnostics for the disk drive system, the disk drive must be able to incorporate variable programming capabilities.

Currently most high speed data separators for disk drives make use of an analog PLL implemented in bipolar technology using emitter coupled multivibrator or starved ring oscillator VCO's. But these techniques are limited in their ability to implement the above capabilities and more sophisticated fast-acquisition algorithms at high speed, and also are not well-suited to CMOS implementation which is needed to achieve higher levels of integration and lower power dissipation in disk drive read channel electronics. There are several reasons why current clock recovery circuits is not well-suited to the CMOS implementation.

1. The static and dynamic phase error depends on precision matching of analog devices and path delays.
2. Fast phase acquisition requires complex analog design and often requires many external components.
3. Phase acquisition algorithm selection is limited to those that can be practically implemented in analog PLL's.
4. It is difficult to implement sophisticated techniques such as window shift and shrinks needed for high level system tests and diagnostics within the disk drive system.
5. The traditional digital phase-lock loops can handle the above four problems. However, they are not practically usable because of large phase jumps normally associated with them. i.e. granularity in the time domain is limited by the traditional DPLL architecture.

6. The speed of CMOS circuits is currently slower than that of bipolar circuits.

To overcome those problems and the limitations, a new architecture, namely the *hybrid analog/digital clock recovery circuits*, which can be used in disk drive read channel electronics, and which can easily configured for other high speed broadband clock recovery circuits is presented in this chapter. The hybrid clock recovery uses an analog PLL to generate multiple clocks for a 1 GHz effective sampling in a 2 μ m technology, and a digital PLL (actually digital signal processor) to process edge detection, acquisition, tracking and programming.

4.2 Architecture

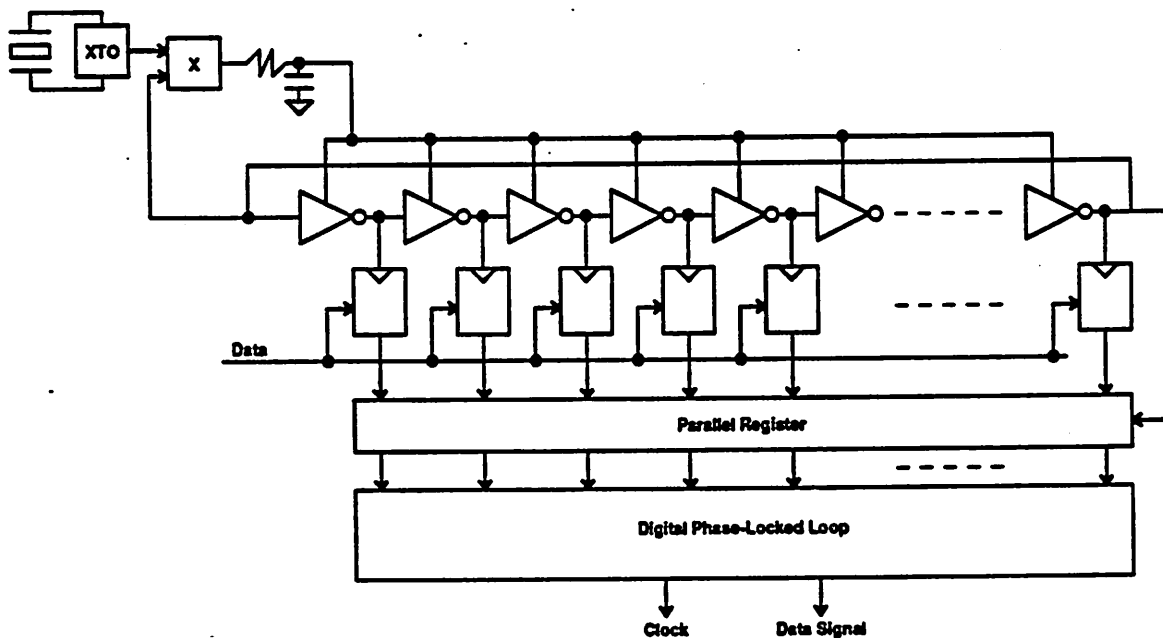


Figure 4.2 Hybrid Analog/Digital Clock Recovery Architecture

The basic concept of the clock recovery system is shown in figure 4.2. A ring oscillator composed of a long chain of inverters is permanently locked onto a reference clock at the data window rate, (the read reference clock in the case of a disk drive data separator). In the prototype described later, the ring oscillator composed of 16 differential stages having 32 taps on the ring is locked onto the 30 MHz reference clock with a conventional phase/frequency comparator with an on-chip charge pump and loop filter. Since the dynamics of this loop do not affect phase acquisition or tracking which happens in the digital part of the system, the analog PLL design is not critical. Each tap spaced one gate delay apart is used to latch the data samples into one of the 32 latches, so that at the end of one round trip (one bit interval), 32 samples spaced 1 ns delay apart (one gate delay is 1 ns) are stored in the 32 data latches. In a 2 μm CMOS technology, this gives an effective sampling rate of 1 GHz; in 1 μm CMOS technology, it is about 5 GHz. However, since these samples are sequential, the following logic circuits need to process the 1 GHz sampled data. This is not possible using current CMOS logic circuits. Therefore, a serial-to-parallel translator is necessary to reduce the data rate.

The parallel phase sampler, explained in section 4.2.2 in detail, accomplishes the data rate translation. In the parallel phase sampler, 1 bit, 1 GHz sampled data rate is translated into 32 bits, 30 MHz data rate. Figure 4.3 shows the data patterns stored in parallel phase registers.

After subsequently being moved to the holding register, the bit pattern is evaluated to determine the location of valid data transitions in the decode window using digital transition detectors presented in section 4.2.3. Noise filtering is easily applied to eliminate the effects of isolated noise pulse and transition sampling noise. The location of the data transition is then encoded into a binary number through a binary encoder.

The following digital PLL operation is purely digital and is based on the location of detected transition represented by a binary number through the binary encoder. Using the binary number, the digital phase-locked loop does phase acquisition and tracking. Since the phase information is processed by digital logic, complex data processing of the phase values are possible. The digital PLL is explained in section 4.2.3.

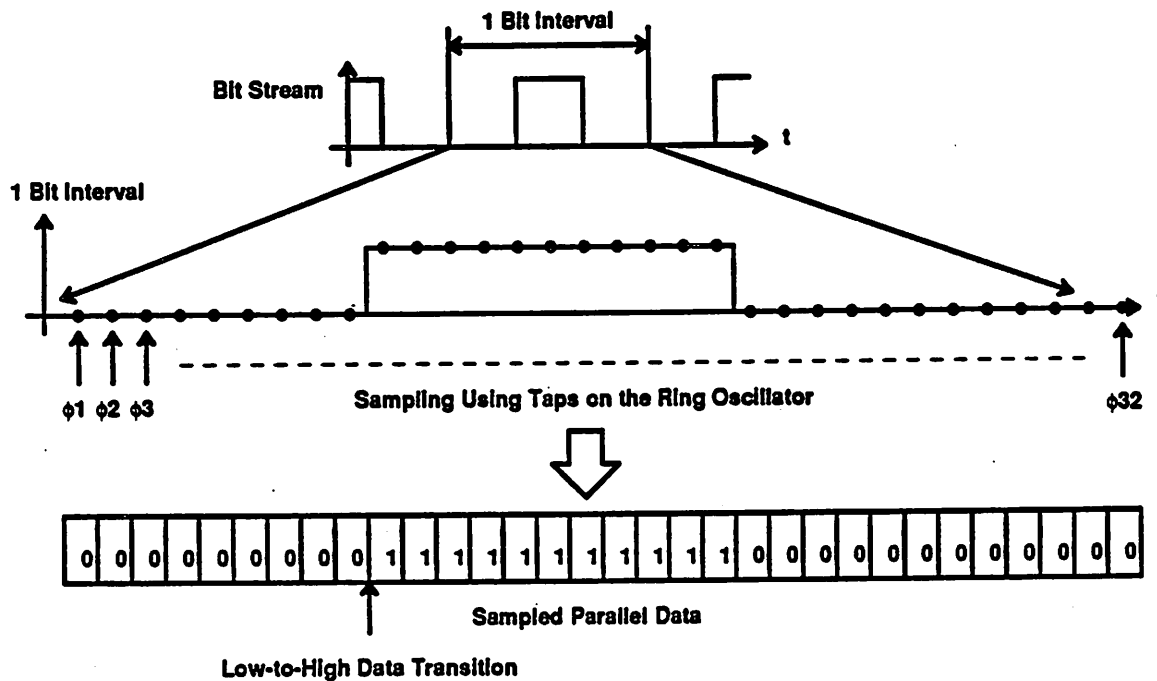


Figure 4.3 Sampled Date Pattern for One Bit Interval

4.2.1 Analog Phase-Locked Loop

Figure 4.4 shows the block diagram for the analog phase-locked loop.

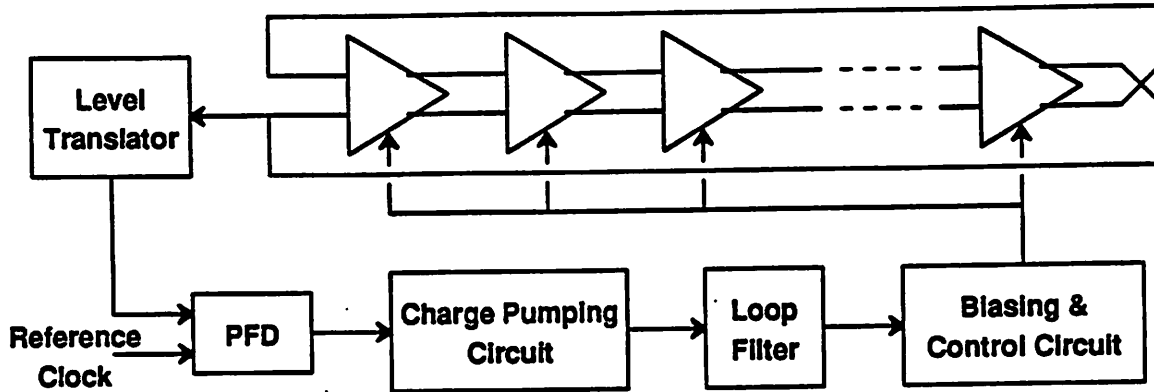


Figure 4.4 Analog Phase-Locked Loop

Here, the rising edge of the reference clock coming from crystal oscillator is compared by a phase-frequency detector with that of a local clock taken from one of the taps in the ring oscillator. Then the output of the phase-frequency detector controls the charge pumping circuits. Figure 4.5 shows the phase-frequency detector logic diagram and the associated charge pumping circuit.

Here, the two D flip-flops act as memory devices which can store the previous phase of each input clock in order to detect the frequency difference. Without a memory device, a phase detector can detect only the phase difference. This phase frequency detector was developed at National Semiconductor and is used in many places when frequency detection is necessary because of its simplicity.

If the rising edge of the reference clock lags behind that of the local clock, the pump-down signal goes high when the rising edge of the local clock starts and the pump-up signal stays low. Then the pump-down signal goes low when the rising edge of the reference clock begins. Due to the active pump-up signal, the charge stored in the capacitor of the loop filter is discharged through the lower switch during the high state of the pump-up signal. Then the voltage of the node connected to the bias circuits falls. The bias circuits detect an reduced voltage and adjusts the rising edge of

the local clock to the right (i.e. the direction of increased delay). If the rising edge of the reference clock leads that of the local clock, the reverse action happens. Note here that the pulse width of the clock does not affect the phase comparison. This is very important because the pulse width is hard to control in many cases.

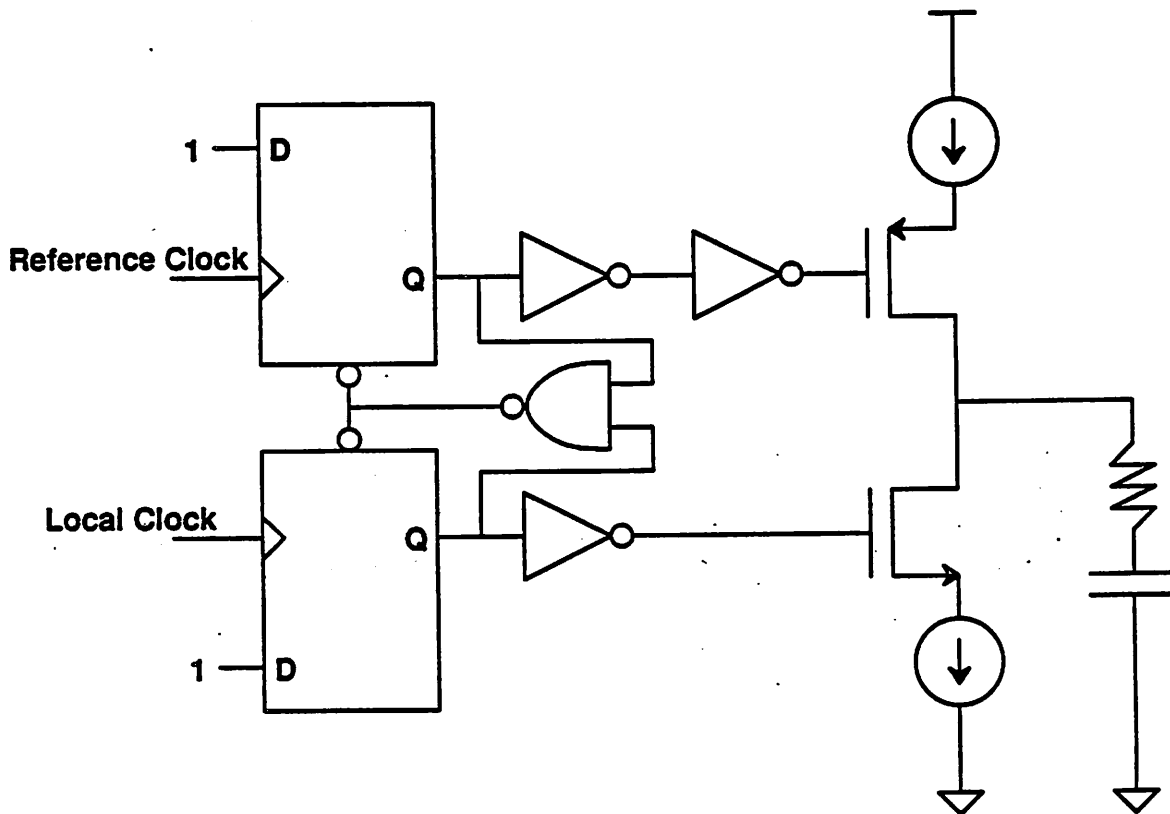


Figure 4.5 Phase Frequency Detector and Charge Pumping Circuit

If there is a frequency offset between the reference clock and the local clock, this offset can also be detected by the frequency-phase detector in the following way. Let f_R be the frequency of the reference clock, T_R be the period of the reference clock, and f_L be the frequency of the local clock. Assume $f_R > f_L$. Then $p(0)$, the probability of no local clock rising edge in $[t, t+T_R]$ is $1 - \frac{f_L}{f_R}$ and $p(1)$, the probability of one

local clock rising edge in $[t, t+T_R]$ is $\frac{f_L}{f_R}$. Then the average difference of the occurrences of "pump-up" and "pump-down" is $p(0) \times 1 + p(1) \times 0$ or $1 - \frac{f_L}{f_R}$ which is always greater than 0 as long as $\frac{f_L}{f_R} < 1$. This difference makes the frequency of the pumping-up higher than that of the pumping-down, therefore the output voltage of the charge pump circuit increases and the frequency of the ring oscillator also becomes higher. The opposite case occurs for $\frac{f_L}{f_R} > 1$.

Next, we consider the loop filter block. Here a simple RC single pole, single zero filter was used to reduce the input jitter. In this case, the zero compensates the phase shift from 180 degree to 90 degree when the loop transfer curve crosses the 0 dB line and makes the loop stable.

The next block is the biasing circuit. The ring oscillator is usually controlled by a current source, and so a voltage-to-current converter is necessary in general. But, in this case, since the ring oscillator is controlled by voltage reference, the voltage-to-current converter is not used. This circuit will be explained with the ring oscillator delay cell.

The last block is the voltage controlled oscillator (VCO). To generate multiple clocks spaced one gate delay, a ring oscillator was used as a voltage controlled oscillator in the analog phase-locked loop. But one of the main problems of a ring oscillator is jitter caused by the supply noise injection. Therefore, a differential delay cell is used instead of a current controlled inverter delay cell used in a starved ring oscillator. Figure 4.6 shows the differential delay cell.

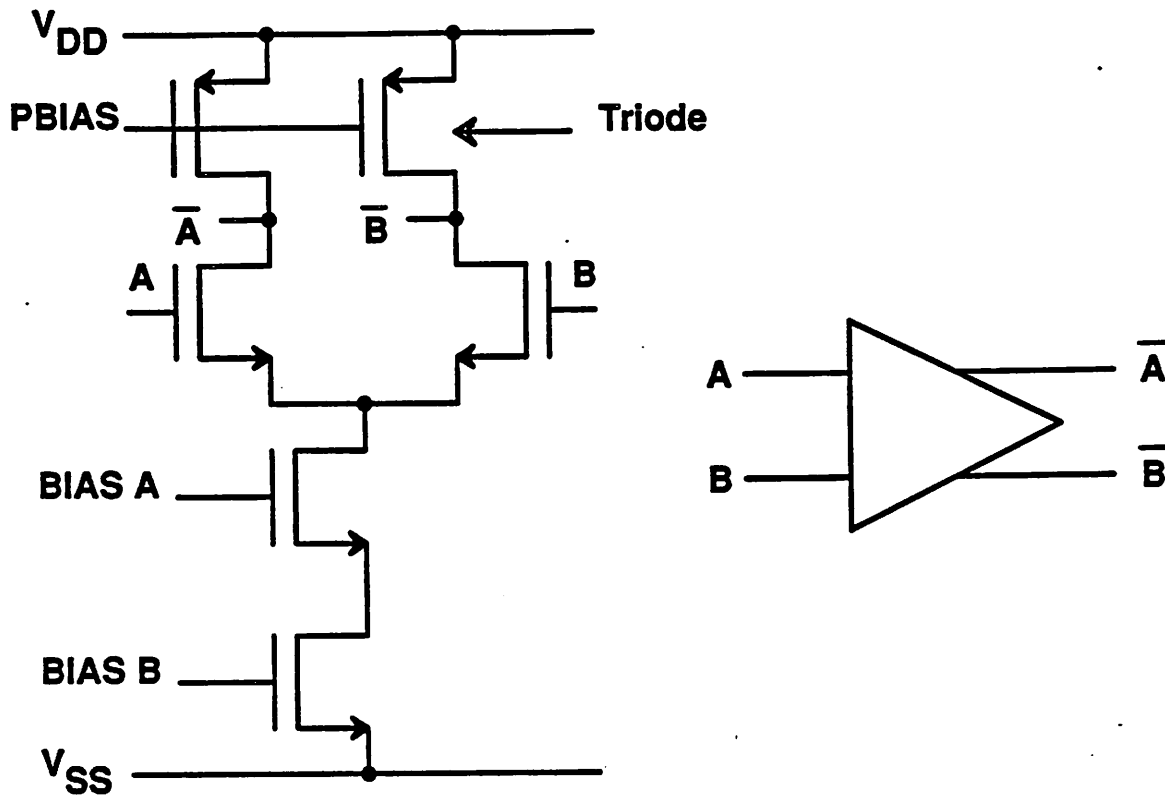


Figure 4.6 Differential Delay Cell

Here, the PMOS loads are in the triode region and act as voltage controlled resistors. The on-resistance is controlled by the PBIAS voltage level. If the PBIAS voltage goes up, the resistance will become larger and the voltage swing will grow since the voltage swing is on-resistance times the supplied current, and here, the supplied current is fixed. To increase the common mode rejection, a cascode current source is used. The PBIAS voltage is controlled by a source follower driven by a differential amplifier. Figure 4.7 shows the control circuit.

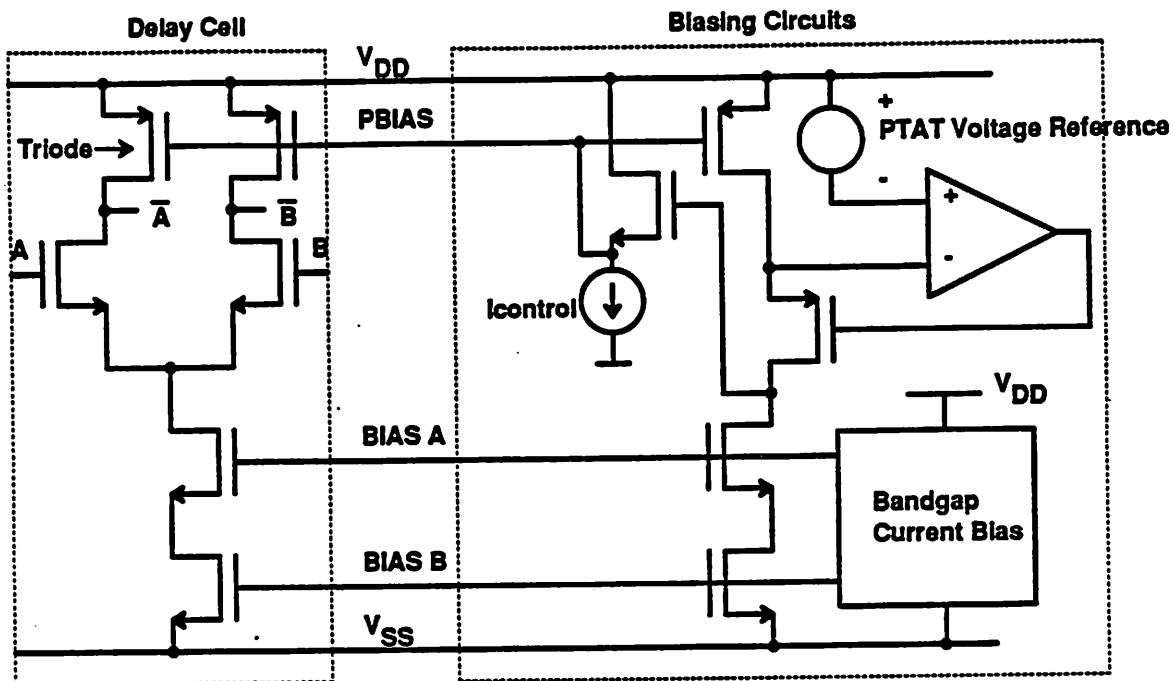


Figure 4.7 Biasing Circuits for Delay Cell

4.2.2 Parallel Phase Sampler

Since the sampling rate is about 1 G bits/sec if the tap-to-tap delay on the ring oscillator is 1 ns, it is not possible for the digital PLL's to process the sampled data in CMOS technology. Therefore, data rate translation is necessary. This is accomplished by *parallel phase sampler*. First, the incoming data is sampled by 32 data latches clocked using 32 taps from the ring oscillator. Since the latching happens sequentially through the 32 latches, the outputs of the latches are also available in the sequential manner. The next registers as shown in figure 4.8 consist of two sets of the latches. Each set contains 32 latches: the first 16 latches in the first set is clocked by ϕ_a , and the second 16 latches in the same set is clocked by ϕ_b , a delayed version (16 taps later) of ϕ_a . The first 16 latches in the second set is clocked by ϕ_c , and the second 16 latches in the same set is clocked by ϕ_d , a delayed version (16 taps later) of ϕ_c . Here, ϕ_a, ϕ_b, ϕ_c , and ϕ_d are coming from an four phase clock generator. Since there are 32

different delayed version of the clocks are available from the ring oscillator, clock generation is not problem.

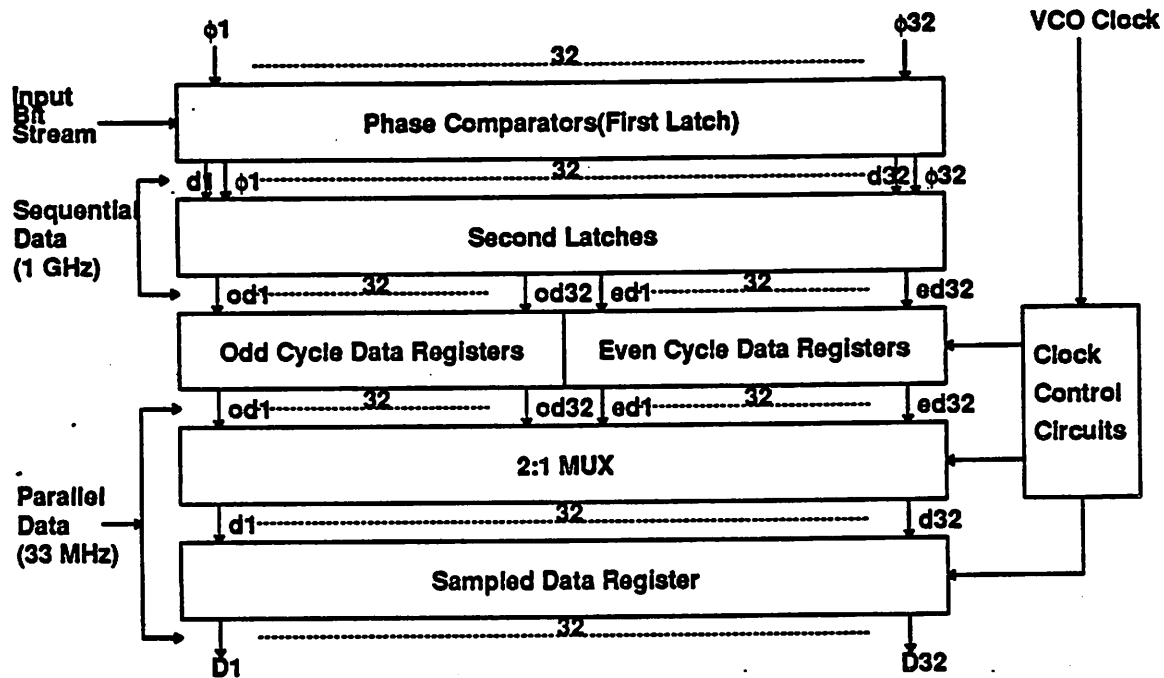


Figure 4.8 Parallel Phase Sampler Block Diagram

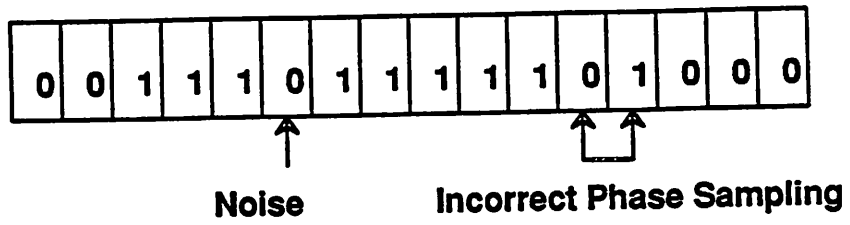
Next a multiplexer takes the 32-bit data from the two registers alternatively. For the odd cycle, the multiplexer takes the left register outputs (32 bits), for the even cycle, the multiplexer takes the right register outputs (32 bits). Then the outputs of the multiplexer are latched in the following latches. Now the data rate is 1/32 of the sampling rate. If the sampling rate is 1 G bits/sec, the data rate of the multiplexer outputs is about 30 M Words/sec. Now, the digital PLL can process the data for acquisition and tracking at this data rate.

