A 2.5 Gb/s SONET Clock and Data Recovery Macro Cell

by

Andrew Holden Dickson

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Master of Science

and

Bachelor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1994

Author .....
Department of Electrical Engineering and Computer Science
May 18, 1994

Certified by .....
Hae-Seung Lee
Professor
Thesis Supervisor

Certified by .....
Douglas Hoy
Manager
Thesis Supervisor

Accepted by .....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

A 2.5 Gb/s SONET Clock and Data Recovery Macro Cell

by

Andrew Holden Dickson

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 1994, in partial fulfillment of the
requirements for the degrees of
Master of Science
and
Bachelor of Science

Abstract

This thesis covers the design and SPICE simulation of a 2.5 Gb/s clock and data recovery integrated circuit suitable for a SONET OC-48 based system. Critical circuits, such as the VCO and the phase-detector were analyzed and simulated with SPICE in detail. The design is evaluated for compliance with the SONET standard specifications of phase noise, jitter, etc. using a temperature range of 25 deg C to 125 deg C, a supply voltage of $-5.2V \pm 400mV$, and ± 3 sigma process limits (up/down model libraries). Extracted layout parasitic capacitors are included in the final simulations. Device models for SPICE are taken from Tektronix Component Corporation GST-1, a high speed self-aligned double-polysilicon bipolar process with an $f_t > 12$ GHz.

Thesis Supervisor: Hae-Seung Lee
Title: Professor

Thesis Supervisor: Douglas Hoy
Title: Manager

Acknowledgments

First of all I thank my parents who have been a blessing and a loving support to me throughout my entire education. Without their support and encouragement I would never have made it to or through MIT.

A large thank you to Doug Hoy, my manager at Tektronix, for giving me the opportunity to work on a project like this. His knowledge and expertise of digital and system-level design were necessary to help me complete this project.

Much thanks goes to Chris Martinez for his analog design help, and fellow group members Cary Renzema and George Caspell for their support and circuit ideas.

I am indebted to Jim Lamb and Art Metz who gave up many hours to discuss circuit ideas, oscillator noise, and phase-locked loops. Skip Hillman, Linley Gumm, Larry Lockwood, and Laszlo Dobos, and John McNeill also deserve thanks in this area.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The current generation and the next generation of fiber optic telecommunications is driving the clock and data speeds to multi-GHz rates. A set of standards, known as SONET (Synchronous Optical Network) or SDH (Synchronous Digital Hierarchy), has been developed in order to standardize global high-speed optical communication. The SONET OC-48 or SDH STM-16 standard is a multi-layer operations, administration, maintenance and provisioning standard, covering the path, section, line, and physical layers of a communication system. The physical layer of these standards consists of an optical 2.48832 Gb/s NRZ scrambled serial data stream.

Because of the high clock rate of OC-48/STM-16, more of the required functionality, such as byte alignment, frame detection and demultiplexing, are constrained to monolithic or hybrid package implementation. The move towards fully integrated clock and data recovery is also motivated by space requirements. Clock recovery at this speed requires high precision components in order to provide fast acquisition and at the same time, low phase noise. Many high frequency clock recovery units use Surface Acoustic Wave (SAW) filters which have very high Qs, but cannot be fully integrated using present technology. Complete OC-48/STM-16 rate Phase-Locked Loops (PLL) have be implemented on a single chip [2, 5], reducing cost, complexity, space, and power requirements. Few of these have been produced for the commercial market, and none have been further integrated to include significant digital processing other than data re-timing.

It is for these reasons, among others, that there is a need for a high-speed clock recovery IC with a higher level of integration. It is desirable that the data output of the IC be demultiplexed down to a low enough rate so that further processing can be accomplished with a lower-speed, lower-cost, highly integrated and less power-hungry CMOS IC. Power is also reduced due to lower-speed outputs, and the cost and sensitivity of the board level design is relaxed, such as routing and transmission line design.

This thesis covers the development of a OC-48/STM-16 rate clock and data recovery IC which meets these needs, providing integrated byte alignment, frame detection, and a 1:16 data demultiplexer. The macro cell is intended to be used on a larger chip that also contains a post-amplifier and all the necessary input and output (I/O) cells—the development of the post-amp and the I/O cells will not be covered in this thesis.