4.2.3 Transition Detector and Digital Phase-Locked Loop

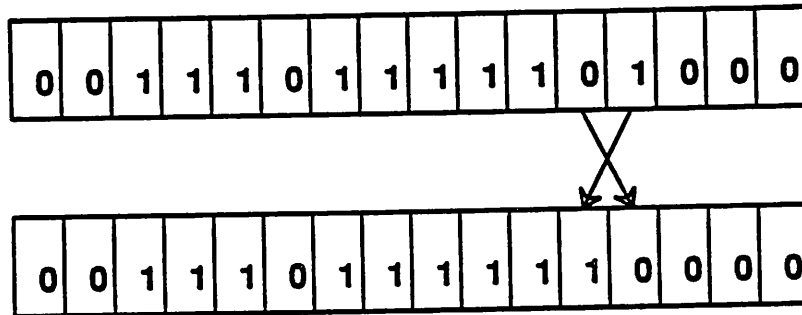
As shown figure 4.3, the outputs of the parallel phase sampler represents a snapshot of the incoming data bit. When a bit appears, a data transition edge occurs. If the incoming data is RZ, a low-to-high transition associated with each "1" occurs. Hence, the low-to-high transition contains all information associated with the data and timing of the data. If the shape of the incoming data bit is square and the sampling process is perfect, the edge detection can be done by simple exclusive-or gate. However, in a real system environment, there are several error sources which result in the data pattern not having one transition. Insteads, it may have several spurious bit flipping and bit loss due to sampling offset and noise.

Figure 4.9 shows several possible errors. The first one shows false sampling in the vicinity of the data transition because of sampling offset and noise. The second shows bit inversion due to burst noise. In the first case, the 1's and 0's should be swapped to obtain the correct data transition. In the second case, the "0" should be replaced by "1", or "1" should be replaced by "0". This kind of correction can take several clock cycles when it is processed through a serial smoothing filter which handles the correction in bit-by-bit manner. In the digital PLL application, this 32 bit correction should be done in one cycle. Therefore, a parallel smoothing filter is required. Figure 4.10 shows the real time digital signal processing algorithm for this smoothing filter.

Sampled Data



i) Reversing (To Remove Transition Noise)



ii) Filling (To Remove Isolated Noise)

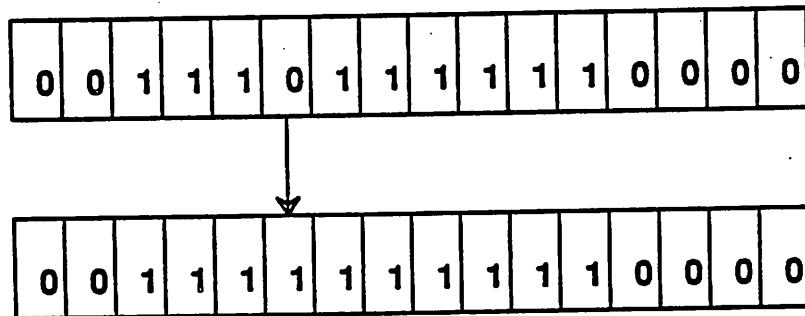
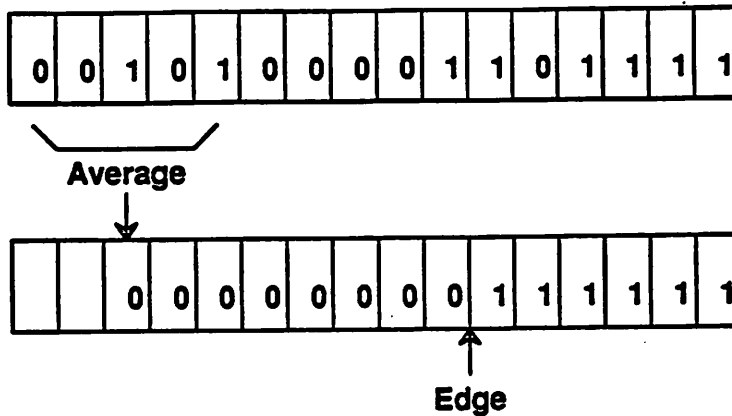


Figure 4.9 Possible Sampling Errors and Corrections



• **Single Stage 5-Bit Averaging Smoothing Filter**

Remove 2 Bit Wide Isolated Noise

Correct 4 Bit Wide Transition Noise

00011000 -> 0000

0001010111 -> 00000111

• **Double Stage 5-Bit Averaging Smoothing Filter**

Remove 2 Bit Wide Isolate Noise

Correct 8 Bit Wide Transition Noise

0001100110011111 -> 000110011111 -> 00011111

Figure 4.10 Smoothing Algorithm

Having been processed through the smoothing filter, the location of the low-to-high transition is detected by a simple logic gate. Then the location is encoded into a binary number. The encoded number is used to estimate the real phase position of the incoming data through the digital PLL. Figure 4.11 shows the digital PLL. The center of the current decode window (actually, the averaged location of the data transition

edges) is held in a current window phase register, and this register is updated by the digitally lowpass filtered phase error signal in the following manner.

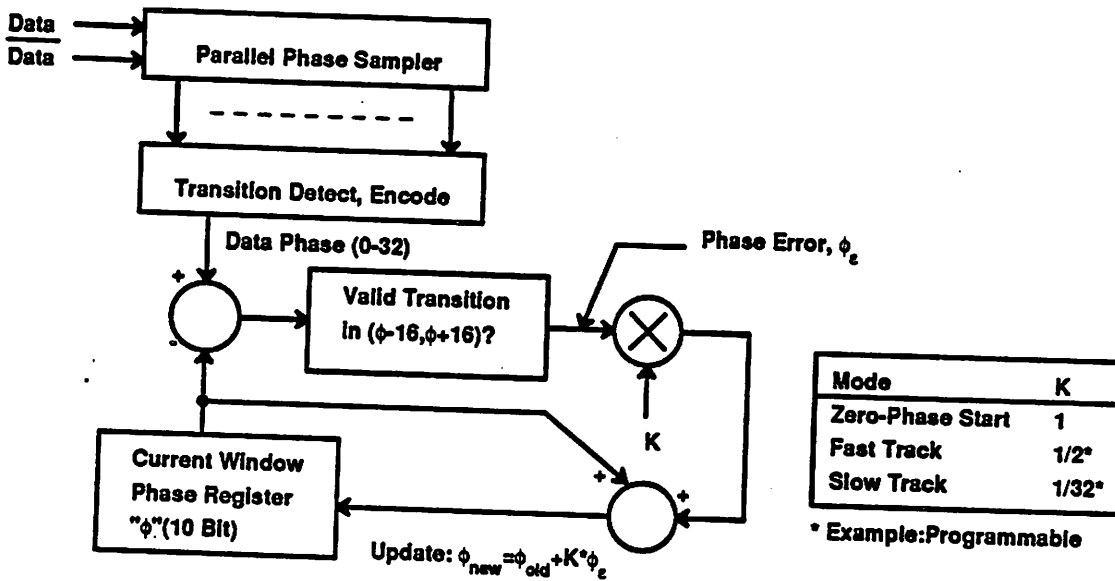


Figure 4.11 Digital Phase-Locked Loop Implementation

For each cycle, a new location for the transition is compared with this value if a transition occurred. If no transition occurred, no comparison happens. A subtractor is used to generate the difference between the new location and the center of the current decode window. If the new location leads the window center, a positive number is generated. This positive number is then multiplied by a constant K and the result is added to the window center. This new value updates the contents of the current window phase register. The multiply constant K acts like a loop time constant in an analog phase-locked loop. If K is a small number, the loop bandwidth gets narrower, the input jitter is largely attenuated. If K is 1, the new location is identical to the contents of the accumulator. This corresponds to the case of infinite loop bandwidth. Since the loop is composed of digital logic, the number K can be programmed easily by a sequence generator, and so "gearshifting" algorithm is easily implemented.

In a typical phase acquisition sequence, initially K is set to 1, and the new location for the first data bit fills the accumulator instantaneously. Then the K is set to an intermediate number such as $1/2$, and fast tracking and moderate jitter attenuation is introduced for the following several bit cycles. Finally, after fast tracking mode, during the data decoding, K is set to a very small number, and large jitter attenuation is allowed. By using this programmable K sequence, the acquisition takes only several bit cycle. Comparing this with traditional analog clock recovery circuits which usually take 20-40 bit cycle for the acquisition, this is one of the principal advantage of the digital PLL approach.

Once this window center is obtained, the decoding window is defined as a region which covers ± 16 taps from the center of the decoding window. If a new transition occurs in the decoding window, output "1" is generated from the digital PLL. If no transition occurred, output "0" is generated. When the window center is not at the center of the 32-bit data latches, the window over which valid transitions are accepted as valid within that particular time slot can extended into the previously stored 32 samples or the next, necessitating a simultaneous pipeline storage of transition locations from three sets of samples in order to evaluate symbol values.

Zero phase start, variable loop time constant, and other modes are implemented digitally. Other specialized disk drive functions, such as variable window width and offset, pulse pair compensation, and so forth can also be implemented directly in the digital domain. Also the loop dynamics are determined digitally avoiding critical design of the analog PLL normally used to lock and track the incoming data in clock recovery circuits. This approach can achieve virtually full window-width decoding since the window-width depends only on phase resolution. Here the phase resolution is about 1 ns in a 2 μm CMOS process. In local area network applications such as Ethernet, the more sophisticated symbol decoding allowed by this approach allows the jitter margin to be extended from the ± 18 ns, typical of current Ethernet serial

interface chips sampling at the 1/4 point, to a value that approaches the window half-width (50 ns).

The loop as described here is the first order, but in the event that frequency differences exist between the reference frequency and the data frequency, it is straightforward to add a frequency offset register to form a second order loop with zero static phase error in the presence of frequency offsets. In summary, the following algorithm explains the mechanism of the digital phase-locked loop for the first order case.

Step I:

- Current Input Phase if a Transition Exist: $\phi_i(nT)$
- Feedback Clock Phase: $\phi_o(nT)$
- Phase Difference: $\phi_e(nT) = \phi_i(nT) - \phi_o(nT)$
if $\phi_i(nT)$ exists, otherwise $\phi_e(nT) = 0$.

Step II:

- Current Feedback Clock Phase: $\phi_o(nT)$
- $\phi_o((n+1)T) = \phi_o(nT) + K \phi_e(nT)$

Where

- Zero Phase Start: $K=1$
- Fast Mode: $K=1/4$
- Slow Mode: $K=1/32$

4.2.4 Characteristics of Hybrid Analog/Digital Clock Recovery

The proposed hybrid analog/digital clock recovery architecture has several important characteristics. In this section, those characteristics are summarized as follows.

1. Currently, most high speed clock recovery circuits for data communications and magnetic applications make use of an analog PLL implemented in bipolar technologies. However these techniques are limited in their ability to implement sophisticated algorithms at high speed and are not well-suited for implementation in CMOS technology, which is necessary to achieve higher levels of integration and lower power dissipation. The hybrid architecture allows the use CMOS technology and incorporates the capability to implement sophisticated algorithms such as zero-phase start, gearshifting, programmable window width/offset, and programmable pulse pair compensation.
2. Conventional approaches for clock recovery usually use an analog PLL directly locked onto the incoming data bit stream. Therefore, the static and dynamic behavior of the PLL loop is directly related to input pulse jitter. In practice, the design of an analog PLL which meets both dynamic and static requirements is very difficult. Since the hybrid clock recovery circuit moves most loop behavior, such as locking and tracking, into the digital domain, the design of the analog PLL loop can be straightforward.
3. In hybrid clock recovery circuits, maximum effective decode window width is virtually full since the decode window width depends only on sampling noise. However, since the window width of the conventional clock recovery systems depends on circuit offsets, impairments of the loop components, and the loop behavior for the input and clock jitter, it is very difficult to make the effective decode window wide.
4. In order to lock onto the data, a conventional PLL requires certain amount of time as a training period, called the *preamble period*. Since the preamble period delays the data transmission time, it is desirable to

reduce this time period. Since the hybrid clock recovery circuits knows the phase of the incoming data accurately at the beginning, it can reduce the preamble time to 1-2 bit cycles. This is a significant reduction when compared to the preamble period required for conventional clock recovery circuits.

4.3 Circuit Design Issues

Realization of a low jitter, high performance clock recovery circuit using hybrid analog/digital architecture requires careful attention to certain aspects of the circuit design. In this section, several circuit design issues are addressed. The first issue is the phase noise accumulation. Each delay cell creates phase jitter at the output. Since the ring oscillator consists of several stages of delay cells, and the phase jitter is additive, the total jitter is very large because of the jitter accumulation through the ring oscillator. Chapter 5 describes this issue in detail. Next the jitter caused by power supply noise is considered. Since the simple inverter delay cell does not have good power supply rejection performance, the power supply noise creates large amount of jitter in the ring oscillator and degrades the performance of the hybrid analog/digital clock recovery circuit. This issues is addressed in the section 4.3.1. Also, Temperature and power supply variation creates drift of the free running frequency of the ring oscillator. If this drift is too large, the PLL can lose locking because the PLL has a finite range of locking. This problem is solved by using a replica bias scheme and illustrated in the section 4.3.2. The next issue is about the harmonic oscillation which can occur in the ring oscillator composed of long chain of inverters. A solution for this issue is suggested in the section 4.3.3. The following section 4.3.4 deals with the issue of the metastability involved in the data latches. For high speed clock recovery circuits, the digital data path can be a bottle neck because it involves several complex digital signal processing. A solution of this problem is to use pipelining scheme for

last stage outputs are crossed and connected to the first stage inputs. Finally, the differential delay cell has two outputs, therefore each cell generates two taps of the ring oscillator. If a 16 stage differential ring oscillator is used, a total of 32 taps are available. i.e. each delay cell gives an output and its inverse. While, in the starved ring oscillator case, if 32 taps are required, a 32 stage ring oscillator should be used. (Actually 33 stage should be used to make the ring oscillator oscillate.)

There are two problems associated with the differential delay cell. The first problem is longer gate delay. Since in CMOS technology, the square law MOSFET characteristic makes the delay inversely proportional to the voltage swing. Since the voltage swing is limited to 1 volt in the differential delay cell, the delay is longer than that of the inverter delay cell. However, since the differential delay cell has two outputs, tap-to-tap spacing (for example, the time distance between one rising edge and next following rising edge) is one gate delay. In the case of the starved ring oscillator, two stages are required to generate between two successive rising edges. Therefore, the time resolution of the inverter delay cell is two gate delay while the time resolution of the differential delay cell is one gate delay. Since the differential delay cell is about 30-50% slower than the inverter delay cell, this delay penalty is compensated.

The second problem is the difficulties in biasing. Since the PMOS loads should be in the triode region, we require a method to control both the cell bias current and the PMOS drain-to-source voltage in order to prevent the PMOS from going out from the triode region. A replica biasing scheme has been developed which can maintain the PMOS device in the triode region and which can also compensate the free running frequency variation due to temperature drift. The next section illustrates the biasing scheme.

4.3.2 Replica Biasing Circuit

As mentioned earlier, it is necessary that the free running frequency of the ring oscillator should be independent of temperature and power supply variation. In many cases, a PLL has a finite frequency locking range. If the free running frequency of a ring oscillator varies too much as temperature drifts, the PLL will lose the locking state. For power supply variation, the same argument applies. Figure 4.13 shows a replica biasing scheme which compensates for temperature and power supply variation.

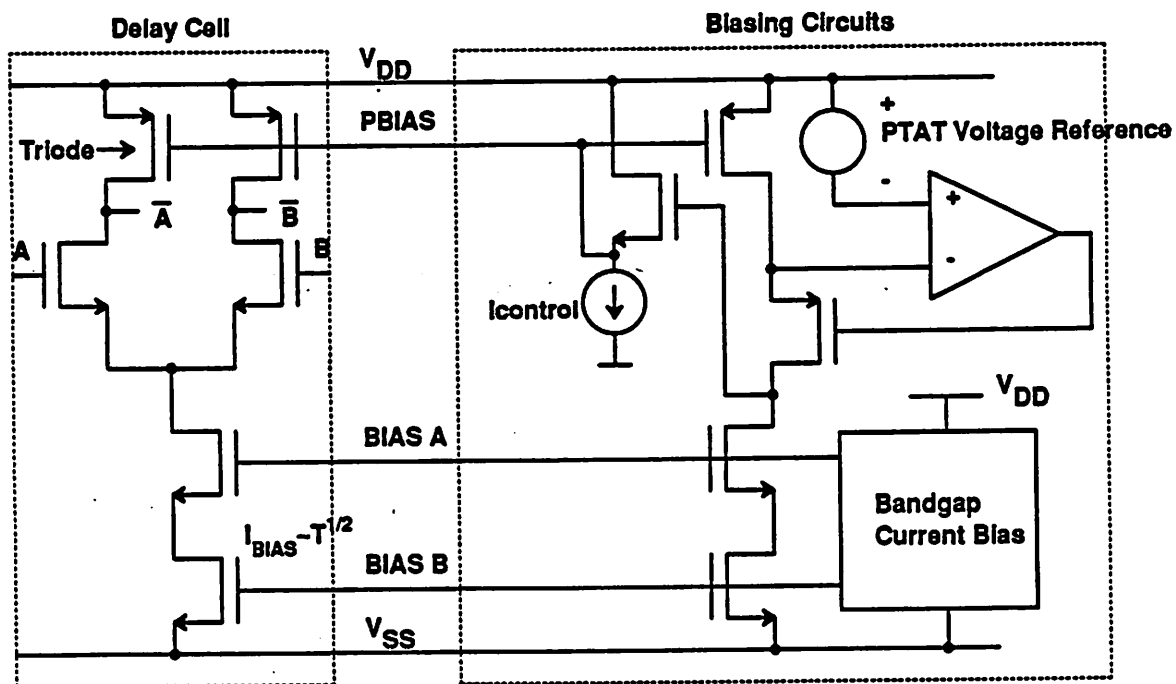


Figure 4.13 Replica Biasing Circuit

Here, a bandgap current bias circuit which is insensitive to power supply variation supplies the delay cell biasing current which is proportional to the square root of absolute temperature. Also a replica of an inverter cell is used to force the PBIAS voltage to a value which gives the voltage swing of the differential delay cell equal to

that of the PTAT (proportional to the absolute temperature) voltage reference. Since the delay of an inverter cell is proportional to the voltage swing and inversely proportional to the cell bias current, the net delay is proportional to the square root of absolute temperature. However, the cell bias current generated from the bandgap current bias circuit is usually proportional to the power of more than $1/2$, less than 1 of the absolute temperature, the net delay variation due to the temperature drift is very small.

Since the PTAT voltage reference circuit generates a reference voltage which is insensitive to power supply variation, and the bandgap current bias also generates the cell bias current insensitive to the power supply variation, the delay cell behavior is therefore insensitive to power supply variation. Figures 4.14 and 4.15 show the bandgap current bias circuit and the PTAT voltage reference respectively.

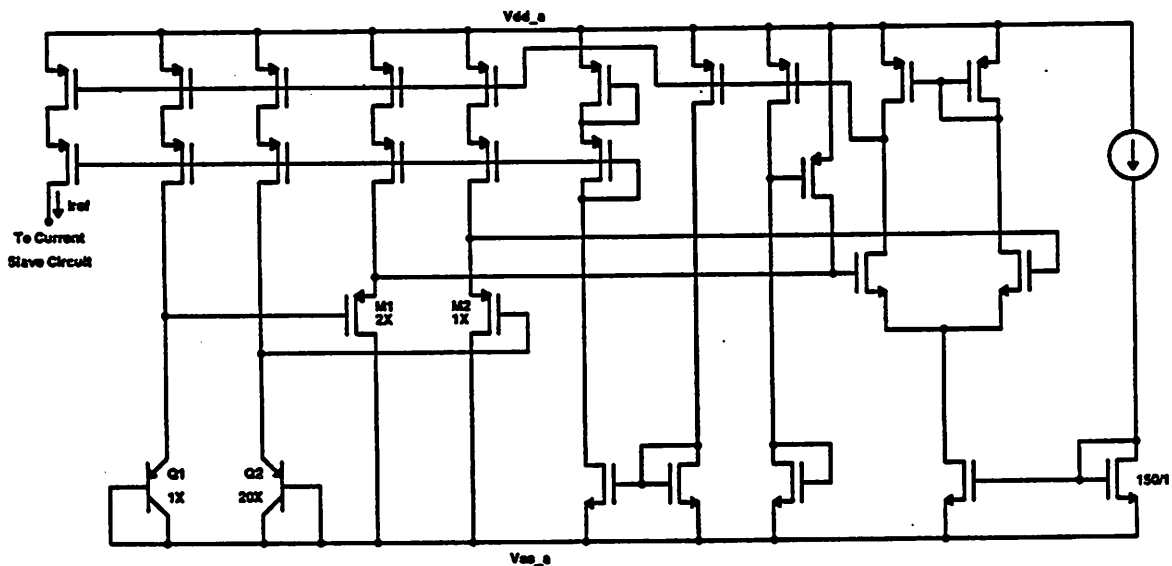


Figure 4.14 PTAT Voltage Reference Circuit

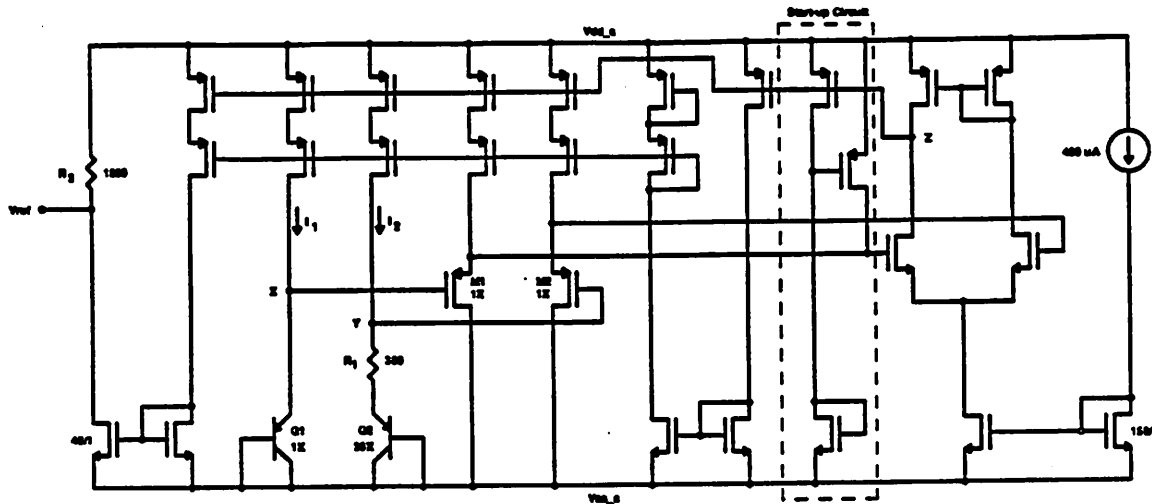


Figure 4.15 Bandgap Current Biasing Circuit

4.3.3 Harmonic Oscillation

In the ring oscillator composed of a long chain of inverters, a spurious harmonic oscillation can occur. Figure 4.16 shows the possible harmonic oscillation. In this picture, the second rising can propagate without any resistance and can persist indefinitely. Then two rising edges are traversing the ring oscillator and the ring oscillator generates the harmonics of the fundamental frequency. These harmonics create serious problems in the hybrid analog/digital clock recovery including general clock recovery since the harmonics deteriorate the normal circuit behavior.

In general, the second edges on the ring oscillator usually moves to the first rising edge and are merged to the first edge due to jitter and other circuit impairments. But since the probability of being merged and the time taken to be merged are not deterministic, a deterministic solution should be prepared to prevent spurious harmonic oscillation.

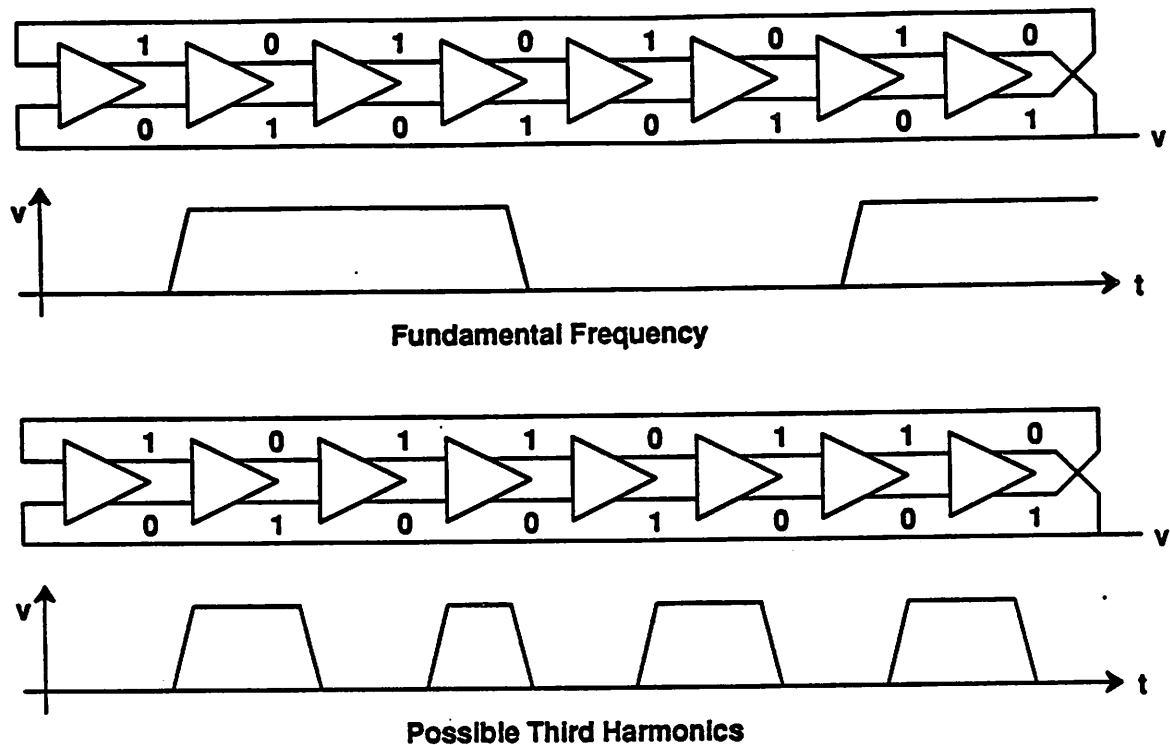


Figure 4.16 Spurious Harmonic Oscillation

A simple start-up circuit is usually used as a protection. As shown in figure 4.17, two PMOS gates are connected to each of the two outputs in the delay cell. When the circuit is idle, one gate is connected to ground and the other gate is connected to V_{dd} . Then, the two outputs of the delay cell are latched to "1" and "0". The following differential delay cell has the inversed outputs and the next etc... All the delay cells in the ring oscillator remain in one of the 2 possible latched states depending on their position. If the grounded gate is switched to the V_{dd} , the ring oscillator will become oscillate. Since there is only one rising edge on the ring, no harmonic oscillation is possible.

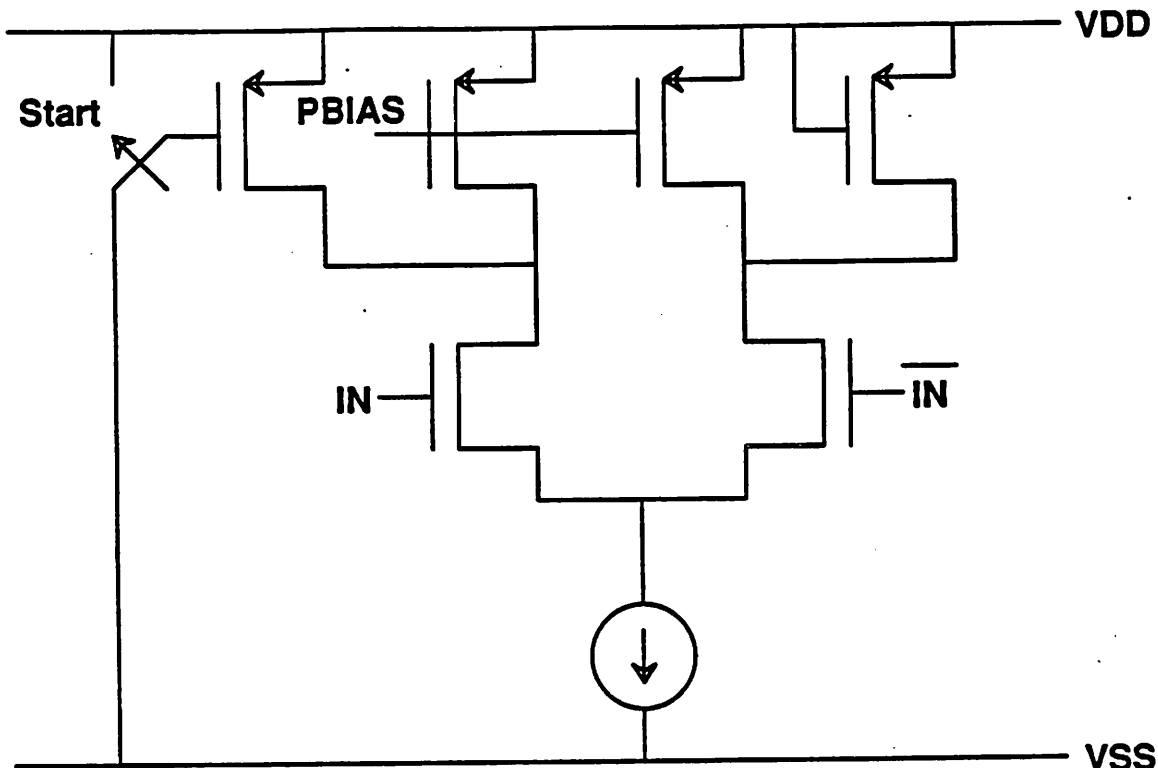


Figure 4.17 Start Up Circuit

4.3.4 Dual Latch Data Path

Since the hybrid analog/digital clock recovery relies heavily on the sampling of the incoming data, the problem of metastability involved in the data latches should be reduced to a tolerably low level. A dual latch data path is used to reduce the probability of metastability induced data errors. The probability reduction happens in the following two senses. First with respect to gain, the first latch has reasonably large gain, which amplifies the small voltage to a resolvable voltage for the next latch. Hence the metastability region is reduced by that gain factor. Second, in the sense of resolving time: the second latch is placed in the regenerative mode by a delayed version of the clock taken from an inverter several stages down the ring and allowed to have enough time to make its decision. Effectively, this is equivalent to a high gain single latch that is allowed to have more resolving time than it normally would. Figure 4.18 shows the

dual latch data path.

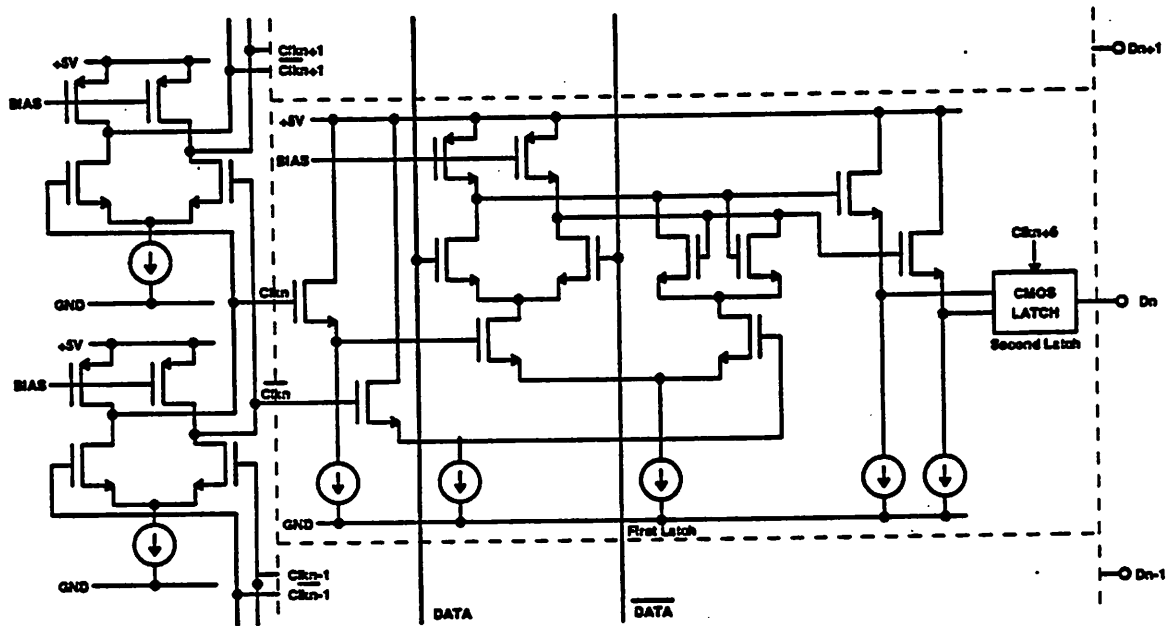


Figure 4.18 Dual Latch Data Path

4.3.5 Pipelining in the Digital Phase-Locked Loop

The digital phase-locked loop (DPLL) consists of several arithmetic units. Since the digital phase-locked loop update the location in one bit cycle and has a feedback path, the total delay taken to process the data through the feedback path should be less than one bit cycle. In a high speed clock recovery circuit, the total data path delay can be too long to allow complete processing within a single bit cycle. In this case, pipelining of the feedback should be considered. However, two important factors should be considered when a pipelining scheme is used. The first one is the stability of the pipelined digital PLL since the pipelining adds an additional pole to the loop transfer function. The second is the static phase offset of the digital PLL because the pipelining also gives one more zero to the loop transfer function.

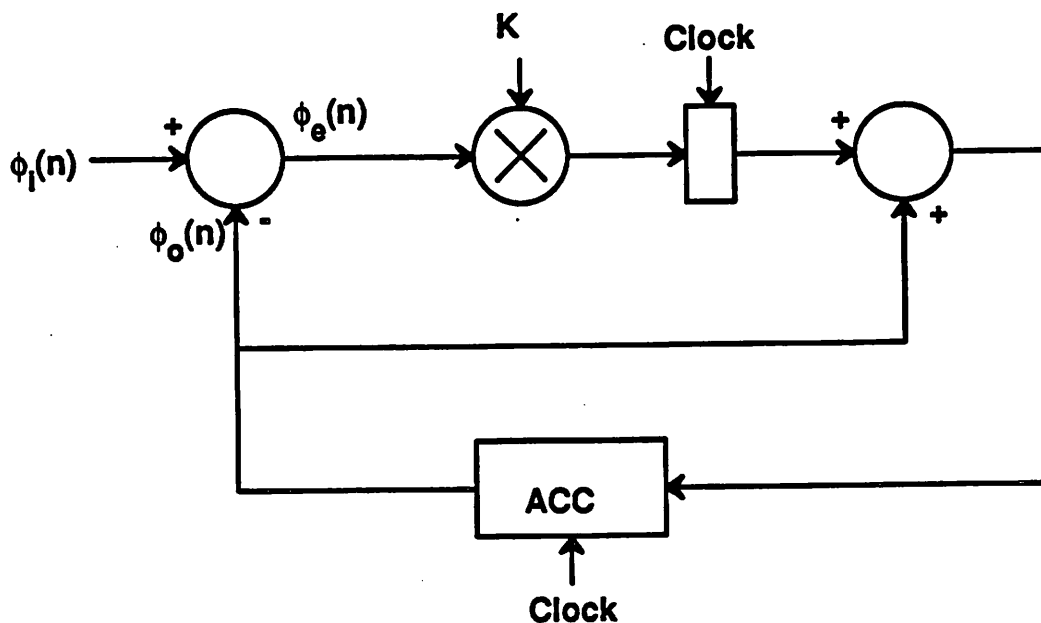


Figure 4.19 Digital Phase-Locked Loop Model

Figure 4.19 shows the simplified discrete time model of the digital phase-locked loop. Here, a pipeline buffer is used between the multiplier and adder. When there is no pipelining, the buffer is replaced by a direct connection. The loop transfer function in the z domain for the non-pipelined DPLL is then represented by (4.1)

$$\frac{\Phi_o(z)}{\Phi_i(z)} = \frac{K}{z - (1-K)} \quad (4.1)$$

Also the error transfer function is represented by (4.2).

$$\frac{\Phi_e(z)}{\Phi_i(z)} = \frac{(z-1)}{z - (1-K)} \quad (4.2)$$

From the equation (4.2), it is guaranteed that the loop has zero static phase offset in the following way. From the final theorem, the final value of the error will be expressed as (4.3) for the step phase change.

$$\theta_e(t \rightarrow \infty) = \lim_{z \rightarrow 1} \frac{\Phi_e(z)}{\Phi_i(z)} (z-1) \frac{z}{(z-1)} = 0 \quad (4.3)$$

Here, the $(z-1)$ term in the nominator of (4.2) goes zero as z goes 1.

The location of the pole in both equation determines the stability of the loop. Figure 4.20 shows the pole-zero diagram of the transfer function. From this figure, it can be seen that the location of the pole always remains inside of the unit circle as long as $0 < K < 1$. Therefore, the DPLL is always stable.

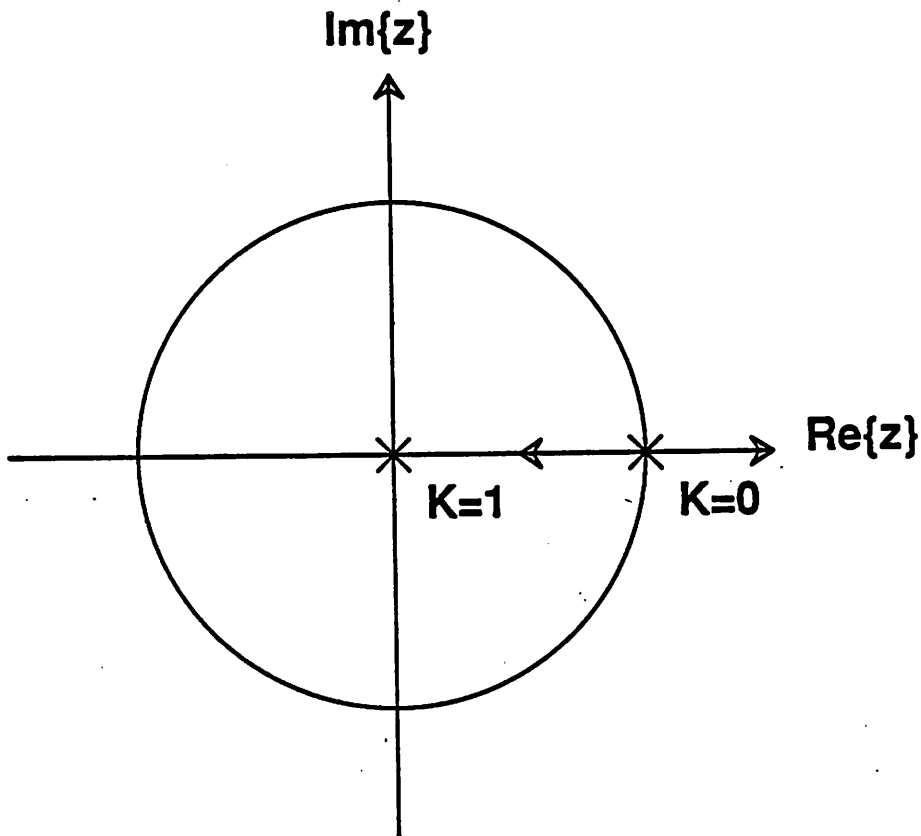


Figure 4.20 Pole-Zero Diagram for Non-pipelined DPLL

In the pipelined DPLL case, the pipeline buffer is active, and it delays the output of the multiplier by one bit cycle. The loop transfer function in this case is represented by (4.4),

$$\frac{\Phi_o(z)}{\Phi_i(z)} = \frac{K}{z^2 - z + K} \quad (4.4)$$

Copyright © 1990, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

and the error transfer function is represented by (4.5).

$$\frac{\Phi_e(z)}{\Phi_i(z)} = \frac{z(z-1)}{z^2 - z + K} \quad (4.5)$$

In (4.5), $(z-1)$ term appeared again in the nominator. Therefore the loop has zero static phase error again. To determine the stability of the loop, a pole-zero diagram for the transfer function is drawn in figure 4.21. Here, again the two poles remain inside of the unit circle as long as $0 < K < 1$, and the DPLL with pipelining is stable.

In the pipelined DPLL, either the initial K sequence or the preamble bit sequence in the acquisition should be carefully chosen. If the incoming bit stream for the preambling is 1,1,1... and RZ, the multiplier takes the phase of the bit transition immediately without any subtraction because the accumulator stores zero initially, and sends it to the pipeline buffer in the first bit cycle since K is one. In the second bit cycle, since K is set to $1/2$, and the accumulator still has zero, the new phase is directly sent to the multiplier and is multiplied by $1/2$. Then this value is added to the previously stored value in the pipelining buffer, and the sum replaces the contents of the accumulator. This causes overshooting of the phase value and slows down the acquisition speed. Therefore, in the next bit cycle, a smaller number for K such as $1/4$, or less should be used and this will reduce the SNR performance in the acquisition sequence. Another better way to solve this problem is to use 1,0,1,0 sequence in the preambling. In this case, for the even bit cycles, since no new phase information is fed to the loop, the accumulator stores the correct location of the first bit transition, and no overshooting occurs. This method can use the normal optimal K sequence that is calculated in chapter 6, therefore no SNR performance loss occurs.

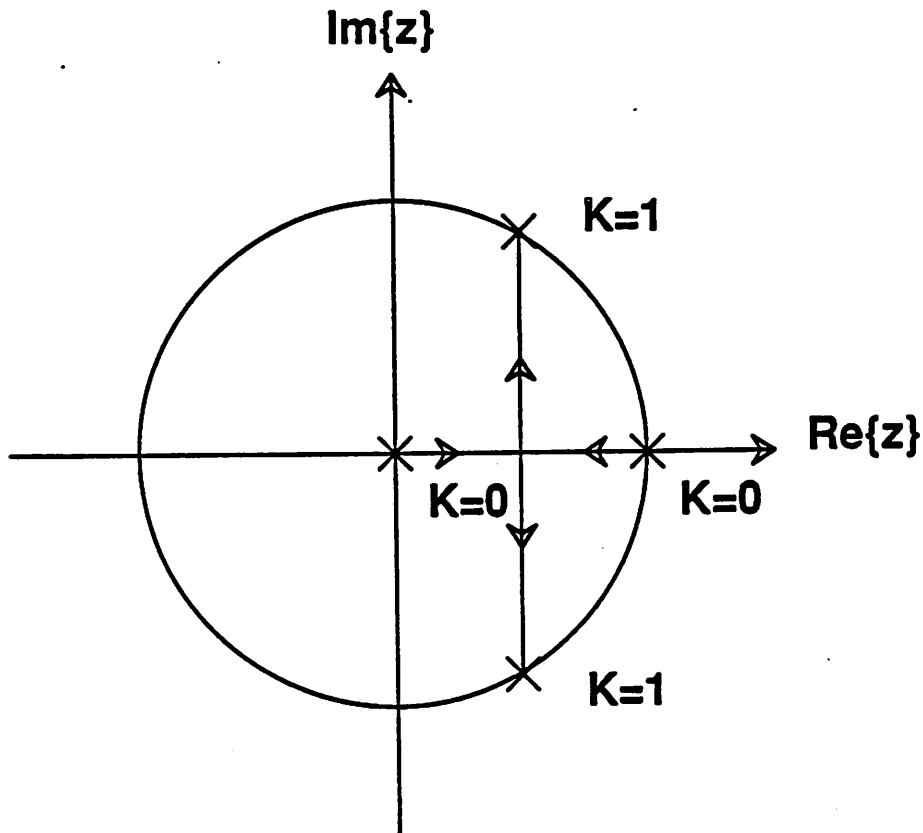


Figure 4.21 Pole-Zero Diagram for Pipelined DPLL

4.3.6 Phase Quantization Effects

The calculation in section 4.3.5 for the zero static phase error is based on the linear analysis without the input phase quantization effect. Since the hybrid clock recovery circuit samples the input at discrete time intervals spaced one gate delay apart, the input phase (the location of the transition) is obtained in the form of a binary number with a finite length, which results in the quantization errors in phase domain. This is analogous to quantization errors introduced in the voltage domain by the A/D converter as it samples the input at discrete voltage intervals. This phase quantization noise can be modeled as an additive noise in the input and represented by (4.6).

$$\sigma_N^2 = \frac{\Delta^2}{12} = \frac{2^{-2b}}{12} \quad (4.6)$$

where Δ is the time interval, and b is the number of digits representing the input phase. Since the quantization phase noise disturbs the loop settling, zero static phase error cannot be achieved. Instead, a small phase fluctuation centered at zero exists in $\phi_e(n)$. To find out the variation of the noise power in $\phi_e(n)$, a simple integration is evaluated in (4.7)

$$\sigma_e^2 = \frac{\sigma_N^2}{2\pi} \int_{-\pi}^{\pi} |H_e(e^{jw})|^2 dw \quad (4.7)$$

Hence,

$$|H_e(e^{jw})|^2 = \frac{2(1-\cos(w))}{1-2(1-K)\cos(w)+(1-K)^2} \quad (4.8)$$

Therefore, the result is represented by (4.9)

$$\sigma_e^2 = \frac{2\sigma_N^2}{2-K} \quad (4.9)$$

For small values of K , the phase error variation is almost the same as the input phase quantization noise power (σ_N^2).

4.4 Design of Hybrid Clock Recovery Circuit

As a conclusion to chapter 4, several important features of hybrid analog/digital clock recovery circuits are reviewed. To be concise, each feature is separated into one paragraph and explained in brief. In section 4.4.1 the circuit issues in designing hybrid clock recovery circuits are summarized briefly. Section 4.4.2 describes possible methods to achieve higher performance for the hybrid clock recovery circuit. Finally, projected performance in scaled technologies is presented in section 4.4.3.

4.4.1 Summary of Circuit Design Issues

- Phase noise accumulation in the PLL should be minimized because it contributes to the decode-window boundary uncertainty. Therefore, a large gate area for the inverters in ring oscillator is preferred.
- For the same reason, the jitter caused by noise injection from power supply must be minimized requiring a power supply insensitive voltage controlled oscillator. A differential delay cell for the ring oscillator is preferred.
- To prevent spurious harmonic oscillation which can occur in a ring oscillator composed of a long chain of inverters, a start-up circuit for the ring oscillator is necessary.
- In practical circuit design consideration, the PLL loop should have reasonably small temperature and supply sensitivities of the center frequency of the ring oscillator. This can be achieved by using replica biasing scheme composed of bandgap current bias and PTAT voltage reference.
- The speed of the digital PLL data path can be a bottle-neck because it involves complex digital signal processing. Therefore, pipelining and parallelism should be applied in the design of the digital PLL.
- The probability of metastability involved in data latches should be reduced. This can be achieved by using a dual-latch data path.

4.4.2 Possible Extensions for Increased Performance

- Many applications require a clock recovery circuit with frequency capturing capability. A second order loop for the digital PLL will give this capability.
- Power consumption can be reduced by using a delay-locked loop(DLL). Since the jitter in a delay locked loop is much lower than that of the PLL, hence small current can be used in the delay cell. Also, the loop design can be simpler since it requires only one capacitor as a loop filter.

- The area and speed of hybrid clock recovery circuits can be greatly improved by using scaled technology. For example, using $1\mu\text{m}$ channel length MOS devices, the speed can be improved by factor of 4 and the chip size can be reduced by factor of 2.
- The control range of the ring of this ring oscillator is limited to $\pm 15\%$. Since process parameter variation changes the free-running frequency by more than 30%, this should be improved. The current design adopted the frequency control scheme using the voltage controlled resistance of the PMOS load. A current controlled scheme gives more than 100% of the control range.
- More sophisticated algorithms such as pulse pair compensation, window shifting, and multiple frequency locking should be implemented in the digital domain to achieve higher performance.

4.4.3 Projected Performance in Scaled Technologies

- The prototype is implemented in $2\mu\text{m}$ CMOS technology. If the prototype is implemented in a scaled CMOS technology with constant voltage scaling, [HODGES] i.e. the horizontal dimensions and vertical dimensions are divided by a constant factor k , the oxide thickness is divided by the same factor, and substrate doping is multiplied by the factor of k^2 , while the voltage source and V_T is maintained to a constant, then the current flowing through an inverter cell increases by the scaling factor k (refer to table 8.1).
- Also, since the output capacitance decreases by the factor k , the delay per each inverter cell decreases by the factor of k^2 . Hence, the operating frequency of the hybrid clock recovery circuits is multiplied by the factor of k^2 .
- The time jitter decreases by the factor of $k^{3/2}$ since the current increases by the factor k , while the gate capacitance decreases by the factor k and from (5.30).

However, the ratio of the jitter to tap-to-tap delay increases resulting in degraded phase resolution because although jitter decreases by the factor of $k^{3/2}$, the tap delay decreases more rapidly by the factor of k^2 .

- The power consumption increases by the factor of k since the current increases by factor of k .
- Finally, the chip size decreases by factor of k^2 , since the channel length and channel width decreased by factor of k respectively.

Table 4.1

Projected Performance with Constant Voltage Scaling

Original	Scaled
W, L, t_{ox}	$(W, L, t_{ox})/k$
V_{DD}, V_T	V_{DD}, V_T
C_{ox}	$k * C_{ox}$
N_D, N_A	$k^2 * (N_D, N_A)$
C_{jo}	$k * C_{jo}$
I_{DD}	$k * I_{DD}$
Jitter	$Jitter/k^{3/2}$
Delay	$Delay/k^2$
Ratio(Jitter/Delay)	$k^{1/2} * Ratio(Jitter/Delay)$
Power	$k * Power$
Area	$Area/k^2$

Chapter 5

Jitter Analysis

5.1 Introduction

Jitter in ring oscillators creates the problem of *window boundary uncertainty* as shown chapter 2. Therefore, the jitter should be controlled to an acceptable value. There are three main jitter sources in ring oscillators. The first one is power supply noise. Since the frequency of an inverter chain based ring oscillator increases as the power supply increases, noise on the power supply directly creates jitter. The second problem is low frequency noise, i.e. $1/f$ noise. Since the low frequency noise is usually modulated by the oscillator free running frequency, it creates sidebands falling off from the center frequency and causes jitter. The third source of jitter is high frequency thermal noise. This noise creates phase jitter at the output of each delay cell. Since the phase jitter is additive, it is accumulated through the ring oscillator.

The jitter caused by the power supply can be rejected by using differential delay cell in the ring oscillator. Also, jitter caused by $1/f$ noise will be rejected by the phase-locked loop because the loop is a highpass filter for phase jitter in the ring oscillator. Therefore only high frequency jitter caused by the thermal noise remains as the main contributor of phase jitter in an analog PLL. Therefore, an analysis of high frequency jitter in the PLL is required in order to design a low phase jitter PLL which generates multiple clocks for the data latches.

Firstly, in section 5.2, jitter in ring oscillators is derived for three different types of ring oscillators. Section 5.3 presents an analysis of PLL jitter using the results of the section 5.2. In section 5.4, jitter of DLL's is analyzed. Section 5.5 gives an example of the analysis. Finally, in section 5.6, the phenomenon of PLL jitter accumulation

is simulated using C programming language.

5.2 Jitter in Ring Oscillators

Figure 5.1 shows a ring oscillator composed of a chain of N stage inverters. Here, if τ represents the delay per stage, and $\Delta\tau_i$ represents the jitter created from stage i , then t_p , the total period of the oscillation is represented by (5.1).

$$t_p = \sum_{i=1}^N (\tau + \Delta\tau_i) \quad (5.1)$$

Then, the mean and the variance of t_p are given by (5.2) and (5.3) respectively since $\Delta\tau_i$ is uncorrelated with $\Delta\tau_j$ if $i \neq j$.

$$E[t_p] = N\tau = T \quad (5.2)$$

$$E[t_p^2] = T^2 + N\Delta\tau^2 \quad (5.3)$$

where $E[\Delta\tau_i^2] = \Delta\tau^2$ since each inverter cell is identical. Therefore, to find the total jitter variance, first each jitter variance per stage should be calculated, then it is simply multiplied by N .