In order to attain this project goal, a high-performance bipolar process is used, along with accurate device modeling and simulation tools. The Tektronix GST-1 Si-bipolar process is ideal for this task, with an npn $f_T > 12$ GHz and excellent device modeling over a wide range of temperature and process variations. For the digital logic, the Tektronix GST-1 standard cell library provides the necessary resources for accurate modeling as well as functional testing. See Appenix A for additional information about the GST-1 process and modeling.

1.1 General Description

The IC that includes the macro cell covered in this thesis can be used in the receiver front-end of a system such as an add/drop multiplexer (shown in Figure 1-1), regenerator, cross-connect, line terminator, or a variety of test equipment. On the system level, the IC lies directly after the optical-to-electrical conversion and the pre-amplifier, and right before the descrambling, overhead processing, and further demultiplexing. The IC contains an on-chip frequency- and phase-locked loop (FPLL) to recover the clock and re-time the data from a 2.48832 Gb/s Non-Return-to-Zero

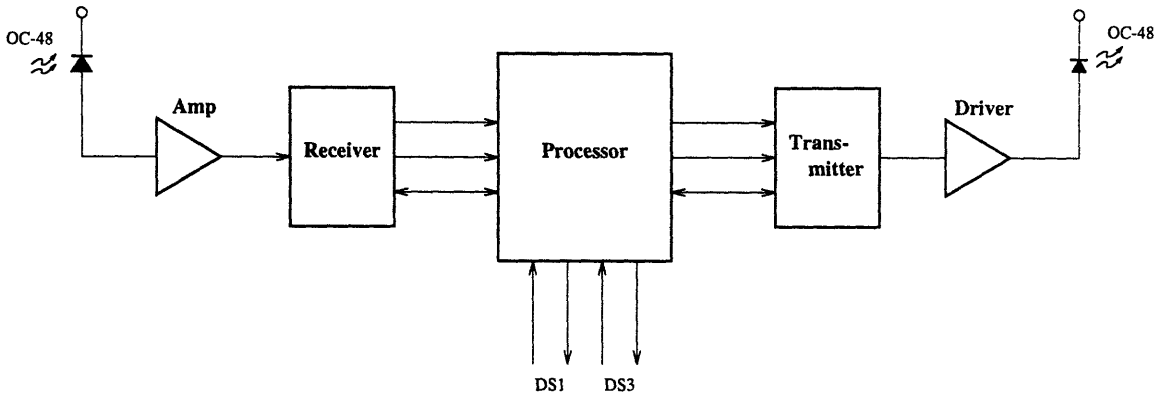


Figure 1-1: Block Diagram of an Add/Drop Multiplexer

(NRZ) serial data stream, and outputs the data in 16 parallel data lines each running at 155.52 Mb/s. It also provides lock detect, byte alignment, frame detection, and supplies a steady output clock if the data signal has been lost. The SONET/SDH framing bytes are used to provide byte alignment and frame detection.

A block diagram of the receiver is shown in Figure 1-2. As stated previously, the design of the post amp and I/O cells will not be covered in this thesis, but will be integrated onto the final IC design. There are three separate input paths to the chip to give the system designer flexibility in designing and testing the chip. The main data input to the chip under typical operating conditions is the Serial Data Input (SDI) line which is intended to be driven by the optical-to-electrical converter pre-amplifier. This line is fed directly into a post-amplifier (post-amp) which regenerates the single-ended, small-signal input to a differential, digital-level signal. If an external post-amp is used instead of the internal post-amp, the Bypass Serial Data Input (BPSDI) lines serve to bypass the internal post-amp when the Bypass Post-Amp Enable (BPPAE) line is asserted high. Also, a Loopback Serial Data Input (LBSDI) is provided to facilitate in-system diagnostic loopback testing. This input is enabled when the Loopback Enable (LBE) line is asserted high. Both the BPSDI and LBSDI inputs are differential digital ECL inputs.