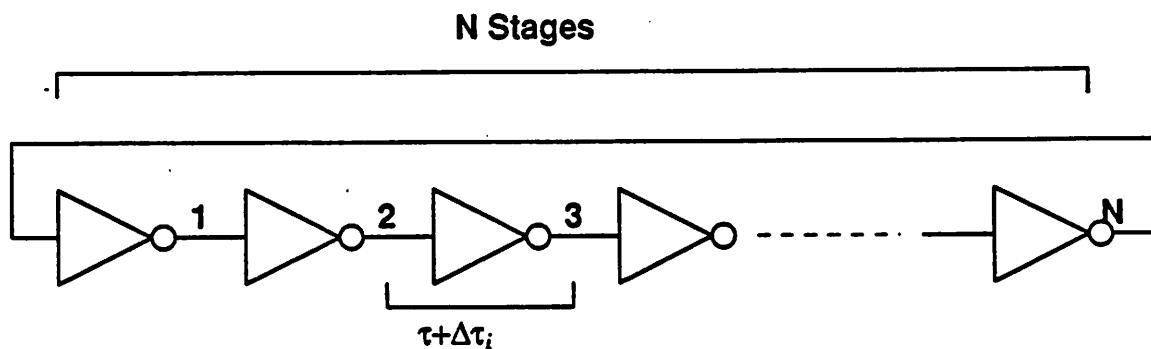


Figure 5.1 N Inverter Stage Ring Oscillator

As mentioned earlier, the main source of phase jitter in PLL's using a ring oscillator as a VCO is the thermal noise coming from each MOS device. Also, since the thermal noise generated from each MOS device is uncorrelated with each other, the total jitter variance of the ring oscillator is simply the sum of jitter variance from each MOS device. Therefore, to evaluate the phase jitter, first only one MOS device is assumed to have a thermal noise source, and the others are assumed to be noise free. Then, the jitter variance due to the thermal noise source is derived under the above assumptions. Next, another MOS device is assumed to have a thermal noise source, and the others including the previous noisy MOS device are assumed to be noise free. Finally, this procedure is applied to the last MOS device, and the total phase jitter variance is found from the sum of those jitter variances.

In general, ring oscillators can be categorized into three different types. The first one is a *fast-slewing saturated ring oscillator*, which uses delay cells having fast rise time and fall time, and performing full switching (i.e. if the delay cell is a CMOS inverter, then each PMOS and NMOS device completely turns off when switching.) Figure 5.2(a) shows the output voltage swings of three adjacent delay cells. Note here that each delay cell can consist of one inverter, one inverter plus one buffer [23], or several inverters plus several buffers. In this case, the noise process is not stationary since each MOS device turns off completely once in every cycle of the ring oscillator and no noise is generated for those periods. Also, whenever the MOS device turns on, the noise process can affect the delay *only up to the time instant when the output signal of the delay cell crosses the input threshold voltage of the next delay cell* since once the output of the delay cell crosses the threshold, the input buffer of the next delay cell switches state rapidly.

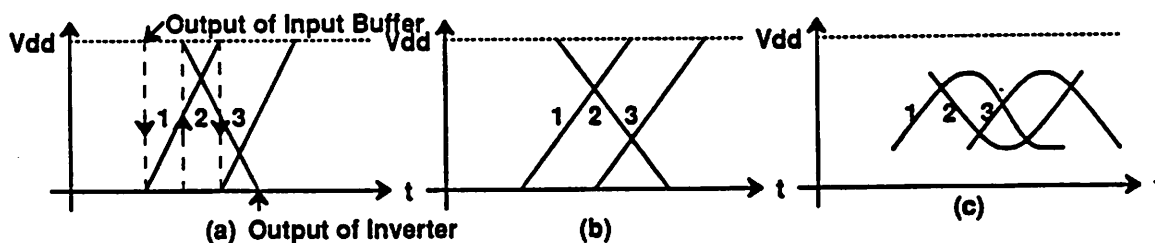


Figure 5.2 Output Voltage Swings for Three Different Types of Ring Oscillator

The second type is a *slow-slewing saturated ring oscillator*. Figure 5.2(b) shows the output swings. In this case also, full switching occurs and the noise process is still non-stationary, but the noise process affects the delay even after the output of the delay cell crosses the input threshold voltage of the next delay cell. This kind of delay cell can be found in [12].

The third type is a *non-saturated ring oscillator* [8]. In this ring oscillator, the noise process can be approximated as a stationary process since all MOS devices are on all the time. Here, each delay cell can be modeled as an linear amplifier in this case since complete switching does not occur.

In the next three sections, jitter analysis is performed for each of the above types and a comparison of the three types is given in the following section. From this analysis, it can be concluded that the fast-slewing saturated ring oscillator shows the best jitter performance, the non-saturated ring oscillator shows the worst jitter performance, and the slow-slewing saturated ring oscillator shows intermediate jitter performance.

5.2.1 Fast-Slewing Saturated Ring Oscillator

To perform the jitter analysis for the fast-slewing saturated ring oscillator, a delay cell is assumed to consist of one input buffer (Schmitt Trigger) and one inverter

cell as shown in figure 5.3 [23]. Also, initially the inverter cell in the i th stage is modeled as an inverter composed of an NMOS device with thermal noise current source and a PMOS device without any noise source. (Later, the noise source of the PMOS will be included.) The delay of stage i is then defined as the time difference between the instant the *previous* Schmitt trigger output switches from low to high (i.e. when the NMOS turns on and the PMOS turns off) and the instant the inverter output reaches the threshold voltage of the *next* Schmitt trigger. As shown in figure 5.3, the delay is determined by two components, the slewing current and the noise current. The slewing component determines the nominal delay of the delay cell and the noise component determines its jitter. Hence, to find the jitter variance, the behavior of the noise voltage which started at time t_a due to the noise current should be estimated.

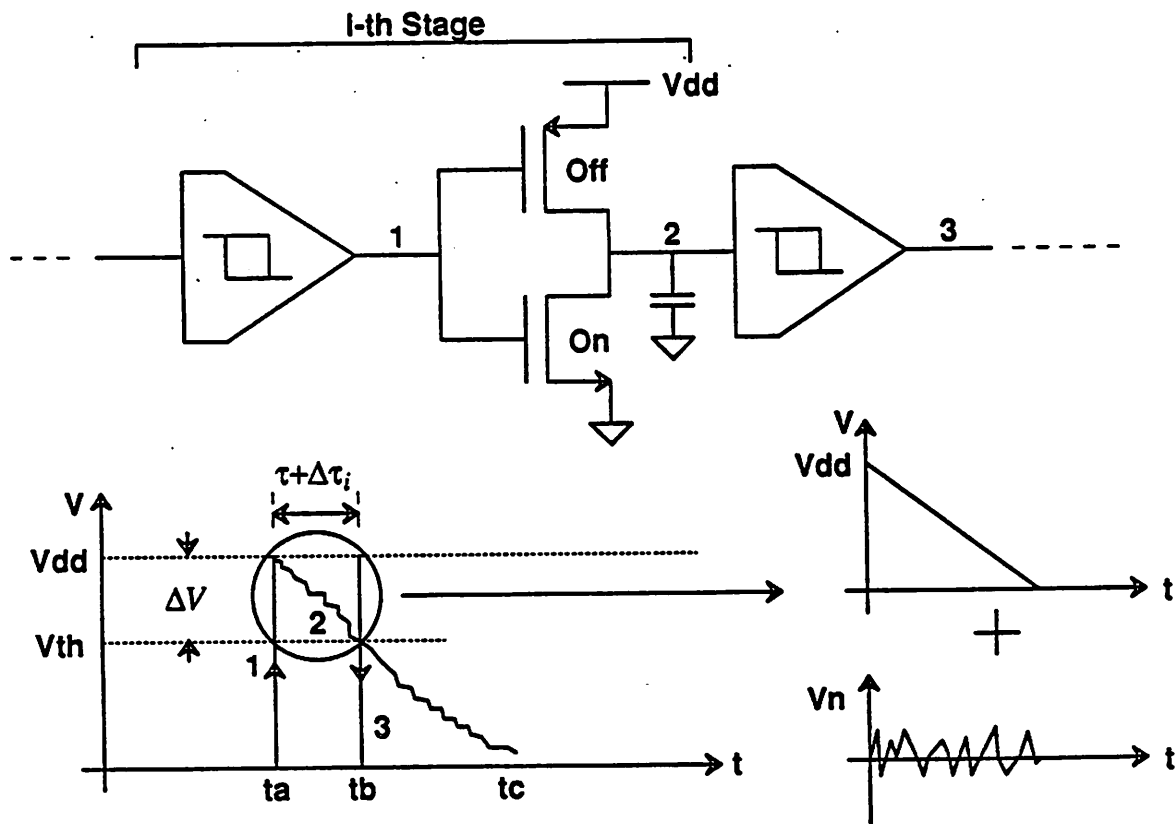


Figure 5.3 Delay Defined by Slewing Current and Noise Current

In figure 5.3, the NMOS device is in the saturation region and carrying a constant current. On the other hand, the PMOS is off, no current is flowing through it. In this case, the equivalent circuit for the delay stage can be modeled as shown in figure 5.4. Here, g_{ds} is the NMOS saturation region output conductance and C_g is the gate capacitance of the inverter in the next stage. For simplicity, the NMOS device is assumed to show square law characteristics.

In MOS technology, the thermal noise current spectral density of a MOS device in saturation is represented by (5.4).

$$\frac{\overline{i_n^2}}{\Delta f} = 4kT \frac{2}{3} g_m \quad (5.4)$$

where k is Boltzman Constant and g_m is the transconductance of the MOS device. Therefore, the autocorrelation function of the noise current when $i_n(t)$ is a stationary process is given by (5.5).

$$R_{ii}(\tau) = 4kT \frac{2}{3} g_m \delta(\tau) \quad (5.5)$$

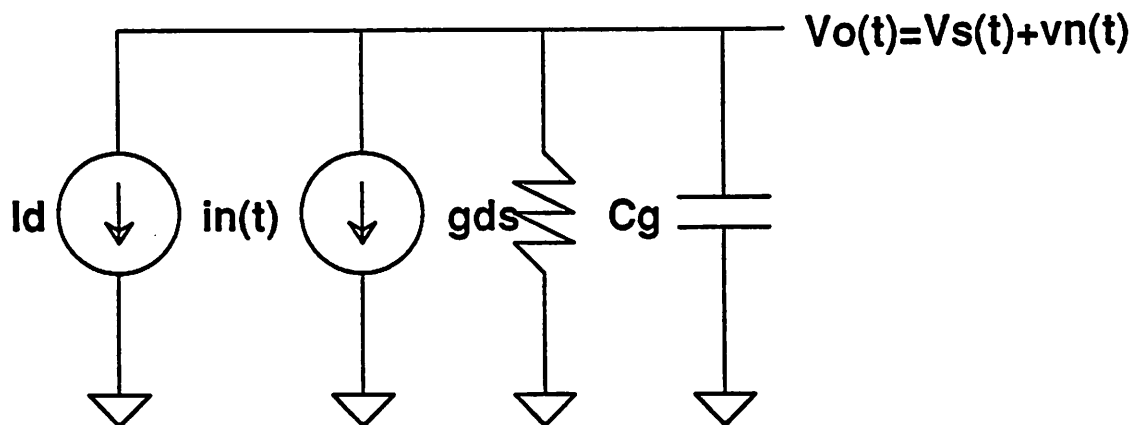


Figure 5.4 Equivalent Noise Circuit for Delay Stage

However, this process is clearly nonstationary since the noise current process, $i_n(t)$, is applied to a linear system, generates noise voltage, $v_n(t)$, at the output only between t_a and t_c ; that is, the input to the system is given by (5.6).

$$x(t) = \begin{cases} i_n(t) & t_1 \leq t \leq t_3 \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

In this case, the autocorrelation function is represented by (5.7).

$$R_{xx}(t_1, t_2) = 4kT \frac{2}{3} g_m \delta(t_1 - t_2) \quad t_a \leq t_1, t_2 \leq t_3 \quad (5.7)$$

Hence, finding the delay is the problem of zero crossing with a nonstationary process and is very difficult to solve. But, if the noise voltage is small, the time jitter caused by this noise voltage is modeled as shown figure 5.5 [24].

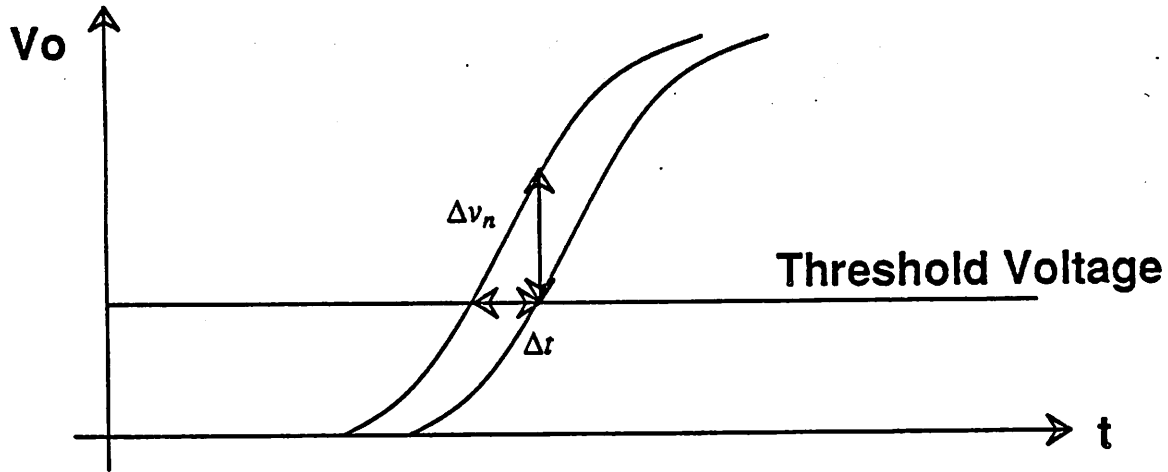


Figure 5.5 Jitter vs Noise Voltage

Here, the amount of time jitter is proportional to the amount of noise voltage present with the inverse of the slope being the proportionality constant and hence giving (5.8).

$$\Delta t = \left(\frac{dv}{dt} \right)^{-1} \Delta v_n \quad (5.8)$$

Therefore, the jitter is found from the noise voltage at time $t_a + \tau$, where τ is a true delay when there is no noise in the NMOS device. Since the noise voltage is a statistical quantity, the noise voltage variance should be calculated. To find the noise variance, the autocorrelation of the noise voltage should be found first using (5.9).

$$R_{vv}(t_1, t_2) = R_{ii}(t_1, t_2) * h(t_1) * h(t_2) \quad (5.9)$$

where $h(t)$ is the impulse response function of the linear system and represented by (5.10).

$$h(t) = \frac{1}{C_g} e^{-\frac{g_{ds}}{C_g} t} u(t) \quad (5.10)$$

where $u(t)$ is a unit step function. Then, from (5.9), (5.11) is given.

$$R_{vv}(t_1, t_2) = 2 \frac{kT}{C_g} \frac{2}{3} \frac{g_m}{g_{ds}} e^{-\frac{g_{ds}}{C_g}(t_1 - t_2)} (1 - e^{-\frac{2g_{ds}}{C_g}(t_2 - t_a)}) \quad t_a \leq t_1 \leq t_2 \leq t_c \quad (5.11)$$

Since $E[v_n(t_a + \tau)v_n(t_a + \tau)] = R(t_a + \tau, t_a + \tau)$, therefore, noise voltage variance is given by (5.12).

$$v_n^2 = \frac{4kTg_m}{3C_g g_{ds}} (1 - e^{-2\frac{g_{ds}}{C_g}\tau}) \quad (5.12)$$

Usually, $\tau \ll \frac{C_g}{g_{ds}}$, and τ is given by $\frac{C_g \Delta V}{I_d}$. Also, $\Delta V = V_{dd} - V_{th}$ as shown in figure

5.3. Then, (5.12) becomes (5.13).

$$v_n^2 = \frac{8kTg_m \Delta V}{3C_g I_d} \quad (5.13)$$

To find the jitter variance, the proportionality constant, $(\frac{dv}{dt})^{-1}$ is found from (5.14)

since the slope is mainly determined by slew rate.

$$\frac{dv}{dt} = \frac{I_d}{C_g} \quad (5.14)$$

Then, the jitter variance is given by (5.15).

$$\overline{\Delta t^2} = \frac{8kTC_g g_m \Delta V}{3I_d^3}$$

If the NMOS parameters are as follows,

$$g_m = \frac{W}{L} \mu C_{ox} (V_{gs} - V_T)$$

$$I_d = \frac{1}{2} \left(\frac{W}{L} \right) \mu C_{ox} (V_{gs} - V_T)^2$$

Then, (5.15) can be further approximated into (5.18).

$$\overline{\Delta t^2} = \tau_T^2 \left(\frac{64}{3} \right) \left(\frac{kT}{C_g V_{dsat}^2} \right) \left(\frac{\Delta V}{V_{dsat}} \right)$$

where, $\tau_T = \frac{L^2}{\mu V_{dsat}}$ is the intrinsic NMOS device transit time

ω_t of the device. From (5.18), the phase jitter is proportional

time, and proportional to $\sqrt{\frac{kT}{C_g}}$.

Since the noise signals generated by each MOS device are independent of each other, therefore the jitter variances are additive. If there are N inverter delay cells with each delay cell consisting of one PMOS and one NMOS device, then the total jitter variance is given by (5.19).

$$\overline{\Delta t_N^2} = \frac{8NkTC_g}{3} \left(\frac{g_{mN} \Delta V_P}{I_{dN}^3} + \frac{g_{mP} \Delta V_N}{I_{dP}^3} \right)$$

where ΔV_P and ΔV_N are two threshold voltages of the PMOS and NMOS devices respectively.

5.2.2 Slow-Slewing Saturated Ring Oscillator

The most popular ring oscillator of this type is the differential ring oscillator consisting of two differential delay cells as shown in figure 5.6. As mentioned earlier, the main source for PLL's using differential ring oscillator is thermal noise.

MOS device in the delay cell. According to [13], the MOSFET thermal noise is represented by the following equations. When the MOSFET is in the saturation, the noise current density is represented by (5.20).

$$\frac{\overline{i_n^2}}{\Delta f} = 4kT \frac{2}{3} g_m \quad (5.20)$$

where g_m is $\frac{W}{L} \mu C_{ox} (V_G - V_T)$. Also when the MOSFET is in the linear region, the noise current is (5.21).

$$\frac{\overline{i_n^2}}{\Delta f} = 4kT \frac{W}{L} \mu C_{ox} (V_G - V_T) \frac{2}{3} \frac{1 - (1-u)^3}{u(2-u)} \quad (5.21)$$

where $u = \frac{V_{ds}}{V_G - V_T}$. At $V_{ds} = 0$, (5.21) reduces to (5.22)

$$\frac{\overline{i_n^2}}{\Delta f} = 4kT g_{ds} \quad (5.22)$$

where g_{ds} is $\frac{W}{L} \mu C_{ox} (V_G - V_T)$.

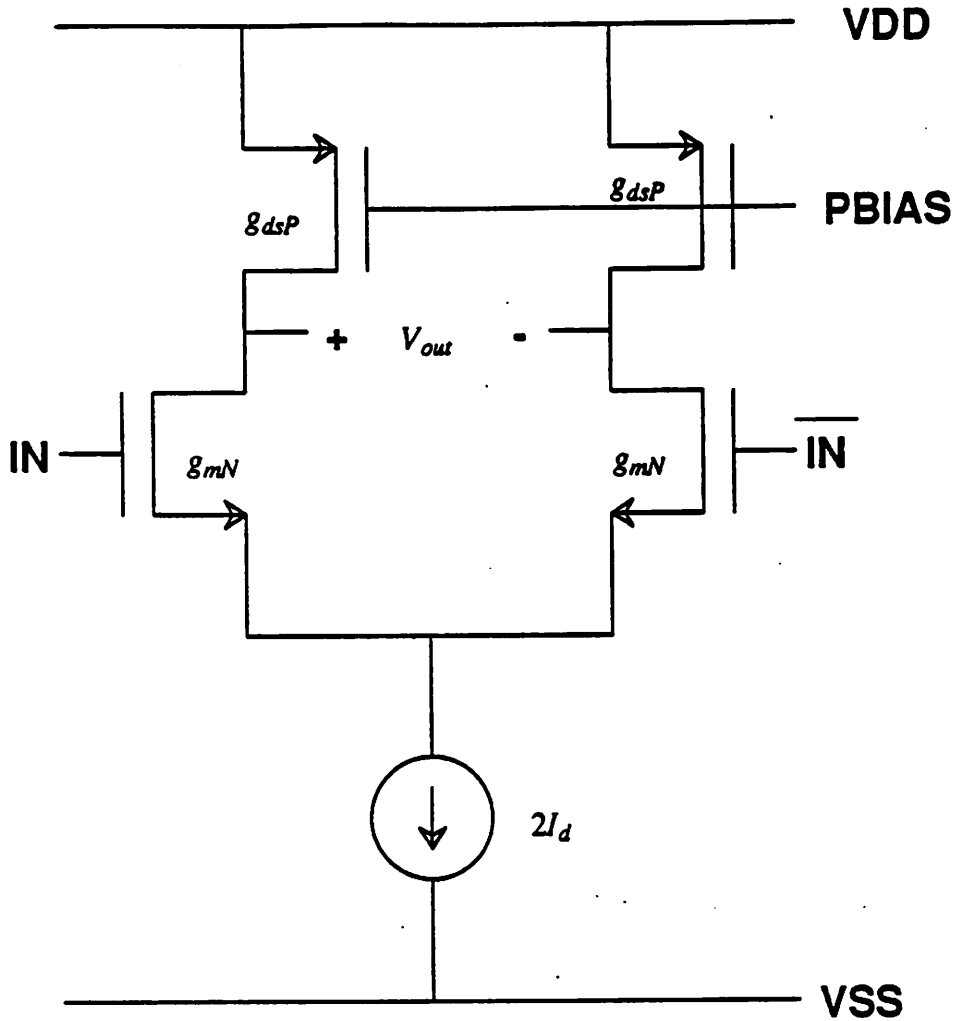


Figure 5.6 A differential Delay Cell

To find the noise voltage of the output of the delay cell as shown figure 5.6, equation (5.20) is applied to NMOS differential pairs in saturation, and equation (5.21) is applied to PMOS loads which are in triode region. The thermal noise current density from a PMOS load is then,

$$\overline{\frac{i_{Pn}^2}{\Delta f}} = 4kTg_{dsP}\gamma \quad (5.23)$$

where $\gamma = \frac{2}{3} \frac{1-(1-u)^3}{u(1-u)(2-u)}$. Also, the thermal noise current density from a NMOS device in the differential pair is (5.24).

$$\overline{\frac{i_{Nn}^2}{\Delta f}} = 4KTg_{mN} \quad (5.24)$$

Unlike the delay cell used in section 5.2.1, the differential delay cell is not a fast switching delay cell. Therefore, the Schmitt trigger assumption is not practical in this case. However, the linear system approximation and the constant noise current density assumption when each MOS device is on, can be used again. Since the noise currents generated from each MOS device are independent, the total differential noise power at the output of the differential pair is represented by (5.25) from (5.12).

$$\overline{v_n^2} = \frac{4kT}{C_g} \left(\frac{2g_{mN}}{3g_{dsP}} + \gamma \right) \left(1 - e^{-\frac{2g_{dsP}}{C_g} t_r} \right) \quad (5.25)$$

Here, t_r is not the delay per stage, but the total rise time since the next stage output is still slewing although the input voltage signal crosses the threshold voltage, therefore the next stage follows the noise process until the first noise process dies away. Then, t_r is not small when compared with $\frac{C_g}{g_{dsP}}$ and the exponential term can not be approximated as section 5.2.1. One interesting factor is, as t_r goes infinity, (i.e. using a non-switching input NMOS differential pair in the delay cell), the exponential term goes zero. The result is the same as calculated from a continuous stationary jitter source (never turns off) in section 5.2.3 and given by (5.26).

$$\overline{v_n^2} = \frac{4kT}{C_g} \left(\frac{2g_{mN}}{3g_{dsP}} + \gamma \right) \quad (5.26)$$

Hence, from (5.25), (5.26), it can be concluded that non-switching scheme is worse than full-switching scheme in jitter performance. In current design, the input pair switches completely, therefore, the rise time, t_r , is finite and given by (5.27) since the rise time is determined by slew rate.

$$t_r = \frac{C_g V_{pp}}{2I_d} \quad (5.27)$$

Here, I_d is half of the cell bias current flowing in the current source in the differential cell and V_{pp} is the peak-to-peak voltage swing of the output V_{out} . The slew rate also determines the slope of the rise time. Hence,

$$\frac{dv}{dt} = 2 \frac{I_d}{C_g} \quad (5.28)$$

Therefore, the jitter variance generated from one delay cell is represented by (5.29).

$$\overline{\Delta t^2} = \frac{KTC_g}{I_d^2} \left(\gamma + \frac{2g_{mN}}{3g_{dsP}} \right) \left(1 - e^{-\frac{g_{dsP} V_{pp}}{I_d}} \right) \quad (5.29)$$

If there are N stages in the ring oscillator ($2N$ switchings in one period), the total jitter variance in the ring oscillator is,

$$\overline{\Delta t_N^2} = \frac{2NKT C_g}{I_d^2} \left(\gamma + \frac{2g_{mN}}{3g_{dsP}} \right) (1 - \lambda) \quad (5.30)$$

where $\lambda = e^{-\frac{g_{dsP} V_{pp}}{I_d}}$. To see the effect of the size of the differential pair, each variables in (5.31) are replaced by the following basic parameters.

$$C_g \approx C_{gsn} + C_s = W_n L_n C_{ox} + C_s \quad (5.31)$$

where C_s is the stray or parasitic capacitance.

$$g_{mN} = \frac{W_n}{L_n} \mu_n C_{ox} (V_{gsn} - V_{Tn}) \quad (5.32)$$

$$g_{dsP} = \frac{W_p}{L_p} \mu_p C_{ox} (V_{gsp} - V_{Tp} - V_{dsp}) \quad (5.33)$$

$$I_d = \frac{1}{2} \frac{W_n}{L_n} \mu_n C_{ox} (V_{gsn} - V_{Tn})^2 \quad (5.34)$$

Then, the total jitter power is represented by (5.35).

$$\overline{\Delta t_N^2} = \frac{8NkT (W_n L_n C_{ox} + C_s) (1 - \lambda)}{(W_n / L_n)^2 \mu_n^2 C_{ox}^2 (V_{gsn} - V_{Tn})^4} \left[\gamma + \frac{2(W_n / L_n) \mu_n (V_{gsn} - V_{Tn})}{3(W_p / L_p) \mu_p (V_{gsp} - V_{Tp} - V_{dsp})} \right] \quad (5.35)$$

(5.35) can be divided into three terms. The first one is $\frac{K_1}{W_n^2}$, the second is $\frac{K_2}{W_n}$, and the last one is K_3 based on the dependencies on W_n . As W_n increases, the jitter power decreases because the first and the second terms decrease while the third term is constant. Therefore, to control the jitter to an acceptable value, it is necessary to use large device for the differential pairs in the delay cell. Also, from (5.30), to reduce the jitter, large cell bias current is preferred. If (5.35) is further approximated to give some insight, then,

$$\overline{\Delta t_N^2} = \frac{8NkTL_n^3}{\mu_n^2 W_n C_{ox} V_{dsat}^4} \left(1 + \frac{2}{3} a_V\right) \left(1 - e^{-\frac{2V_{pp}}{a_V V_{dsat}}}\right) \quad (5.36)$$

where a_V is stage gain given by $a_V = \frac{g_{mN}}{g_{dsP}}$. (5.36) can be further simplified into (5.37).

$$\overline{\Delta t_N^2} = \tau_T^2 \frac{8NkT}{C_{gn} V_{dsat}^2} \left(1 + \frac{2}{3} a_V\right) \left(1 - e^{-\frac{2V_{pp}}{a_V V_{dsat}}}\right) \quad (5.37)$$

where τ_T is the NMOS device transit time given by $\frac{L_n^2}{\mu_n V_{dsat}}$. Hence, the normalized jitter variance is given by (5.38).

$$\frac{\overline{\Delta t_N^2}}{\tau_T^2} = \frac{8N}{V_{dsat}^2} \left(\frac{kT}{C_{gn}}\right) \left(1 + \frac{2}{3} a_V\right) \left(1 - e^{-\frac{2V_{pp}}{a_V V_{dsat}}}\right) \quad (5.38)$$

From (5.38), the normalized phase jitter based on the device transit time, τ_T , is proportional to $\sqrt{\frac{kT}{C_{gn}}}$, inversely proportional to V_{dsat} and a weak function of the stage gain, a_V .

5.2.3 Non-Saturated Ring Oscillator

In this third ring oscillator type, the delay cell devices never turn off. Therefore, the

delay cell can be approximated as an linear amplifier. Here, again the differential ring oscillator in figure 5.6 is used as an example. However, in this case, the input NMOS pair devices are always on. Equations (5.23) and (5.24) are still valid since they are derived when PMOS and NMOS devices are on. Again, if the equivalent noise circuit can be represented as shown in figure 5.4, then the total differential noise power at the output of the differential pair is represented by (5.39).

$$\overline{v_n^2} = \int_0^{\infty} \left(2 \frac{\overline{i_{Nn}^2}}{\Delta f} + 2 \frac{\overline{i_{Pn}^2}}{\Delta f} \right) \frac{2df}{g_{dsP}^2 (1 + (f/f_{3dB})^2)^2} \quad (5.39)$$

Since the 3 dB cutoff frequency of the differential pair is $\frac{g_{dsP}}{2\pi C_g}$, (5.39) becomes (5.40).

$$\overline{v_n^2} = \frac{4kT}{3C_g} \left(\gamma + \frac{2g_{mN}}{3g_{dsP}} \right) \quad (5.40)$$

This result is exactly same as (5.26). Therefore, the jitter variance generated from one delay cell is represented by (5.41).

$$\overline{\Delta t^2} = \frac{kTC_g}{I_d^2} \left(\gamma + \frac{2g_{mN}}{3g_{dsP}} \right) \quad (5.41)$$

Again, if there are N stages in the ring oscillator, the total jitter variance in the ring oscillator is,

$$\overline{\Delta t_N^2} = \frac{2NkTC_g}{I_d^2} \left(\gamma + \frac{2g_{mN}}{3g_{dsP}} \right) \quad (5.42)$$

Using the same basic parameters used in section 5.2.2, the total jitter can be approximated by (5.43).

$$\overline{\Delta t_N^2} = \tau_f^2 \left(\frac{8NkT}{C_{gn} V_{dsain}^2} \right) \left(1 + \frac{2}{3} a_v \right) \quad (5.43)$$

Hence, from (5.43), as before the jitter is proportional to the device transit time, proportional to $\sqrt{\frac{kT}{C_{gn}}}$, inversely proportional to V_{dsain} , and linearly dependent on stage

gain. Here, the phase jitter is worse than that of the slow-slewing saturated ring oscillator by $1/(1-e^{-\frac{2V_{pp}}{a_v V_{dmsl}}})$.

5.2.4 Comparison of Ring Oscillator Jitter Performances

In this section, the jitter performance of the three different types of ring oscillators is compared in table 5.1. In this table, key assumptions are summarized and the jitter equations are described in terms of the intrinsic device transit time τ_T . From this table, it can be concluded that the fast-slewing saturated ring oscillator can achieve the best jitter performance, the slow-slewing saturated ring oscillator shows the second best, and the non-saturated ring oscillator shows the worst.* Therefore, a ring oscillator composed of completely switching delay cells with fast slew rate is desirable when low jitter performance is required.

* It is not immediately obvious that the fast-slewing saturated ring oscillator shows better jitter performance than the slow-slewing saturated ring oscillator since the comparison was performed between single-ended and differential delay cells, respectively. However, if a differential delay cell (figure 5.6) with high slewing input buffer is used in the fast-slewing saturated ring oscillator (which perhaps could be implemented in BiCMOS process), the comparison will be obvious. Here, from (5.25), the rise time, t_r is replaced by nominal delay time, τ for the fast-slewing saturated type. Therefore, total jitter in the N stage ring oscillator is given by (*.1) since $\tau \ll \frac{C_g}{2g_{ds}}$.

$$\overline{\Delta t_N^2} = \frac{2NkTC_g \Delta V_{g_{dsP}}}{I_d^2} \left(\gamma + \frac{2g_{mN}}{3g_{dsP}} \right) \quad (*.1)$$

Using the same parameters used in section 5.2.2, equation (*.1) becomes (*.2).

$$\frac{\overline{\Delta t_N^2}}{\tau_T^2} = \frac{8NkT}{C_g V_{dmsl}} \left(\frac{V_{pp}}{a_v V_{dmsl}} \right) \left(1 + \frac{2}{3} a_v \right) \quad (*.2)$$

Here, $V_{pp} = 2\Delta V$ is assumed. From (*.2), the jitter of the fast-slewing saturated type is 2 X smaller than the minimum possible jitter of the slow-slewing saturated type. The reason is, in the fast-slewing saturated type, the noise process affects the delay only up to the time point when the output of the previous delay cell crosses the threshold voltage. But, in the slow-slewing case, the noise process affects the delay for the full rise time and this gives the penalty of 2.

Table 5.1

ring oscillator type	key assumptions	$\overline{\Delta t_j^2}/\tau_j^2$
fast-slewing saturated	non-stationary noise process fast-slewing input buffer full switching, single-ended	$\frac{64NkT}{3C_g V_{dsat}^2} \left(\frac{\Delta V}{V_{dsat}} \right)$
slow-slewing saturated	non-stationary noise process no buffer full switching, differential	$\frac{8NkT}{C_g V_{dsat}^2} \left(1 + \frac{2}{3} a_v \right) \left(1 - e^{-\frac{2V_{pp}}{a_v V_{dsat}}} \right)$
non-saturated	stationary noise process no buffer no switching, differential	$\frac{8NkT}{C_g V_{dsat}^2} \left(1 + \frac{2}{3} a_v \right)$

5.3 PLL Phase Noise Accumulation

In this section, based on the jitter equations derived above, the PLL jitter at the system level is analyzed. In the ring oscillator, if a clock pulse has a phase shift at a certain time, the phase shift will remain forever because there is no correction mechanism in the ring oscillator. For example, if a phase shift of a clock pulse at time nT from a single jitter event which is caused by an accumulated jitter through the inverters in the ring oscillator occurred, the phase shift will persist in the following clock pulses as shown in figure 5.7(a).

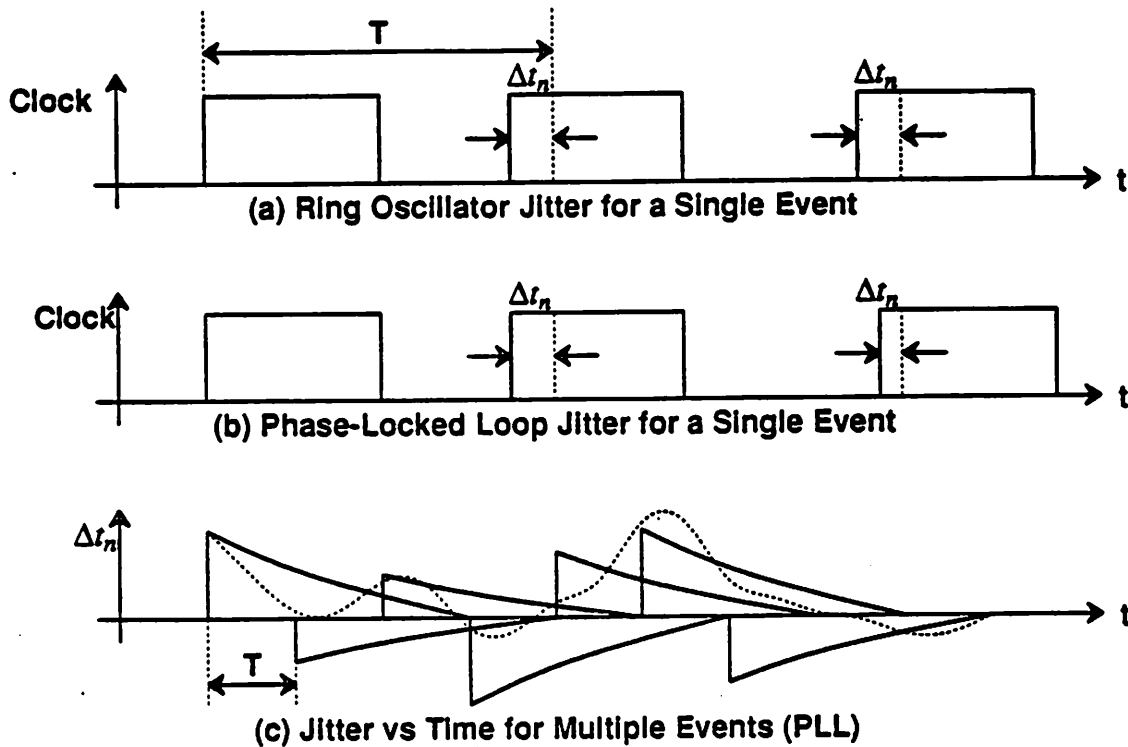


Figure 5.7 PLL Jitter Accumulation

However, if the ring oscillator is locked into the reference clock via a phase-locked loop, the phase shift will be reduced because of the correction mechanism generated by the phase-locked loop as follows. First, the phase detector detects the phase shift, controls the charge-pumping circuit to generate charge to be stored in the loop capacitor and to control the ring oscillator to adjust phase shift in the correct direction. Since the amount of phase adjust is small, the phase shift is not corrected in one cycle, but it is reduced gradually in the following cycles as shown in figure 5.7(b). If we measure the phase shifts for multiple jitter events and draw a diagram of phase shifts vs time, we can get picture like figure 5.7(c). Note the different scale here. Therefore, the result total phase jitter for those multiple events is shown in the dotted line in figure 5.6(c).

To find the rms jitter, a PLL which uses a sequential phase detector and a charge-pumping circuit can be represented by a simple discrete-time model as shown figure 5.8 since the charge-pumping circuit can be represented by a switch connected to current sources. Here, the phase jitter is a noise source added to the output clock phase of the voltage controlled oscillator.

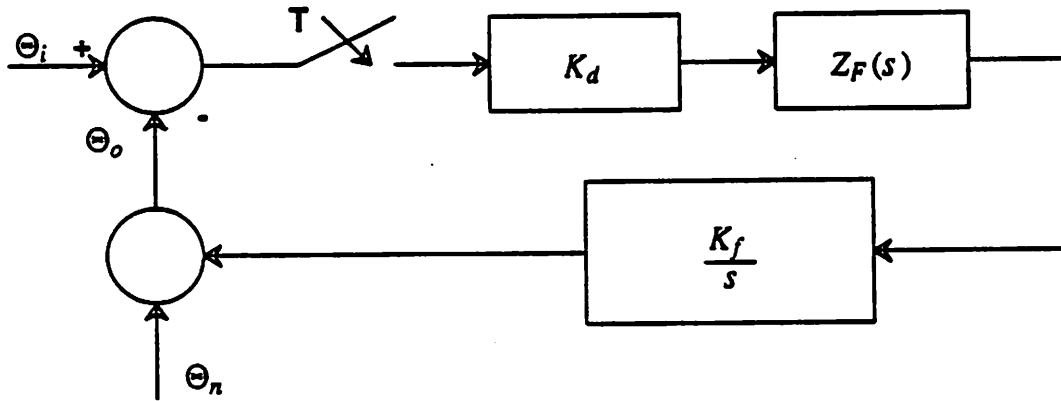


Figure 5.8 PLL Linearized Model

Then, the transfer function of $\Theta_{on}(z)$ in terms of $\Theta_n(z)$, the phase jitter, is represented by (5.44).

$$\Theta_{on}(z) = \frac{\Theta_n(z)}{1 + K_d K_f Z_F'(z)} \quad (5.44)$$

where K_d and K_f are the phase detector gain and VCO gain respectively, and $Z_F'(z)$ is the z-transform of the sampled version of $Z_F(s)/s$. If the loop filter is assumed to be a lead-lag filter composed of a resistor, R , in series with a capacitor, C . Then, the above equation becomes (5.45).

$$\Theta_{on}(z) = \frac{(1-z^{-1})^2 \Theta_n(z)}{(1 + K_d K_f) + \left(\frac{K_d K_f T}{RC} - K_d K_f - 2 \right) z^{-1} + z^{-2}} \quad (5.45)$$

where $K_d = \frac{I_p RT}{2\pi}$, and $K_f = \frac{df}{dv}$. I_p indicates charge pumping current. In PLL's,

usually,

$$\frac{K_d K_f T}{RC} \ll K_d K_f \ll 1 \quad (5.46)$$

Since $\frac{K_d K_f T}{RC} \ll K_d K_f$ and $K_d K_f \ll 1$, we can ignore $\frac{K_d K_f T}{RC}$ and put $\epsilon = K_d K_f$.

Then, the transfer function is,

$$\Theta_{on}(z) = \frac{(1-z^{-1})\Theta_n(z)}{1-(1-\epsilon)z^{-1}} \quad (5.47)$$

Since the phase jitter coming from the ring oscillator is a sequence of step phase jumps which have random magnitudes, a single step phase jump event at time n can be represented by (5.35). Here, the magnitude of the step is Δt_n .

$$\Theta_n(z) = \frac{2\pi\Delta t_n}{T(1-z^{-1})} \quad (5.48)$$

Hence the output jitter in z -domain is,

$$\Theta_{on}(z) = \frac{2\pi\Delta t_n}{T(1-(1-\epsilon)z^{-1})} \quad (5.49)$$

Therefore, in time domain, the output phase shift for a single event of the phase jitter is represented by (5.50).

$$\Theta_{on}(nT) = \frac{2\pi\Delta t_n}{T}(1-\epsilon)^n u(nT) \quad (5.50)$$

where $u(nT)$ is the unit step function. For all events up to time nT , the sum of output phase shifts is represented by (5.51)

$$\Theta_{tot}(nT) = \sum_{k=-\infty}^n \frac{2\pi\Delta t_k}{T}(1-\epsilon)^{n-k} \quad (5.51)$$

To find the RMS jitter, the expectation of the square of the sum is calculated as follows.

$$E[\Theta_{tot}^2(nT)] = E\left[\left(\sum_{k=-\infty}^n \frac{2\pi\Delta t_k}{T}(1-\epsilon)^{n-k}\right)\left(\sum_{l=-\infty}^n \frac{2\pi\Delta t_l}{T}(1-\epsilon)^{n-l}\right)\right] \quad (5.52)$$

Since Δt_k and Δt_l are not correlated, the $E[\Delta t_k \Delta t_l] = 0$ when $k \neq l$. When $k = l$, $E[\Delta t_k \Delta t_k]$ can be replaced by $\overline{\Delta t_N^2}$. Therefore, (5.52) becomes (5.53).

$$E[\Theta_{i\alpha}^2(nT)] = \left(\frac{2\pi}{T}\right)^2 \overline{\Delta t_N^2} \sum_{n=0}^{\infty} (1-\varepsilon)^{2n} \quad (5.53)$$

Again, (5.53) can be simplified as (5.54)

$$E[\Theta_{i\alpha}^2(nT)] = \left(\frac{2\pi}{T}\right)^2 \overline{\Delta t_N^2} \frac{1}{2\varepsilon(1-1/2\varepsilon)} = \frac{2\pi^2 \overline{\Delta t_N^2}}{\varepsilon T} \quad (5.54)$$

Note that the expectation value of the phase jitter is independent of nT , the time instant. Hence, the RMS phase jitter is,

$$\sqrt{E[\Theta_{i\alpha}^2(nT)]} = \sqrt{\frac{2}{\varepsilon} \frac{\pi \Delta t_{rms}}{T}} = \sqrt{\frac{2}{K_d K_f} \frac{\pi \Delta t_{rms}}{T}} \quad (5.55)$$

where Δt_{rms} is $\sqrt{\overline{\Delta t_N^2}}$.

Comparing this value with the oscillator jitter, we can find the RMS time jitter in a PLL is $\sqrt{\frac{1}{2K_d K_f}}$ times larger than that of the ring oscillator. Here, the PLL accumulated the phase jitter. In the case of the analog PLL used in the hybrid clock recovery circuit, the ratio is large. Therefore, the ring oscillator jitter should be very small. In chapter 5.5, an example case is given and the phase jitter is calculated.

5.4 DLL Phase Jitter

An alternative scheme to generate multi-phase clocks is to use a delay-locked loop (DLL) [11]. Figure 5.9 shows the DLL schematic diagram.

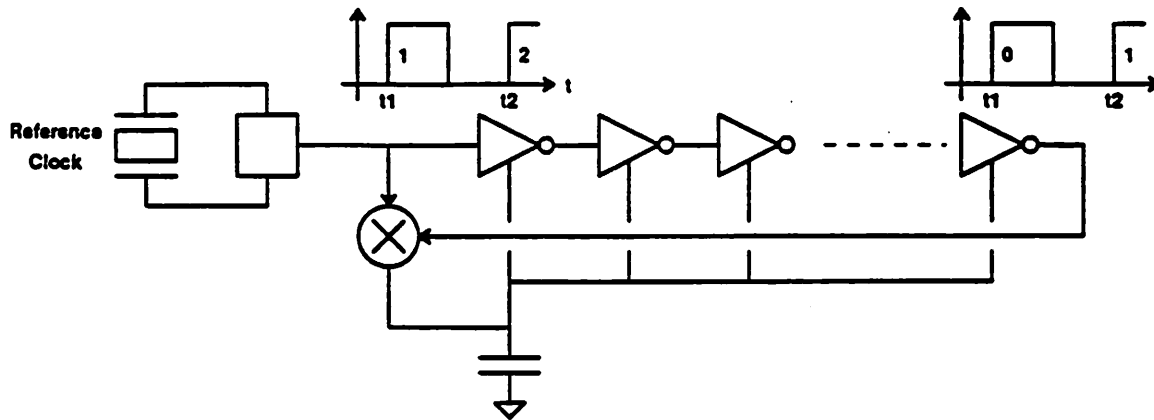


Figure 5.9 DLL Schematic Diagram

Here, the reference clock coming from a crystal oscillator is fed to the input of the delay line, and the rising edge of the output of the delay line is compared to that of the reference clock. Since the rising edge of the reference clock reaches the output of the delay line by passing through all delay cells, the total delay is one period of the reference clock. If the total delay is smaller than the period of the reference clock, the loop creates a correction voltage and causes the total delay to be the same as the period of the reference clock. For example, if the rising edge of the output is lagging behind of that of the reference clock, the phase detector generates positive correction charge, increasing the output of the loop filter and reducing the total delay of the delay line. Since the output of the loop filter just changes the phase of the output of the delay line, the loop does not have any extra poles as a PLL does. Therefore, the stability problem is relaxed and a simple capacitor loop filter can be used without any stability consideration.

In the delay-locked loop, the phase jitter is not accumulated because the jitter created from one event of phase shift flushes away at the end of delay cell and never accumulates. To see this, a simplified discrete time DLL model is used in figure 5.10.

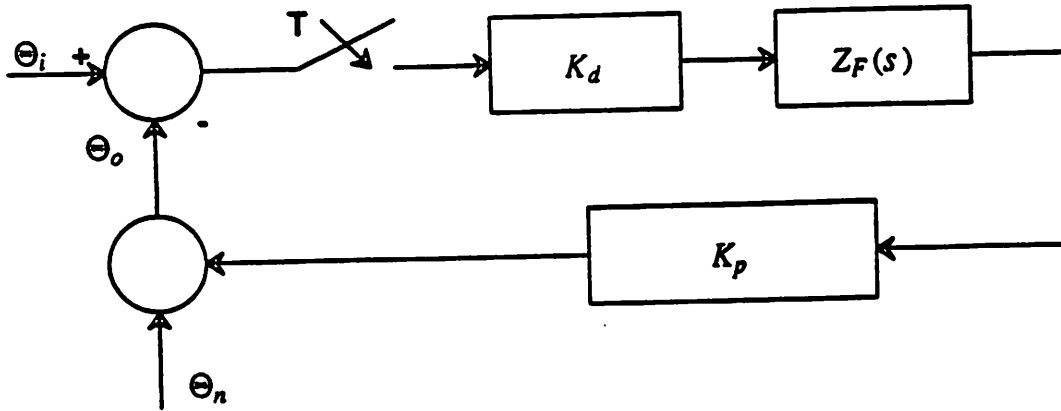


Figure 5.10 DLL Linearized Model

Here, the DLL has a voltage-controlled or current-controlled delay generator and the PLL has a voltage controlled oscillator. If the loop filter in the DLL is a single capacitor, the transfer function for output jitter vs delay generator in z-domain is represented by (5.56).

$$\Theta_{on}(z) = \frac{(1-z^{-1})\Theta_n(z)}{(1 + \frac{K_d K_p}{C})z^{-1}} \quad (5.56)$$

where K_d is phase gain and given by $\frac{I_p T}{2\pi}$, and K_p is phase gain and given by $\frac{d\theta}{dv}$. In DLL, usually,

$$\frac{K_d K_p}{C} \ll 1 \quad (5.57)$$

From (5.57), we can put $\epsilon = \frac{K_d K_p}{C}$. Then, the transfer function is represented by (5.58).

$$\Theta_{on}(z) = \frac{(1-z^{-1})\Theta_n(z)}{(1+\epsilon)z^{-1}} = \left(1 - \frac{\epsilon}{1-(1-\epsilon)z^{-1}}\right)\Theta_n(z) \quad (5.58)$$

The delay generator jitter is represented by (5.59) because the jitter is a single phase shift event and flushes away through the end of the delay chain.

$$\Theta_n(z) = \frac{2\pi\Delta t_n}{T} \quad (5.59)$$

Hence, the output jitter caused by a single phase shift event is,

$$\Theta_{on}(z) = \frac{2\pi\Delta t_n}{T} \left(1 - \frac{\epsilon}{1 - (1-\epsilon)z^{-1}}\right) \quad (5.60)$$

In time domain, the output jitter is represented by (5.61).

$$\Theta_{on}(nT) = \frac{2\pi\Delta t_n}{T} (\delta(nT) - \epsilon(1-\epsilon)^n u(nT)) \quad (5.61)$$

For all events up to time nT , the sum of jitter is,

$$\Theta_{tot}(nT) = \frac{2\pi}{T} \sum_{k=-\infty}^n \Delta t_k (1 + \epsilon(1-\epsilon)^{n-k}) \quad (5.62)$$

Hence, the RMS jitter can be found as was (5.63). The result is,

$$\sqrt{E[\Theta_{tot}^2(nT)]} = \frac{2\pi\Delta t_{rms}}{T} \quad (5.63)$$

Again, note that the RMS value is not function of nT .

In DLL case, as seen from (5.63), there is no jitter accumulation in a DLL. Therefore, the jitter of the DLL is much smaller than that of the PLL which is normally used in a clock recovery circuit.

5.5 Example: PLL Noise Analysis with Slow-Slewing Saturated Ring Oscillator

As an example, the PLL jitter using a slow-slewing saturated ring oscillator using differential delay cell shown in figure 5.6 is calculated. Here, if $N=32$, $C_g=352fF$, $I_d=250\mu A$, $\frac{g_m}{g_{ds}}=2$, $V_{pp}=1.0Volts$, and $V_{dsat}=0.3Volts$, then $\sqrt{\Delta t_N^2}=0.94ps$ using (5.30). This is a reasonably good result when compared with the measured value in chapter 7.

The PLL noise jitter is then calculated using (5.55). Since the jitter accumulation factor is $\sqrt{\frac{1}{2K_dK_f}}$, first K_d is calculated. Since K_d is $\frac{I_pRT}{2\pi}$, $K_f=5\text{MHz/volt}$, and $R=180\Omega$, $K_dK_f=1.8\times 10^{-4}$. Here, I_p is the value of the current generated from charge-pumping circuit. Therefore, the accumulation factor is about 53. Hence, the RMS jitter from the PLL is 50ps . The measured jitter including sampling jitter is about 100ps , and this is approximately equal to the calculated value.

The next four graphs show the relations between PLL jitter and several important parameters. Figure 5.11, 5.12, 5.13, and 5.14 show PLL jitter as a function of stage gain, input gate width, channel length, and gate capacitance per unit area (μm^2), respectively. Here again, the type of the ring oscillator is assumed to be a slow-slewing saturated shown in figure 5.6 and equation (5.35) is used to calculate the jitter. The nominal parameters used in this equation are summarized in Table 5.2.

If the prototype is implemented in a scaled CMOS technology with constant voltage scaling (table 4.1), jitter performance of the prototype can be predicted from the above four figures and equation (5.36). In constant voltage scaling, the gate width and channel length are divided by a constant factor k , the oxide thickness is divided by the same factor, and substrate doping is multiplied by the factor of k^2 , while the voltage source and V_T is maintained to a constant. Figure 5.12 shows the jitter performance degradation with the scaled gate width. On the other hand, figure 5.13 and 5.14 show the jitter performance enhancement with the scaled channel length and gate oxide, respectively. Here, $C_{ox}=\frac{\epsilon_{ox}}{t_{ox}}$. As a result, the time jitter decreases by the factor of $k^{3/2}$ (5.36). However, the relative time jitter to the intrinsic device transit time increases by the factor of $k^{1/2}$ from (5.38). Since the relative jitter performance determines the maximum phase resolution of the hybrid analog/digital PLL, the constant voltage scaling degrades the phase resolution and the maximum effective decoding window defined in chapter 3.