If the signal power on SDI should fall below a specified threshold, the post amp asserts the Loss-Of-Power Indicator (LOPI) line, signaling that the data is most likely invalid. This may happen in the case of a break in the fiber optic cable or a

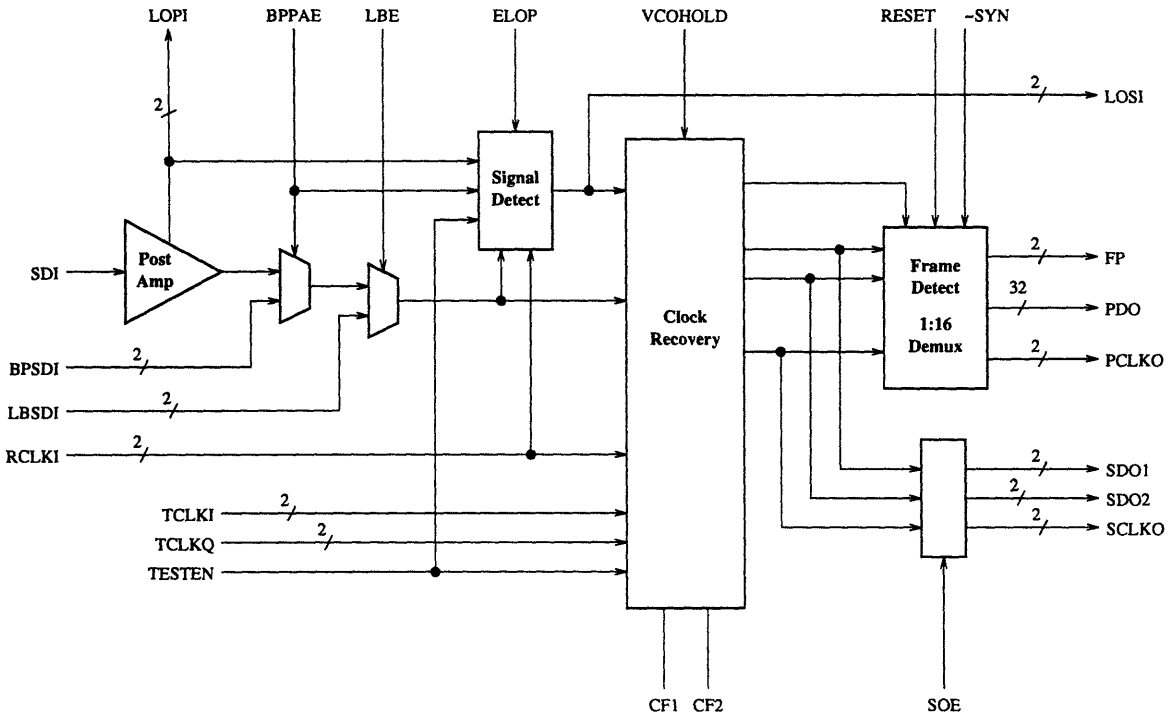


Figure 1-2: Receiver Block Diagram

transmitter failure—no valid data is present on the input lines, but noise inherent in the system may still cause transitions across the digital threshold levels. If the on-chip post-amp is bypassed (BPPAE high), the External Loss-Of-Power (ELOP) line must be attached to the off-chip post amp power sensing output.

The Signal Detect block is similar to the loss-of-power indicator in the post-amp in that it senses a Loss-Of-Signal (LOS) condition. The difference is that the Signal Detect block detects the loss of transitions in the data stream. This block is essentially a digital counter which is reset by any transition on the selected serial data input line. If no transitions occur within a certain number of bit periods, the Loss-Of-Signal Indicator (LOSI) line is asserted. If a loss-of-power condition exists (either LOPI or ELOP line asserted), the LOSI line is automatically asserted. When the Clock Recovery block is bypassed (see below) the Signal Detect block output is disabled.

The Clock Recovery block is a frequency- and phase-locked loop (FPLL) which extracts a synchronous clock signal from the selected serial data input. The FPLL consists of a voltage controlled oscillator (VCO), two phase detectors (PD), a frequency detector (FP), and a loop filter/amplifier with an off-chip capacitor. If a LOSI

condition should occur, the FPLL locks onto the reference clock input (RCLKI) to provide a steady output clock.