Table 5.2

Parameter	Value
W_n	160 μm
L_n	2 μm
W_p	36 μm
L_p	3 μm
$V_{gsn} - V_{Tn}$	0.3 Volts
$V_{gsp} - V_{Tp}$	2.2 Volts
C_{ox}	0.84 fF/ μm^2
C_{jo}	0.20 fF/ μm^2
V_{pp}	1 Volts

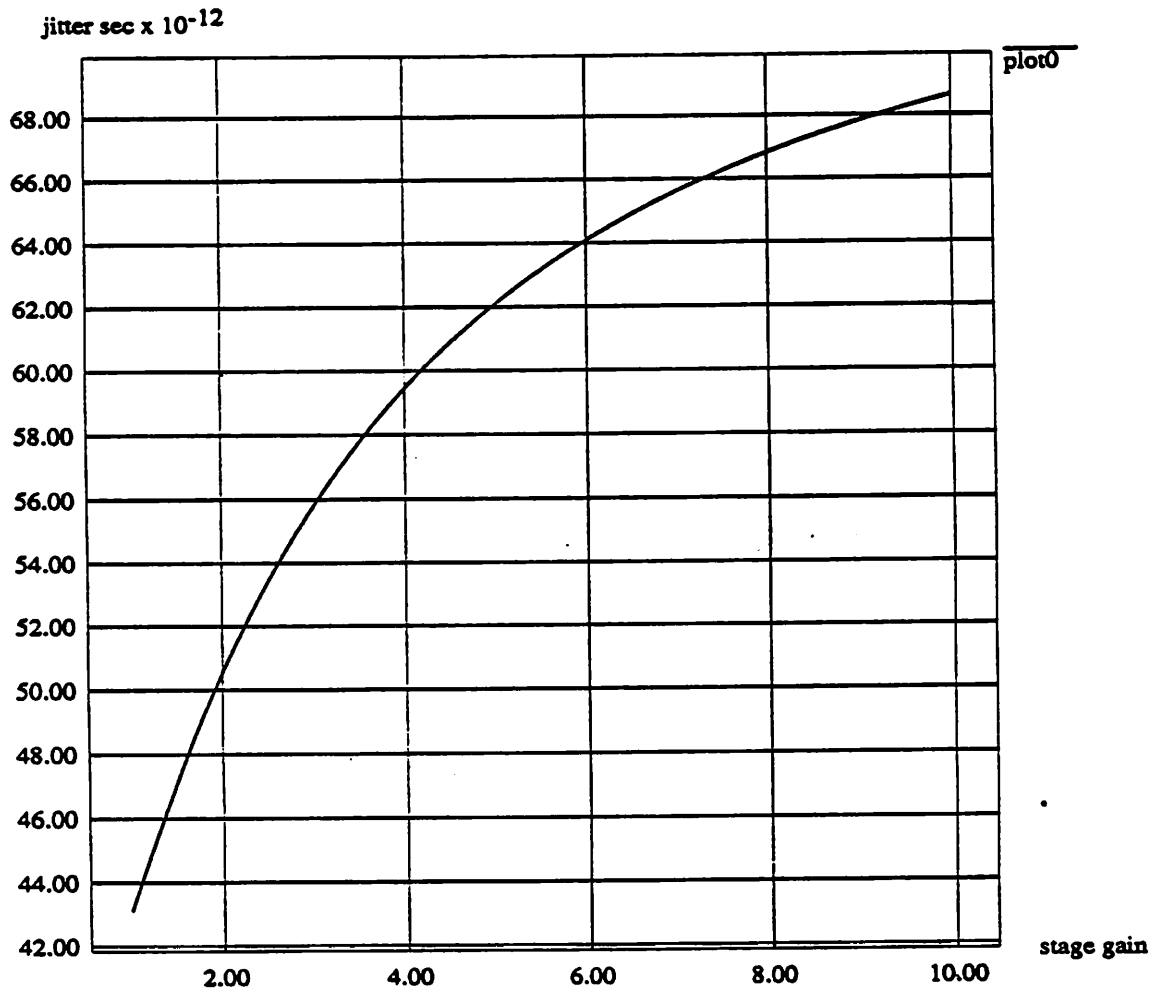


Figure 5.11 Phase Jitter vs Stage Gain

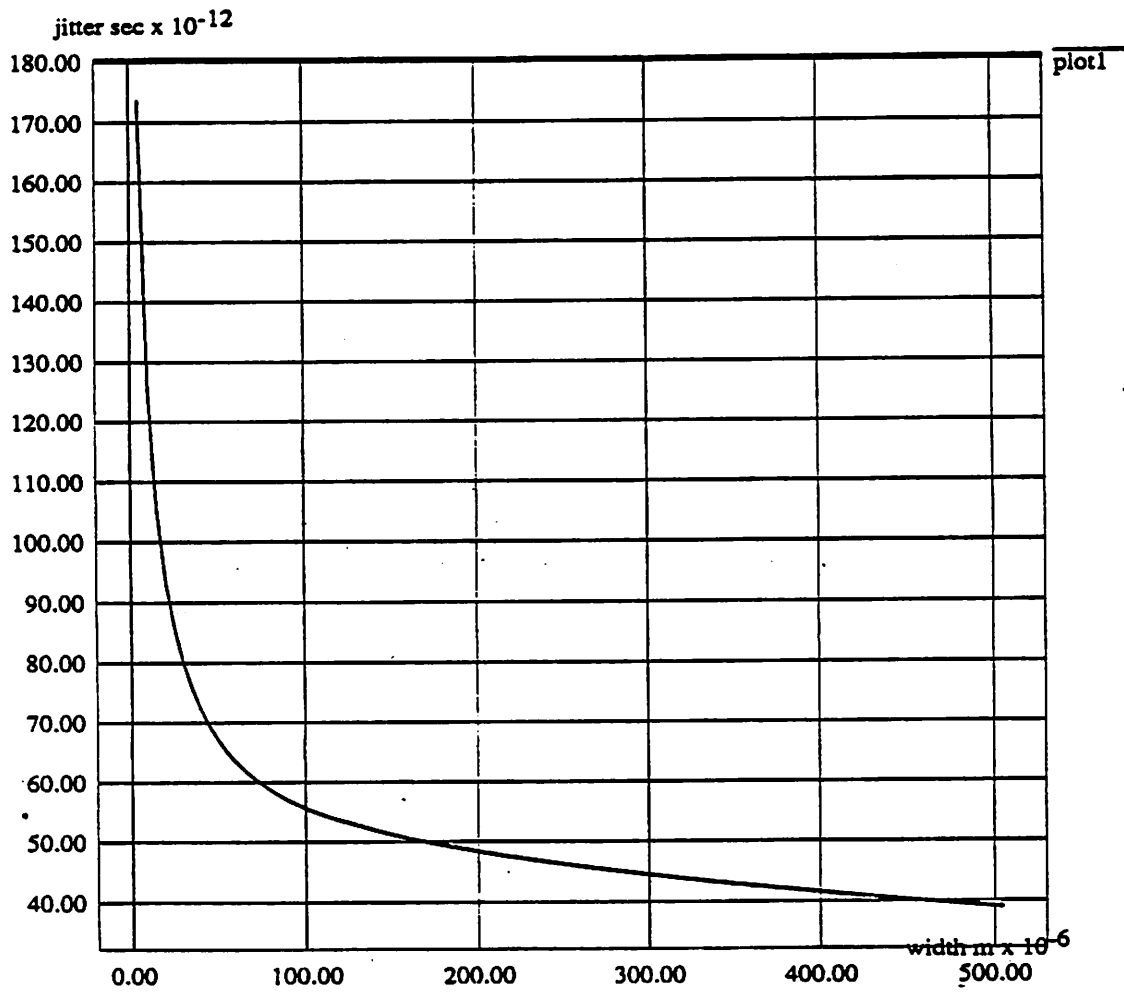


Figure 5.12 Phase Jitter vs Input Gate Width

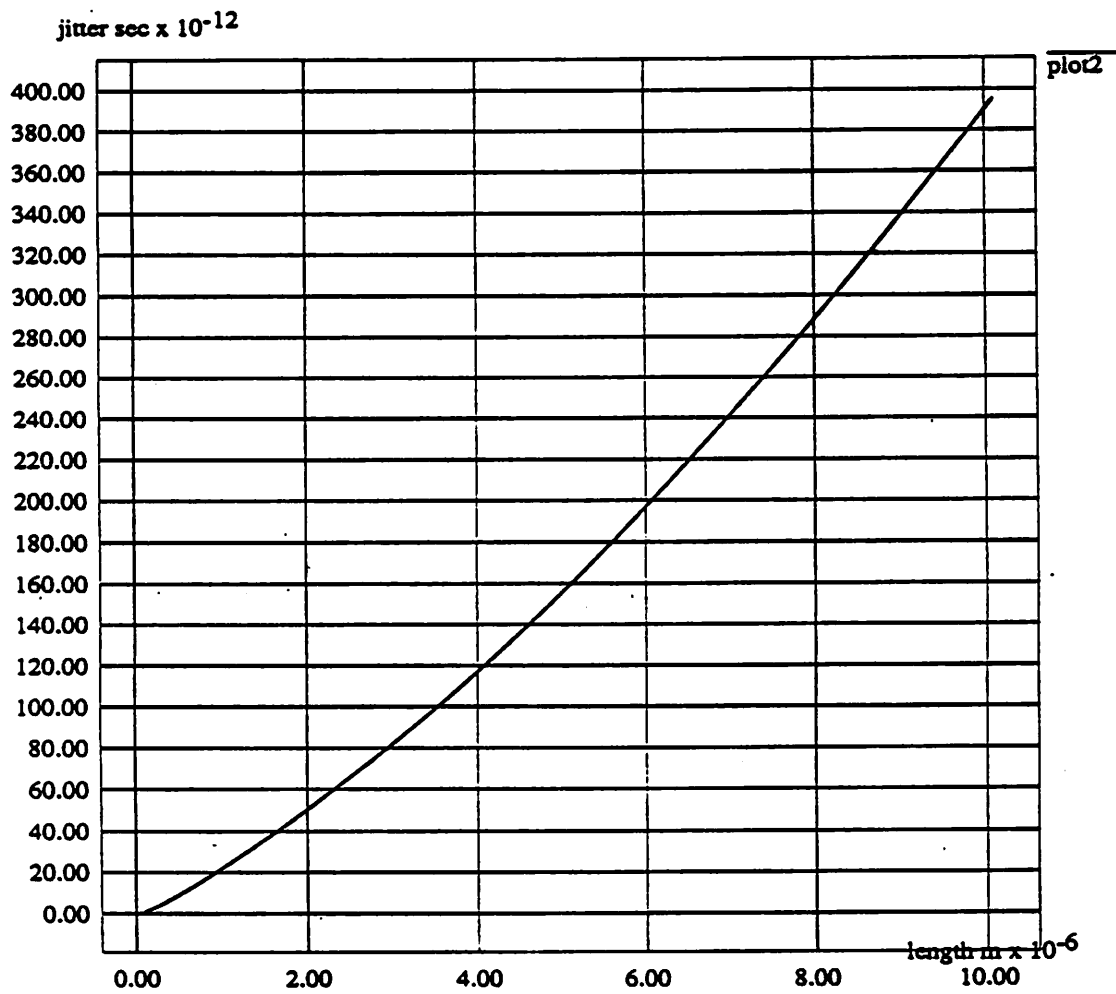


Figure 5.13 Phase Jitter vs Input Channel Length

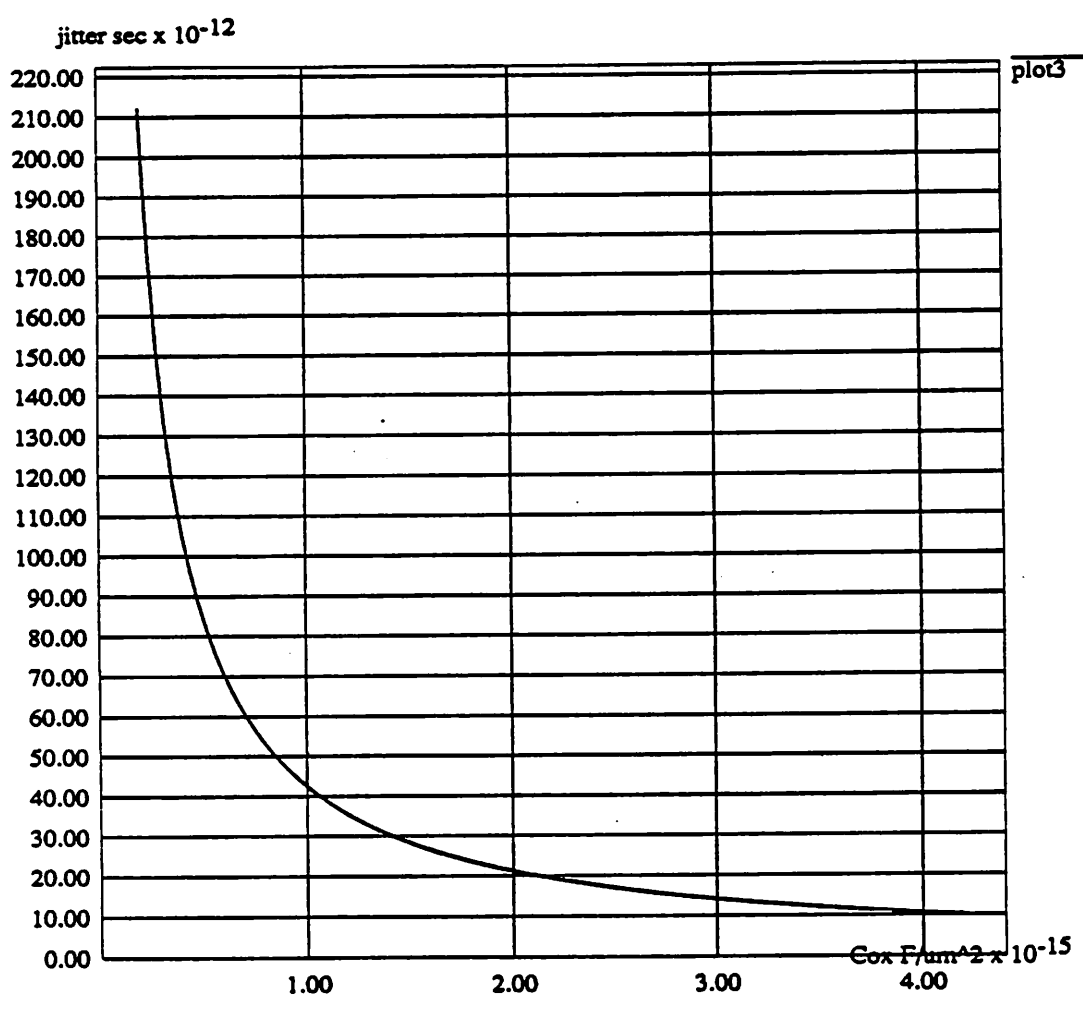


Figure 5.14 Phase Jitter vs Gate Capacitance per Unit Area

5.6 PLL Phase Jitter System Simulation

To verify the PLL jitter accumulation phenomenon which was analyzed theoretically in section 5.3, a simple behavioral simulator using the C programming language was written. Figure 5.15 shows the block diagram of the simulator.

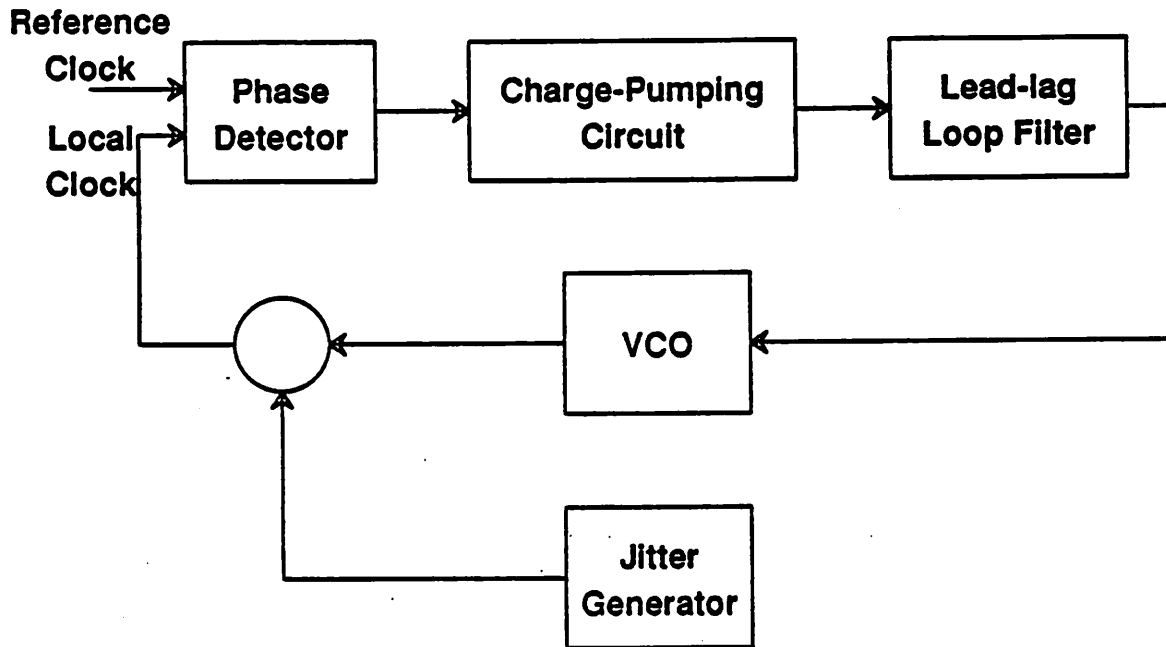


Figure 5.15 Analog Phase-Locked Loop Jitter Model

Here the VCO phase jitter is modeled by a Gaussian Markov Process and generated from the gaussian random number generation module. Figure 5.16 shows the jitter process.

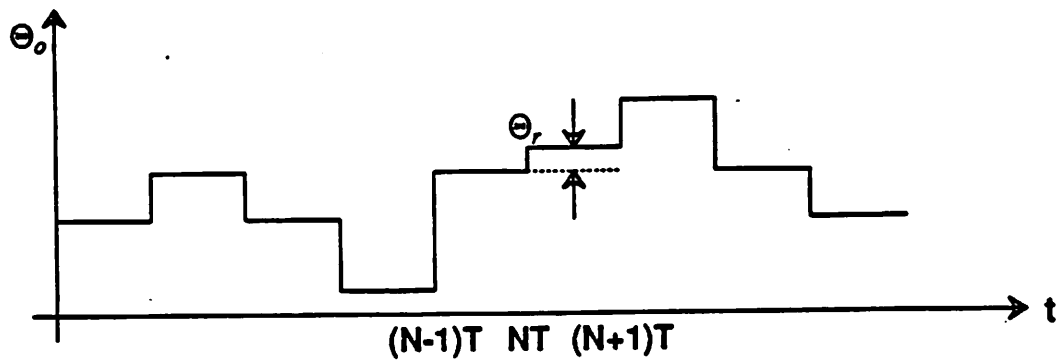


Figure 5.16 Markov Process Model for Random Jitter in Ring Oscillator

The whole PLL loop has a phase detector and a lead-lag loop filter, as described in chapter 4. The VCO is modeled by a linear FM modulator with phase jitter source mentioned above. Figure 5.17 shows the phase jitter accumulation for the first 100 cycles and figure 5.18 shows the phase jitter accumulation for 50,000 cycles with decimation ratio of 100. From these pictures, significant phase jitter accumulation is evident.

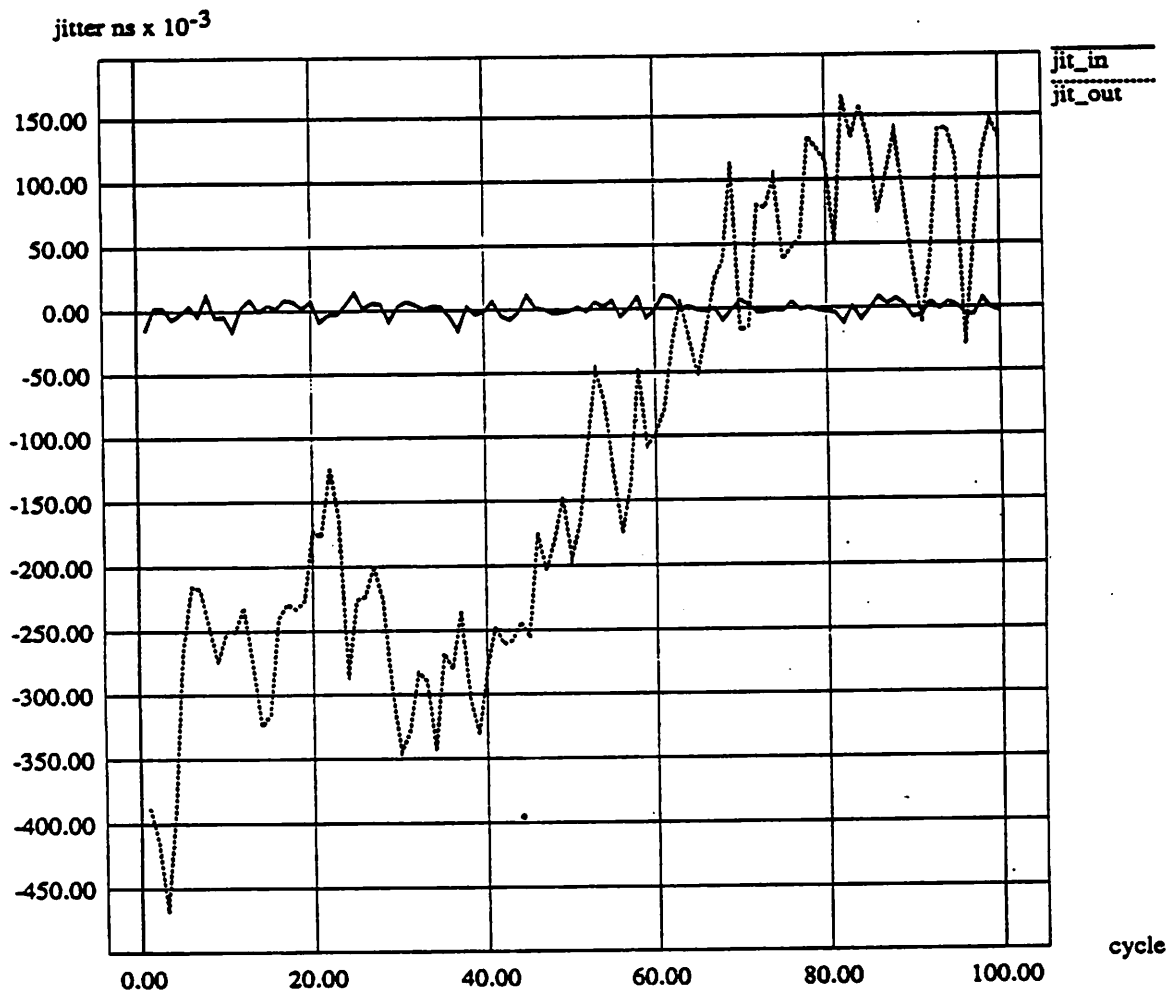


Figure 5.17 Phase Jitter Accumulation for 100 Cycles

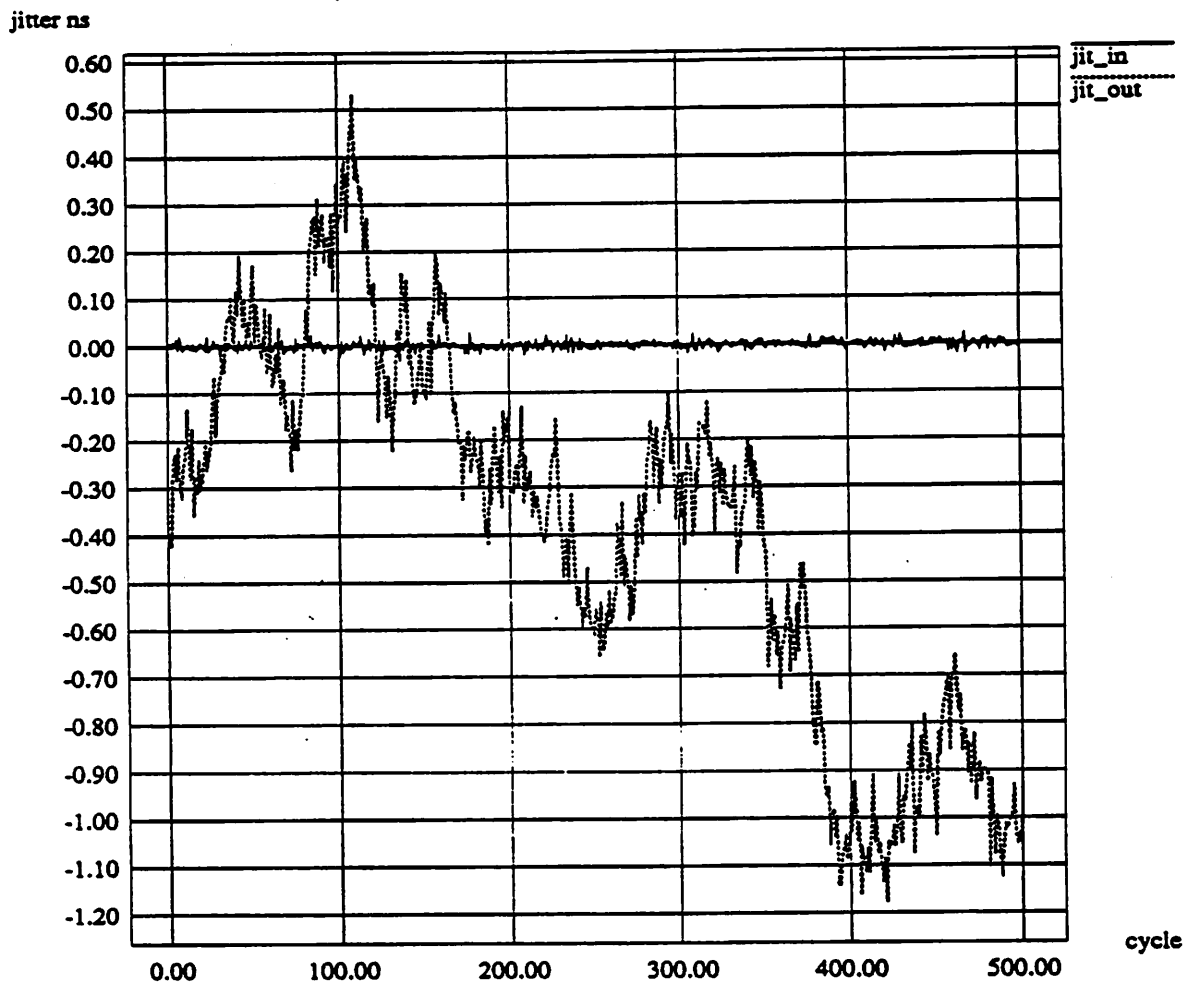


Figure 5.18 Phase Jitter Accumulation for 50,000 Cycles

Chapter 6

Optimal K Sequence for the DPLL

6.1 Introduction

Since the hybrid analog/digital clock recovery architecture allows the process of acquisition and tracking to move from the analog domain to the digital domain, great versatility is gained from the digital signal processing. One of the main goals of the DPLL is to achieve fast acquisition while in addition still satisfying the constraint of achieving large input jitter reduction. But two requirements are in conflict with each other since fast acquisition requires very wide bandwidth in the DPLL loop, while large noise reduction requires very narrow bandwidth. To meet both requirements, a gear-shifting method has been used in conventional clock recovery circuits. However, since the conventional clock recovery circuits rely on analog PLL's, and the loop capacitors, or loop resistors of the analog PLL should vary in order to change the loop bandwidth, the selection of gear-shifting has been limited to very simple ones. On the other hand, the DPLL has no limitation in the selection of the gear-shifting parameters because there are no physical resistors and capacitors which determine the loop bandwidths and because a simple multiplying constant, K can be changed to control the loop bandwidth. Hence, the study of an optimal K sequence which can meet the requirements of fast acquisition in preamble period and large jitter reduction in data decoding period is presented in this chapter.