The Frame Detect/Demux block is a digital block which provides the necessary byte and frame alignment, as well as the 16-bit parallel output. The core of this block is a 24-bit shift register and pipelined decoding logic which searches the re-timed serial data for an A1A1A2 framing pattern (See Chapter 2). The Frame Pulse (FP) line is a synchronous periodic signal which indicates that a valid A1A1A2 framing pattern has been found and that the demultiplexer is in proper byte alignment. The FP line is asserted once per frame, one byte after an A1A1A2 sequence is found, and it is asserted for one two-byte period. The SYN line is a single-ended, active falling edge signal which allows the Frame Detect block to re-align with the incoming data. The demultiplexer uses a resettable clock divider to divide the extracted clock signal down and latch 16 bits of data from the frame detect shift register with the proper byte alignment. The 16 parallel data lines (PDO0-PDO15) from the Demux block are at a 155.52 Mb/s data rate, and the associated clock (PCLKO) is a half-speed clock (77.76 MHz) which can be used to latch the data on each edge. The TESTEN line allows full functional testing of the Frame Detect/Demux block, even at full-speed if necessary. For this test, TESTEN is asserted high, the data signal should be applied to the RCLKI lines and the associated clock signal should be applied to the TCLKI lines.

After lock has been acheived and an A1A1A2 sequence has been detected, the receiver will remain in byte alignment until a LOSI condition exists or the SYN line is asserted. In the case of an LOSI condition, the Clock Recovery and Frame Detect blocks are reset and the Demux output is disabled. If SYN is asserted, the Frame Detect block is reset and the Demux output is disabled, but the Clock recovery block will remain locked to the serial data stream.

The re-timed serial data and its associated clock are available as outputs when the Serial Output Enable (SOE) line is asserted. The data is brought out on two separate 1.24416 Gb/s serial streams (SDO1 and SDO2). These data lines are clocked on each edge of the Serial Clock Output (SCLKO) line, the associated 1.24416 GHz clock.

These lines are intended to be connected to the transmitter to allow in-system line loopback testing, but under normal operating conditions the outputs are squelched to reduce on-chip noise.

Chapter 2 is an overview of the SONET/SDH basics, including a review of the necessary specifications for the receiver. Chapter 3 covers the basics of phase-locked loops (PLL) and some relevant performance issues. Chapter 4 begins the design of the receiver, containing the calculations for the necessary PLL parameters to achieve the specifications. Chapters 5, 6 and 7 cover the design of the clock recovery, signal detect, and frame detect/demux circuits respectively.

Chapter 2

The SONET OC-48/SDH STM-16 Standard

Different countries around the world have their own telephone standards which are not directy compatible with each other. Even within countries such as the United States there are various vendors with incompatible architectures, equipment, codes, etc. that make connection between carriers difficult. These and other problems are major obstacles between connecting systems from around the world together in a unified, global communication network.

The SONET concept was first proposed by Bellcore, recognized by the American National Standards Institute (ANSI), and developed by the Exchange Carriers Standards Association (ECSA) and the International Telegraph and Telephone Consultative Committee (CCITT) in an effort to unify global telecommunications. SONET/SDH defines a set of standard interfaces, signal rates, synchronous multiplexing formats, etc. The standards are designed to create a flexible, long-term "future-proof" solution that allows multi-vendor networking, transmission of a wide variety of types and rates of data, and a host of other benefits.[1, 8, 9]

Electrical Signal	Optical Signal	Data Rate (Mb/s)	CCITT Designation
STS-1	OC-1	51.84	STM-0
STS-3	OC-3	155.52	STM-1
STS-9	OC-9	466.56	
STS-12	OC-12	622.08	STM-4
STS-18	OC-18	933.12	
STS-24	OC-24	1244.16	
STS-36	OC-36	1866.24	
STS-48	OC-48	2488.32	STM-16

Table 2.1: SONET/SDH Designated Signal Rates

2.1 SONET/SDH Basics

2.1.1 Signal Rates

The basic SONET signal is called the Synchronous Transport Signal level-1, or STS-1, which is a 51.84 Mb/s serial data stream—the optical equivalent signal is known as Optical Channel or Optical Carrier level-1, abbreviated as OC-1. The CCITT standards designate signal rates in multiples of the Synchronous Transport Module-1, or STM-1, which is equivalent to three times the STS-1 rate. The SONET/SDH signals can be time-division multiplexed in specific byte-interleaved integer multiples of the basic rates. See Table 2.1 for a listing of the rate designations. A multiplexed N-integer multiple of a STS-1, OC-1, or STM-1 rate is generally referred to as STS-N, OC-N, or STM-N rate respectively.