6.2 Problem Definition

Figure 6.1 shows the DPLL block diagram. Here, each block can be implemented by a real digital multiplier, digital subtractor, accumulators and so forth. Here

the K is the control parameter for the loop bandwidth. If K is ∞ , the loop has infinite bandwidth, and if K is 0, the loop has zero bandwidth.

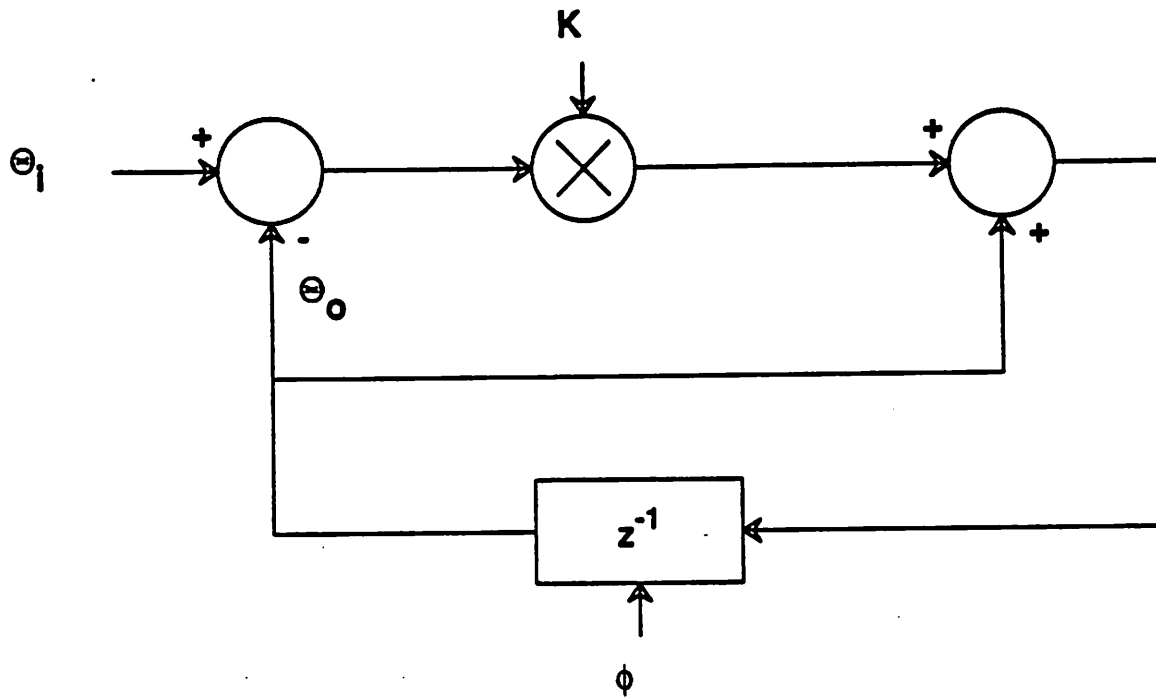


Figure 6.1 Linearized First Order Digital PLL Model

Here, each symbol is defined as follows.

Θ_s : real input signal phase (fixed)

$\Theta_N(n)$: jitter noise phase at time nT

$\Theta_i(n)$: input signal phase at time nT

$\Theta_o(n)$: DPLL output signal phase at time nT

$K(n)$: K (control parameter) number at time nT

Therefore, the input signal phase is represented by (6.1)

$$\Theta_i(n) = \Theta_s + \Theta_N(n) \quad (6.1)$$

Since the jitter noise usually behaves as a wide-sense stationary white gaussian process, the jitter mean value and variance is represented by (6.2), (6.3) respectively.

$$E[\Theta_N(n)] = 0 \quad (6.2)$$

$$E[\Theta_N^2(n)] = \sigma_N^2 \quad (6.3)$$

From the model shown in figure 6.1, the output phase from DPLL is given by (6.4).

$$\Theta_o(n+1) = \Theta_o(n) + K(n)(\Theta_i(n) - \Theta_o(n)) \quad (6.4)$$

Rearranging it, then (6.5) is obtained.

$$\Theta_o(n+1) = (1-K(n))\Theta_o(n) + K(n)\Theta_i(n) \quad (6.5)$$

Since (6.5) is a equation of for a nonlinear time-varying system, finding the optimum K 's is a problem of optimization of a nonlinear time-varying system. In general, it is very difficult to find a solution of the optimal parameters in a nonlinear time varying system. However, in this case, an optimum set of K values exists and is found as shown in the following sections.

Two different situations occur in the optimization process. First, for each instant including preamble period, the difference between output phase and real input phase should be minimized. i.e. the output phase should be the best estimation of the real input phase. Since the input jitter is statistical process, the optimization is also based on a statistical optimization criterion. Hence, the selection of $K(n)$ at time nT should be based on the minimum mean square criterion and given by (6.6).

$$MSE_n = E[(\Theta_o(n) - \Theta_s)^2] \quad (6.6)$$

On the other hand, the optimization of K sequence for the whole period of preamble, P , should be a choice of a set of K values, $K(0), K(1), K(2), \dots, K(P)$ which satisfy the following minimum mean sum of squares criterion.

$$MSSE_P = E[(\sum_{i=0}^P (\Theta_o(i) - \Theta_s)^2)] \quad (6.7)$$

Here if the optimization is required for the whole period including the normal data decoding, the criterion for the optimal K sequence should be based on (6.8).

$$MSSE_{\infty} = E \left[\left(\sum_{i=0}^{\infty} (\Theta_o(i) - \Theta_s) \right)^2 \right] \quad (6.8)$$

Two questions arise from these criteria. First, is the optimal solution of K 's for the finite period of P same as that of K 's found from $P \rightarrow \infty$? i.e. Is the solution independent of P ? The second question is, are the solutions found from (6.6) the same as the K values in the optimal set found from (6.7) or (6.8)? In this chapter, two optimal solutions of K 's are calculated based on (6.6), (6.7) and two questions are answered by investigating the two results. As a result, it is discovered that only one optimal set of K 's exist which satisfies the three criteria.

6.3 Optimization based on Minimum Mean Square Criterion

From (6.5), the difference between output phase at time nT and the real input phase is expressed as (6.9).

$$\Theta_o(n+1) - \Theta_s = (1 - K(n))(\Theta_o(n) - \Theta_s) + K(n)\Theta_N(n) \quad (6.9)$$

Let, prediction error,

$$\Theta_d(n) = \Theta_o(n) - \Theta_s \quad (6.10)$$

Then, (6.9) becomes (6.11).

$$\Theta_d(n+1) = (1 - K(n))\Theta_d(n) + K(n)\Theta_N(n) \quad (6.11)$$

Since, initially $\Theta_o(0) = 0$, the first output phase $\Theta_o(1)$ becomes (6.12).

$$\Theta_o(1) = K(0)(\Theta_s + \Theta_N(0)) \quad (6.12)$$

Hence, the first phase difference is,

$$\Theta_d(1) = -(1 - K(0))\Theta_s + K(0)\Theta_N(0) \quad (6.13)$$

Define,

$$\sigma^2(n) = E[\Theta_d^2(n)] \quad (6.14)$$

Then minimization of (6.14) becomes the minimization of (6.6). But,

$$E[\Theta_d^2(n+1)] = E[((1-K(n))\Theta_d(n) + K(n)\Theta_N(n))^2] \quad (6.15)$$

Then, (6.15) becomes (6.16).

$$E[\Theta_d^2(n+1)] = E[(1-K(n))^2\Theta_d^2(n) + K^2(n)\Theta_N^2(n) + 2(1-K(n))K(n)\Theta_d(n)\Theta_N(n)] \quad (6.16)$$

Since $\Theta_d(n) = \Theta_o(n) - \Theta_s$, and $\Theta_o(n)$ is a function of $\Theta_s, K(0), \dots, K(n-1)$, and $\Theta_N(0), \dots, \Theta_N(n-1)$, $E[\Theta_d(n)\Theta_N(n)] = 0$, where $E[\Theta_N(k)\Theta_N(n)] = 0$ if $k \neq n$.

Therefore, (6.16) gives (6.17).

$$\sigma^2(n+1) = (1-K(n))^2\sigma^2(n) + K^2(n)\sigma_N^2 \quad (6.17)$$

Also, from (6.13),

$$\sigma^2(1) = (1-K(0))^2\Theta_s^2 + K^2(0)\sigma_N^2 \quad (6.18)$$

From (6.17) and (6.18), it is seen that minimization problem of $\sigma^2(n+1)$ for $K(0)$ is minimization problem of $\sigma^2(n)$, and it is again the minimization problem of $\sigma^2(n-1)$, and finally the minimization problem of $\sigma^2(1)$. Hence, first minimization of $\sigma^2(1)$ is necessary. Since $\sigma^2(1)$ is a convex function of $K(0)$, the minimum value of it is found at $\frac{\partial \sigma^2(1)}{\partial K(0)} = 0$. Therefore,

$$\frac{\partial \sigma^2(1)}{\partial K(0)} = -2(1-K(0))\Theta_s^2 + 2K(0)\sigma_N^2 = 0 \quad (6.19)$$

Then, the value $K(0)$ which satisfies (6.19) is given by (6.20)

$$K(0) = \frac{\Theta_s^2}{\Theta_s^2 + \sigma_N^2} \quad (6.20)$$

Also, the minimization of $\sigma^2(2)$ gives (6.21).

$$K(1) = \frac{\Theta_s^2}{2\Theta_s^2 + \sigma_N^2} \quad (6.21)$$

In general, $K(n)$ is represented by (6.22).

$$K(n) = \frac{\Theta_s^2}{(n+1)\Theta_s^2 + \sigma_N^2} \quad (6.22)$$

The minimum value of $\sigma^2(n+1)$ using optimal K values is given by (6.23).

$$\sigma_{\min}^2(n+1) = \frac{\Theta_s^2 \sigma_N^2}{(n+1)\Theta_s^2 + \sigma_N^2} \quad (6.23)$$

As n goes large number,

$$K(n) \approx \frac{1}{n+1} \quad (6.24)$$

and,

$$\sigma_{\min}^2(n+1) \approx \frac{\sigma_N^2}{n+1} \quad (6.25)$$

In the case that $\Theta_s^2 \gg \sigma_N^2$,

$$K(n) = \frac{1}{n+1} \quad (6.26)$$

and,

$$\sigma_{\min}(n+1) = \frac{\sigma_N}{n+1} \quad (6.27)$$

6.4 Optimization Based on Minimum Mean Sum of Squares Criterion

To find the optimum solution which satisfies the criterion (6.6), (6.28) should be minimized.

$$MSSE_N = \sum_{i=1}^N \sigma^2(i) \quad (6.28)$$

To find the $K(0)$ which minimizes equation (6.28), (6.28) is differentiated.

$$\frac{\partial}{\partial K(0)} \sum_{i=1}^N \sigma^2(i) = \sum_{i=1}^N \frac{\partial \sigma^2(i)}{\partial K(0)} = \left(1 + \sum_{l=2}^i (1-K(l-1))\right)^2 \frac{\partial \sigma^2(1)}{\partial K(0)} \quad (6.29)$$

Since, $\frac{\partial}{\partial K(0)} \sum_{i=1}^N \sigma^2(i) = 0$ means $\frac{\partial \sigma^2(1)}{\partial K(0)} = 0$, the optimal value which minimizes

(6.28) is the optimal value which minimizes (6.18) and the result is given by (6.20).

In general,

$$\frac{\partial}{\partial K(n)} \sum_{i=0}^N \sigma^2(i) = (1 + \sum_{i=n+2}^N \prod_{l=n+2}^i (1-K(l-1))^2) \frac{\partial \sigma^2(n+1)}{\partial K(n)} \quad (6.30)$$

(6.30) also gives the same conclusions that optimal K values based on MMSSE criterion is same as those based on MMSE criterion. Hence, again

$$K(n) = \frac{\Theta_s^2}{(n+1)\Theta_s^2 + \sigma_N^2} \quad (6.31)$$

$$\sigma^2(n+1) = \frac{\Theta_s^2 \sigma_N^2}{(n+1)\Theta_s^2 + \sigma_N^2} \quad (6.32)$$

Since the optimal K sequences that minimize the equation (6.28) are independent of the finite period N , the K sequences are again the optimal solutions which minimize the equation (6.8).

To verify the optimal K sequences, a simple simulation program was written using C programming language. Four sets of K sequences are tested for the real time phase acquisition and for the signal-to-noise power ration when the signal is normalized to "1" and jitter noise is assumed to have a gaussian distribution with the standard deviation of 0.15. Figure 6.2, and 6.3 show the real time phase acquisition up to 100 bit cycles in two different noise environments. Here, *in* indicates the input jitter noise. First, *out* indicates the phase acquisition when the optimal K sequences was used. Second, *out2* indicates the phase acquisition when 2^n approximation for the optimal K sequences is used. For example, $1/3, 1/4$ are approximated as $1/4$. Third, *out3* indicates the four step K sequence, which are $K = 1, 1/2, 1/2, 1/4, 1/4, 1/4, 1/4, 1/32, \dots$. Finally, *out4* indicates the fixed K sequence, where $K = 1/32$. The input statistics are shown in table 6.1.

Table 6.1 Input Jitter Noise Statistics

gau1.dat

Statistics: x

Col	N	Mean	SD	Min	25%	50%	75%	Max
1	100	-0.000676	0.1534	-0.3059	-0.1115	-0.01842	0.1254	0.3728

gau2.dat

Statistics: x

Col	N	Mean	SD	Min	25%	50%	75%	Max
1	100	0.01336	0.1656	-0.3378	-0.09133	0.006263	0.1222	0.5608

Figure 6.4 show the signal-to-noise power ratio. Here, the numbers also represent the K sequences indicated the above. Significant signal-to-noise power ratio improvements are obtained when gear-shifting methods are used.

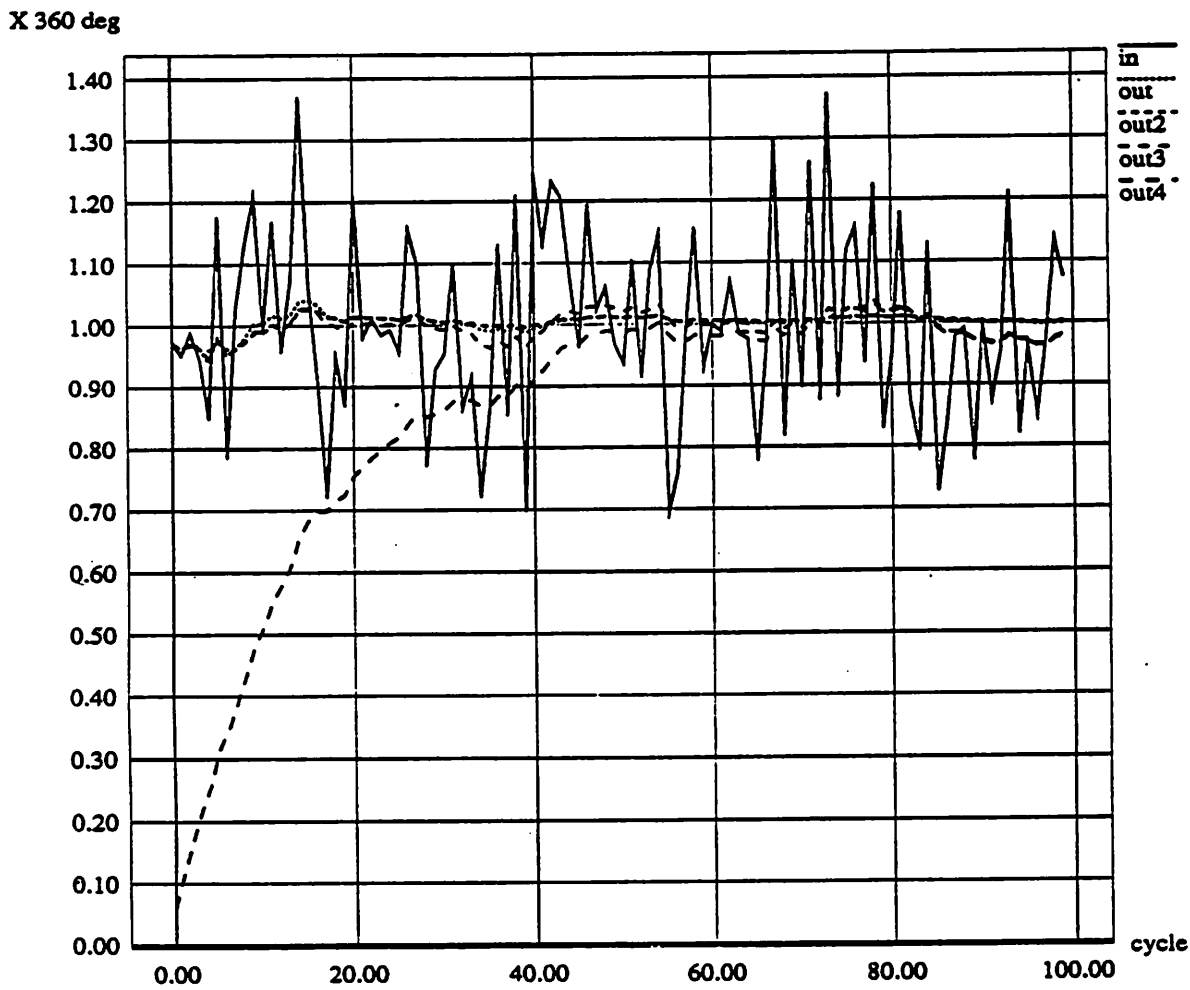


Figure 6.2 Real-Time Phase Acquisition Using Optimal K Sequences (I)

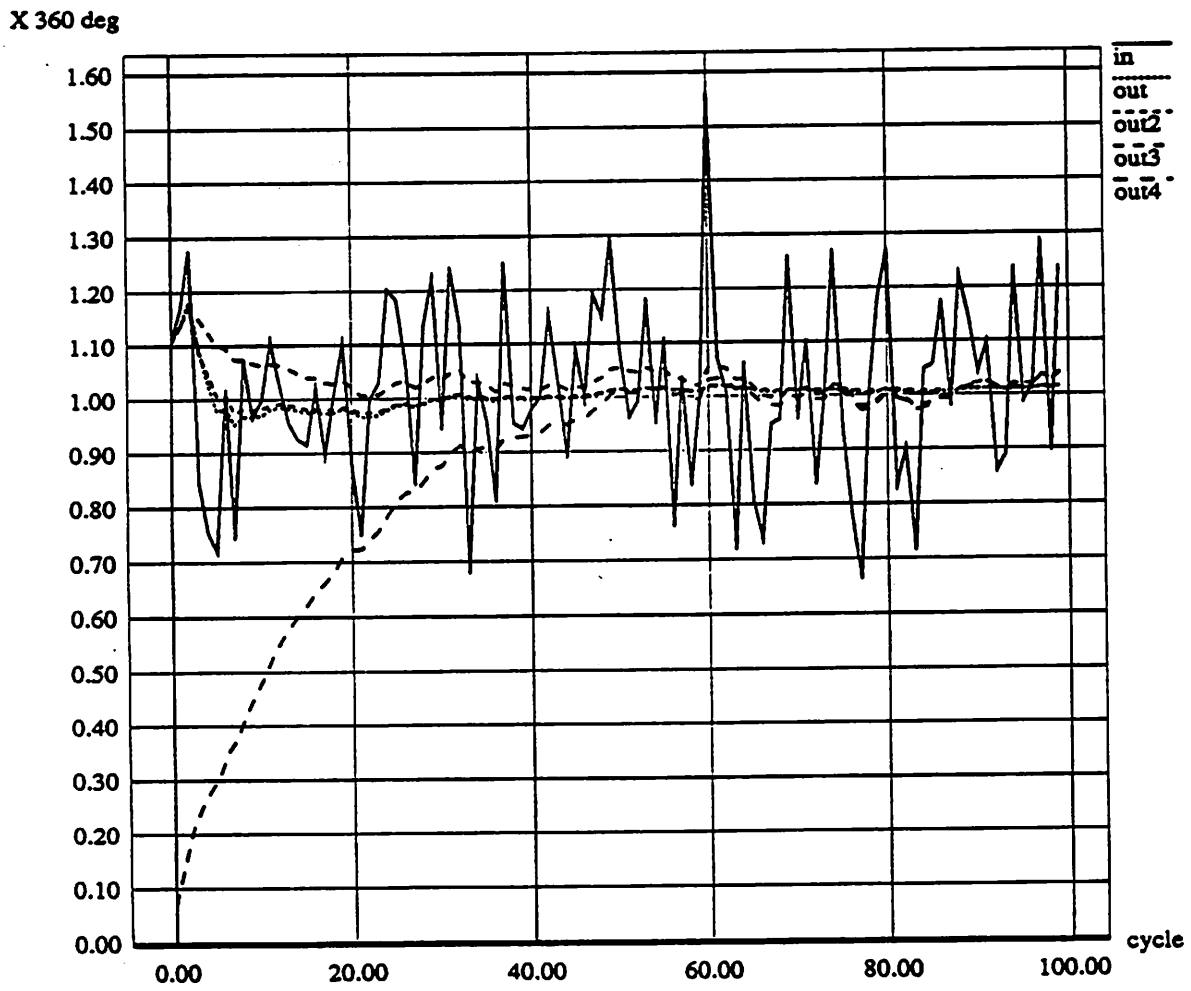


Figure 6.3 Real-Time Phase Acquisition Using Optimal K Sequences (II)

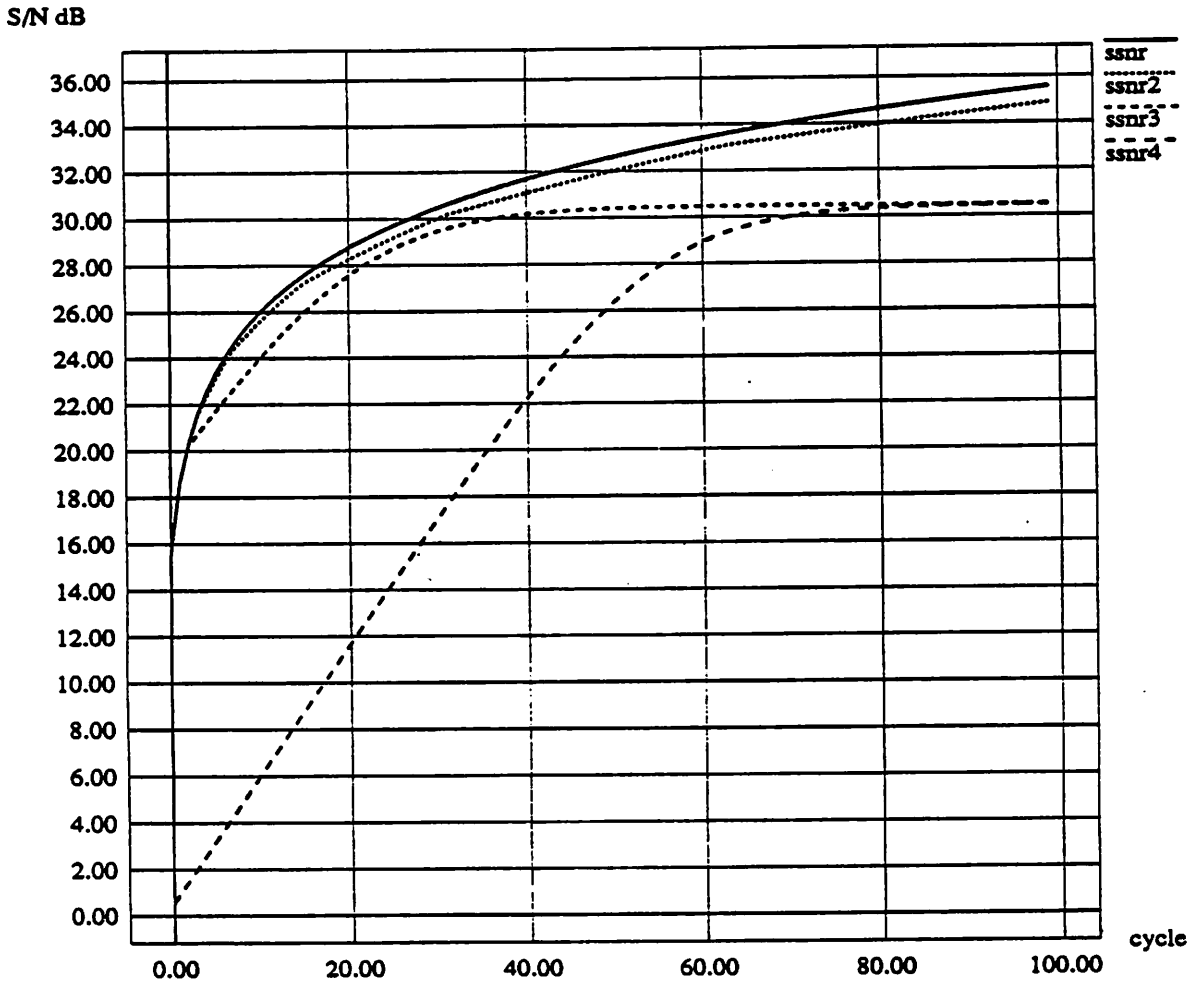


Figure 6.4 Signal-to-Noise Power Ratio Using Optimal K Sequences

Chapter 7

Experimental Results

7.1 Introduction

Two chips were fabricated to measure the performance of the hybrid clock recovery method. The first one contains an analog PLL including the 32 stage ring oscillator and parallel phase sampler. The second is the complete hybrid clock recovery circuits and contains a digital PLL including the previous analog PLL and parallel phase sampler. The chip is composed of a total of 20,000 transistors. Figure 7.1 shows the chip photo of the first one. The technology used here is a 2 μ m single poly, double metal, N-well CMOS process. These chips are fabricated through MOSIS.

7.2 Analog PLL and Parallel Phase Sampler

The ring oscillator in the analog PLL was tested first. The center frequency of the ring oscillator was about 30MHz and its control range is +/- 15%. Since the control range is one of the important factors in designing the VCO in order to be insensitive to the process parameter variations, a larger control range is preferred. In this design, the resistance of the PMOS load was changed to vary the frequency of the ring oscillator. However, if the cell bias current is controlled, the control range will be much larger than that of this design. In simulation, more than 80% control range is expected.

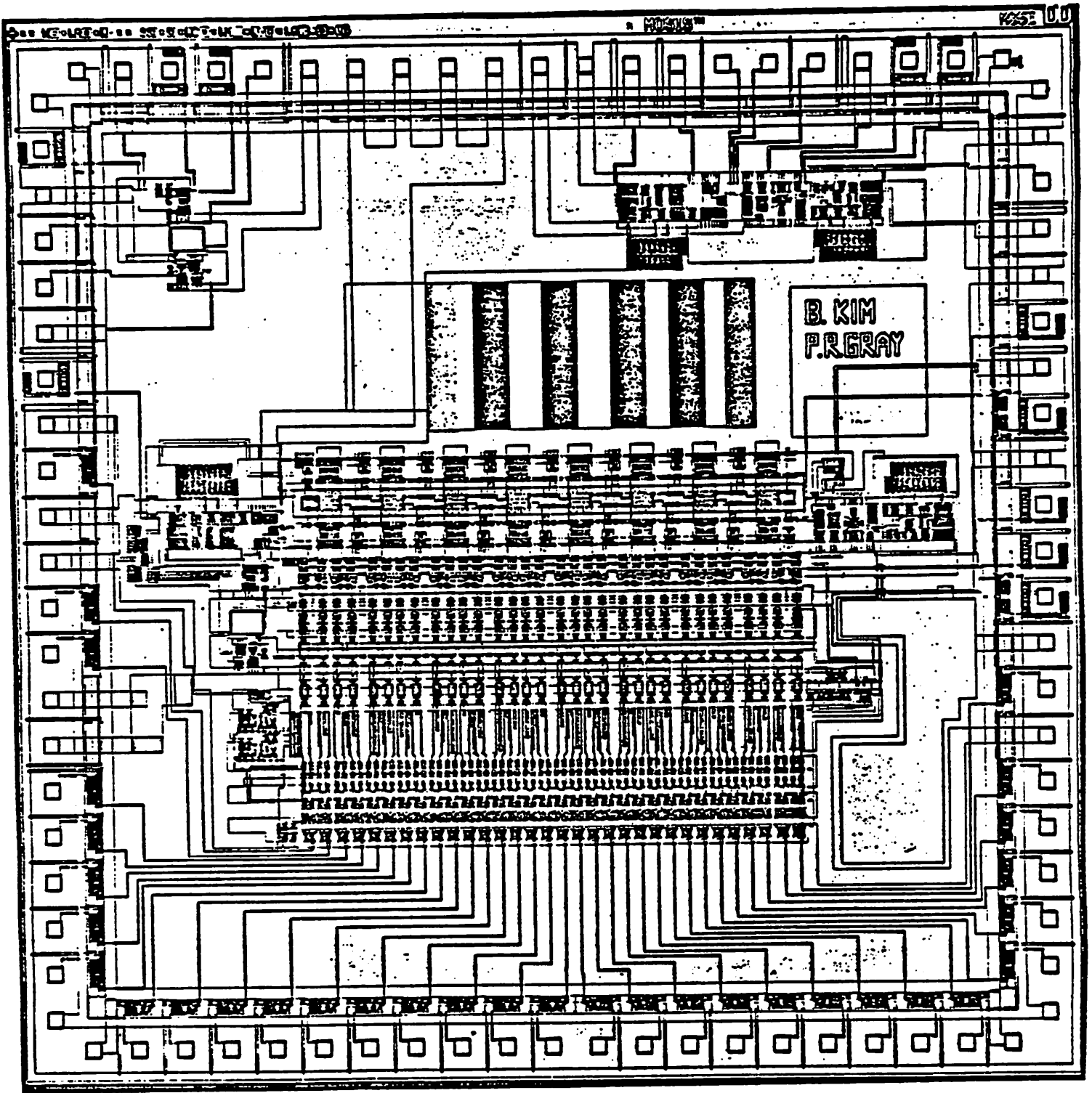


Figure 7.1 Chip Photo for the Analog PLL and Parallel Phase Sampler

The second measurement is the temperature coefficient (TC) of the ring oscillator free running frequency. Since, the chip can be operated at different ambient temperatures depending on the environment, a small TC is required so that PLL locking is maintained over a wide range of temperature variation. In the hybrid clock recovery circuit, since temperature stabilized biasing circuits for the voltage and current sources are embedded, the resulting TC is 62ppm/deg C. This value is very small and comparable to the value announced in [TP LIU].

Power supply sensitivity of the ring oscillator free running frequency is also important since power supply voltage can vary from place to place. In this case, 4.5%/volt power supply sensitivity was measured. This value is a little bigger than expected: the reason is that one of the voltage-to-current converters has a small common mode rejection ratio, and the control voltage changes according to the power supply variations.

The second measurement was performed on the analog PLL. Figure 7.2 shows the input clock waveform and the locked PLL output clock waveform. The waveform degradation comes from wire inductance and power supply noise.

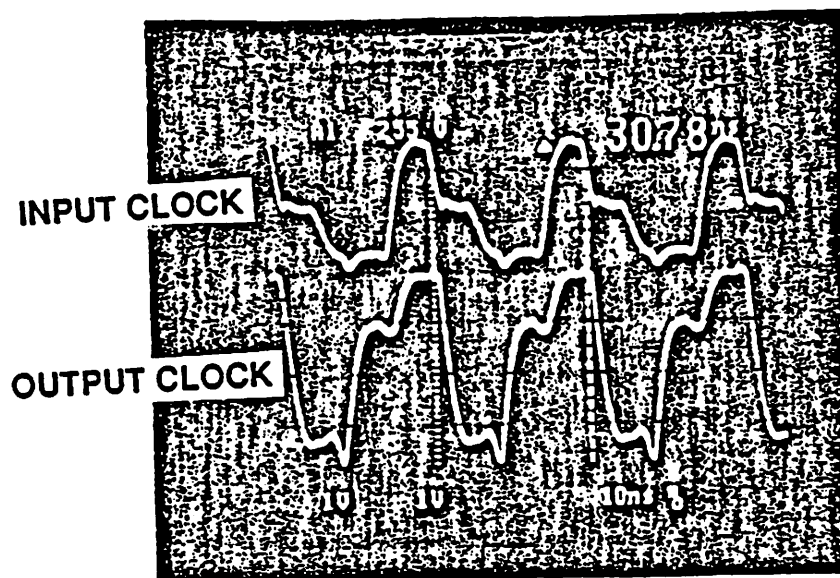


Figure 7.2 Input Clock and PLL Output Clock Waveforms

One of the key performance parameter is sampling jitter (this includes the PLL phase jitter). Figure 7.3 shows the experimentally measured sampling jitter. Here the analog PLL in the clock recovery circuit (DUT) is locked onto the reference clock (or system clock) running at 30MHz. A precision delay is used to sweep the data transition being sampled by the data latches in the clock recovery circuit. Here, at each sampling instant, the probability of "1" is measured for each delayed version of data transitions and a graph of probability "1" (a distribution function) as a function of the swept time delay is drawn. Here, the width of the transition from all "0" to all "1" corresponds to the 6 sigma's under the assumption that the distribution is gaussian. Therefore, 1 sigma is 100 ps and this indicates that the RMS jitter is 100 ps.

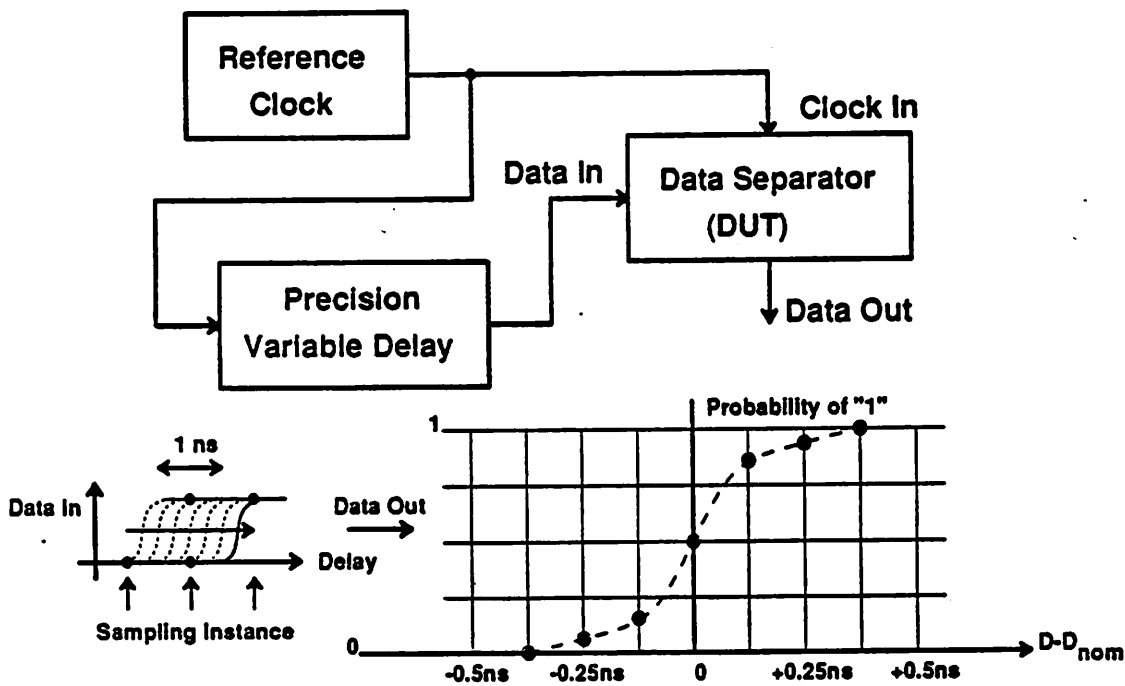


Figure 7.3 Sampling Jitter Measurement

Table 1 summarizes the experimental results from ring oscillator, analog PLL, and parallel phase sampler.

Table 1
Measured Data, 5V, 25°C unless indicated

Ring Oscillator and Phase Sampler	
Free Running Freq	30MHz
Std Deviation of Free-Run Freq, (2 runs, 20 devices)	0.84MHz (2.9%)
TC of Free-Running Freq	62ppm/deg C
Supply Sens of Free-Running Freq	4.5%/volt
Power Dissipation	390mW*
Silicon Area, 2 μ m	10K square mils
Silicon Area, 1 μ m **	4K square mils

*Idiosyncratic of particular level shift implementation- easily reduced to 300mW by straightforward modification of one level shifter in data path.

**Taken from 1 μ m prototype layout, now in fab.

7.3 Hybrid Analog/Digital Clock Recovery

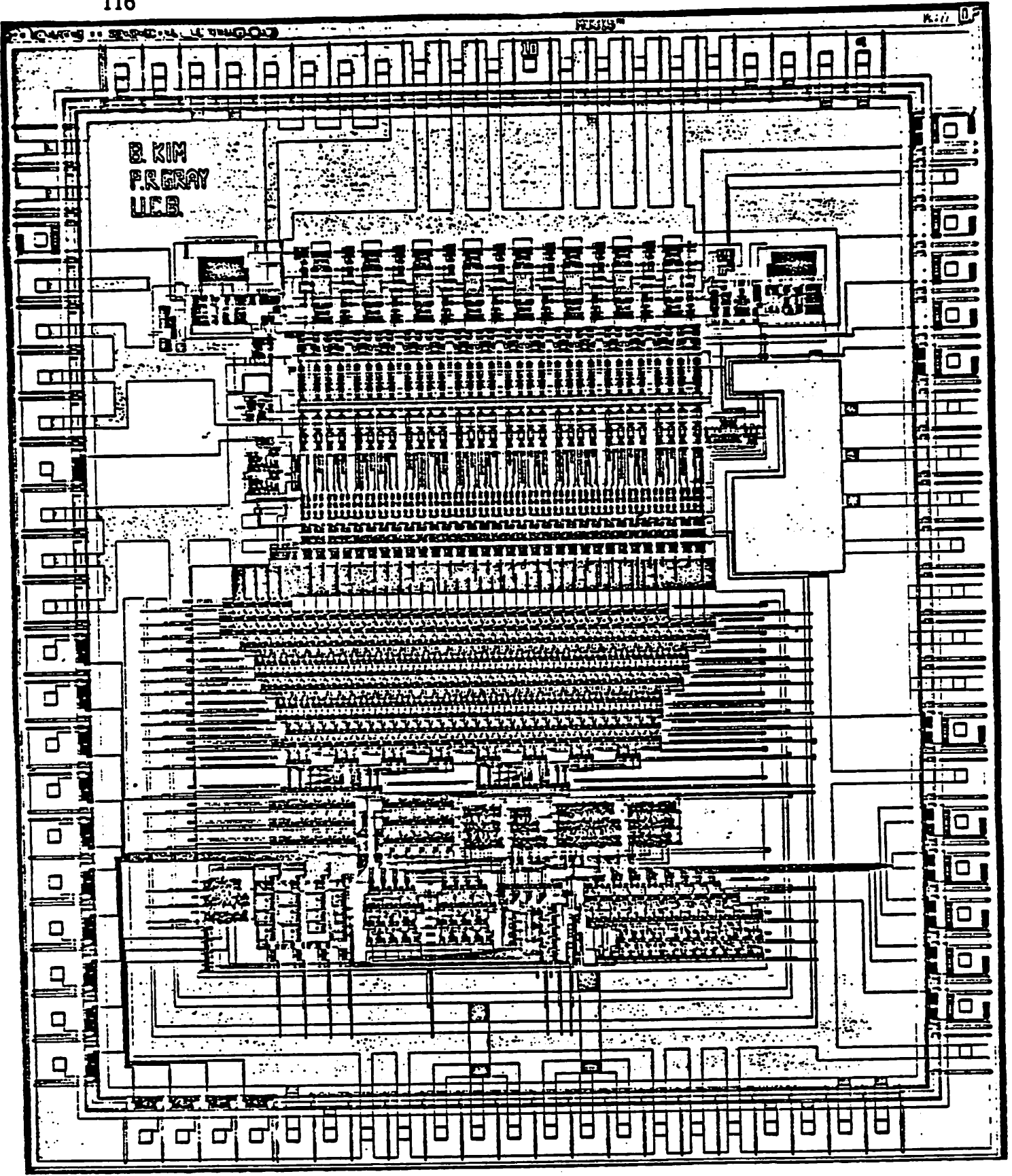


Figure 7.4 Chip Photo of Hybrid Clock Recovery Circuit

Figure 7.4 shows the chip photo of the complete clock recovery circuit. This chip includes an analog PLL, a parallel phase sampler and a digital PLL. Here, most aspects of the digital PLL are programmable, so the chip size is much larger than would be the case for PLLs dedicated to a specific purpose. The technology used here is the same process as for the previous chip.

One of the key performance issues of the clock recovery circuit is the jitter tolerance and represented by the maximum effective decode window. To find the window width, Bit error rate in the complete data detector was measured as a function of data transition location in the window. Figure 7.5 shows the measured bit error rate vs the pulse position in decode. Here the maximum effective decode window is the distance between the two crossing points where the bit error rate curve meets the 10^{-8} bit error rate line. The measured maximum effective decode window was 31 ns, or 94% of the full window width of 33ns. This is limited by a combination of the jitter mentioned above and the sampling resolution (phase quantization noise).

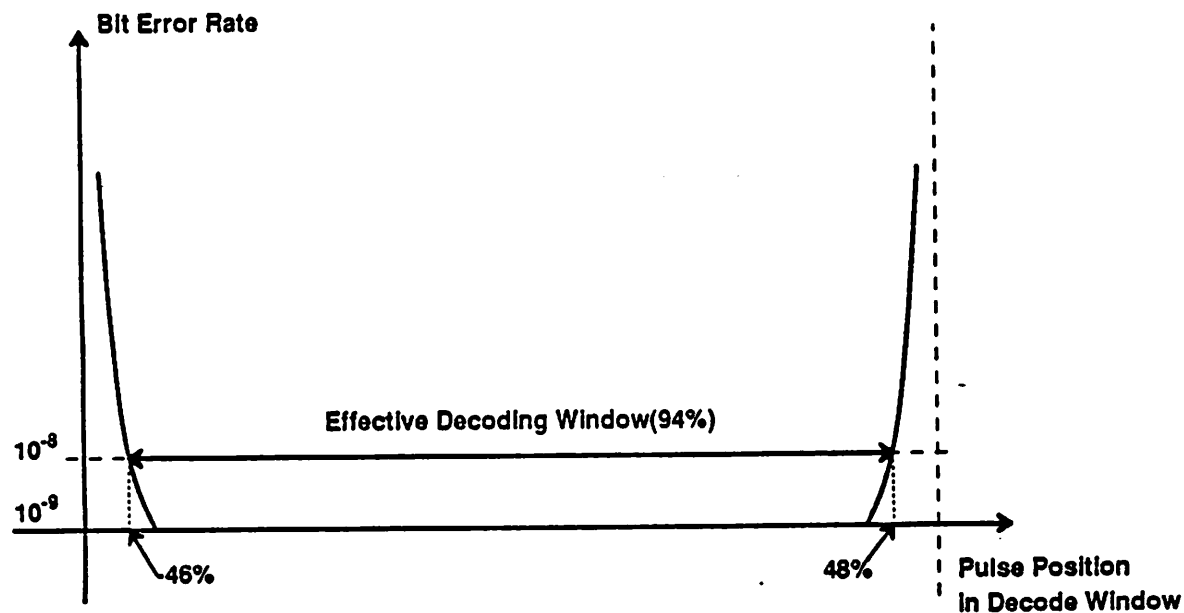


Figure 7.5 Decode Window Measurement

Because of the zero-phase start algorithm, phase acquisition is instantaneous, limited only by the jitter on the first detected transition. Initial acquisition is followed by a period of PLL operation with short time constant to allow decay of phase error due to jitter on the first transition. Then the loop time constant is shifted to a large value for normal operation. Figure 7.6 displays the typical acquisition sequence showing error free data recovery with no preamble. Here the incoming data and the local clock are at the same frequency but out of phase. The vertical lines are added to indicate the window boundaries. Each data transition contains all information concerning the data and clock. From this picture, for each input pulse, the corresponding correctly positioned synchronized output pulse is generated from the data separator chip. The first pulse is intentionally given 25 % jitter to inject error in the acquisition process.

The other parameters such as power dissipation, area etc. are summarized in table 2.

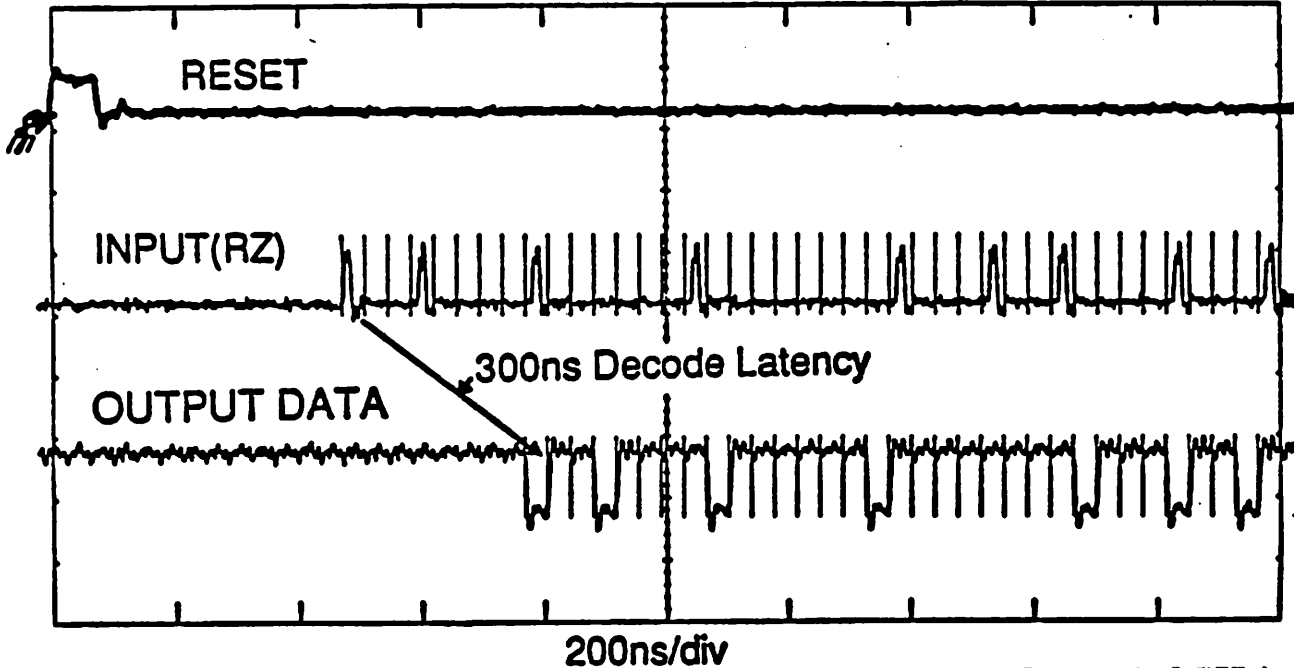


Figure 7.6 Typical Phase Acquisition Sequence

Table 2
Measured Data, 5V, 25°C unless indicated

Ring Oscillator and Phase Sampler	
Free Running Freq	30MHz
Std Deviation of Free-Run Freq, (2 runs, 20 devices)	0.84MHz (2.9%)
TC of Free-Running Freq	62ppm/deg C
Supply Sens of Free-Running Freq	4.5%/volt
Power Dissipation	390mW
Silicon Area, 2 μ m	10K square mils
Silicon Area, 1 μ m	4K square mils
Complete Analog/Digital PLL	
Silicon Area, 2 μ m	30K square mils
Power Dissipation	600mW

Chapter 8

Conclusions

8.1 Summary of Research Results

This research shows that the new architecture for clock recovery, called *hybrid analog/digital clock recovery* is of potential interest in high speed data transmission systems such as disk drive read-channels, local area networks, and optical transmission. The research results are summarized as follows.

1. Hybrid analog/digital clock recovery technique is viable approach for clock recovery in high speed data transmission systems.
2. In general, hybrid analog/digital clock recovery allows short preamble period (1-2 bit cycles).
3. In general, an effective phase resolution of 1 ns in a 2 μm CMOS process is achievable.
4. Jitter performance is improved using differential ring oscillator. Using large input devices in the delay cell, 100 ps RMS jitter is achieved.
5. A delay-locked loop further improves the jitter performance since it does not have the problem of jitter accumulation.
6. In hybrid clock recovery circuits, maximum effective decoding window width is virtually full since the decode window width depends only on sampling noise.
7. The hybrid architecture allows the use of CMOS technology for high speed clock recovery and incorporates the capability to implement sophisticated algorithms in the digital domain.

References

- [1] P. Gray, R. Meyer, "Analysis and Design of Analog Integrated Circuits", John Wiley & Sons, 1984.
- [2] D. Hodges, H. Jackson, "Analysis and Design of Digital Integrated Circuits", McGraw-Hill Book Company, 1983.
- [3] E. Lee, D. Messerschmitt, "Digital Communication", Kluwer Academic Publishers, 1988.
- [4] J. Kellis, S. Mehrotra, "A 15 Mb/s Data Separator and Write Compensation Circuit for Winchester Disk Drives", ISSCC, vol. 29, pp. 232-233, Feb. 1984.
- [5] W. Llewellyn, M. Wong, G. Tietz, and P. Tucci, "A 33 Mb/s Data Synchronizing Phase-Locked Loop Circuit", ISSCC, vol. 31, pp. 12-13, Feb. 1988.
- [6] A. Bell, G. Boriello "A Single Chip NMOS Ethernet Controller", ISSCC, vol. 26, pp. 70-71, Feb. 1983.
- [7] H. Haung, D. Banatao, G. Perlegos, T. Wu, T. Chiu, "A CMOS Ethernet Serial Interface Chip", ISSCC, vol. 27, pp. 184-185, Feb. 1984.
- [8] K. Ware, H. Lee, C. Sodini, "A 200 MHz CMOS PLL with Dual Phase Detectors", ISSCC, vol. 32, pp. 192-193, Feb. 1989.
- [9] US Patent 4,584,695, National Semiconductor, 1980.
- [10] US Patent 4,189,622, NCR Corporation, 1977.
- [11] J. Sonntag, R. Leonowich, "A Monolithic CMOS 10 MHz DPLL for Burst-Mode Data Retiming", ISSCC, vol. 33, pp. 194-195, Feb. 1990.
- [12] B. Kim, D. Helman, P. Gray, "A 30 MHz High-Speed Analog/Digital PLL in 2 μm CMOS", ISSCC, vol. 33, pp. 104-105, Feb. 1990.
- [13] A. Jordan, N. Jordan, "Theory of Noise in Metal Oxide Semiconductor Devices", IEEE Electron Devices, pp. 148-156, March 1965.

- [14] F. Gardner, "Charge-Pump Phase-Locked Loops", *IEEE Communications*, vol. com-28, no. 11, pp 1849-1858, Nov. 1980.
- [15] F. Gardner, "Phaselock Techniques", Wiley-Interscience Publication, 1979.
- [17] T. Liu, R. Meyer, "A 250 MHz Monolithic Voltage-Controlled Oscillator", *ISSCC*, vol. 31, pp. 22-23, Feb. 1988.
- [18] C. Tzeng, "Timing Recovery in Digital Subscriber Loops", *UCB/ERL M85/29*, April, 1985.
- [19] D. Messerschmitt, "Frequency Detectors for PLL Acquisition in Timing and Carrier Recovery", *IEEE Communications*, vol. com-27, no. 9, Sep. 1979.
- [20] C. Laber, C. Rahim, S. Dreyer, G. Uehara, P. Kwok, P. Gray, "Design Considerations for a High-Performance 3- μ m CMOS Analog Standard-Cell Library", *IEEE JSSC*, vol. sc-22, no. 2, April, 1987.
- [21] R. Leonowich, J. Steininger, "A 45-MHz CMOS Phase/Frequency-Locked Loop Timing Recovery Circuit", *ISSCC*, vol. 31, Feb. 1988.
- [22] K. Mueller, M. Muller, "Timing Recovery in Digital Synchronous Data Receivers", *IEEE Communications*, vol. com-24, no. 5, May 1976.
- [23] D. Jeong, G. Borriello, D. Hodges, R. Katz, "Design of PLL-Based Clock Generation Circuits", *IEEE JSSC*, vol. sc-22, no. 2, April 1987.
- [24] L. Bellato, G. Cariolaro, "Time Jitter in Line Regenerators with Pattern Dependent Pulse Waveforms", *Alta Frequenza*, vol. XLI-N.11, Nov. 1972.