2.1.2 Frames

Information is transmitted over the physical layer in "frames" of 9-byte row by 90-byte column matrices which are made up of the "payload" information and overhead bytes. The SONET STS-1 frame is shown in Figure 2-1. The bytes in each frame are transmitted row-by-row from top to bottom and from left to right. The first 3 columns of an STS-1 frame are Transport Overhead (TOH) bytes which contains

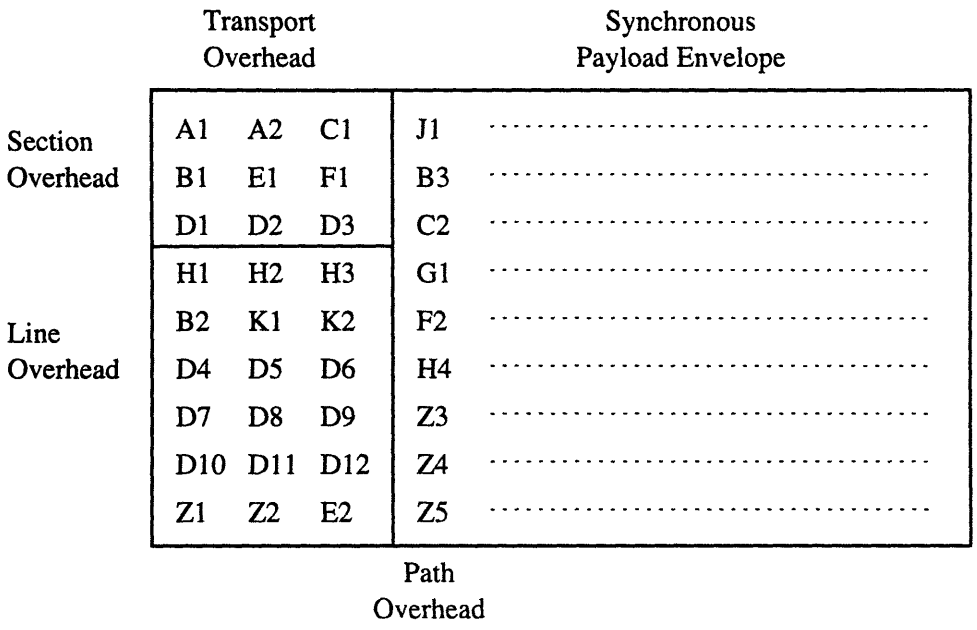


Figure 2-1: SONET STS-1 Frame

framing bytes and administrative information. The other 87 columns in the frame make up the Synchronous Payload Envelope (SPE). The SPE contains the main digital information such as telecommunication or video data, but the first column of the SPE is reserved for the path overhead. The SPE can take on many different formats depending on the type of data that it carries. Many types of data can be accepted—voice data such as telephone conversations, video data such as NTSC-quality or compressed HDTV, and other types of digital information such as computer data. Standard signals which can be transported include DS1, DS1C, DS2, DS3, CEPT-1, CEPT-4 and other types of Broadband ISDN, all of which can be packed into STS-1 or special STS-3c concatenated frames.

Every byte in the overhead section is labelled and has a certain function, some of which are reserved for future designation. The first three rows of the TOH is called the section overhead. It is used for frame synchronization, payload numbering, parity error monitoring, and message signals between equipment. The remaining six rows of the TOH is called the line overhead which is used for pointers, alarm indications, as well as parity error monitoring and message signals between equipment.

Frames can be multiplexed together to form higher-rate formats using a byte-

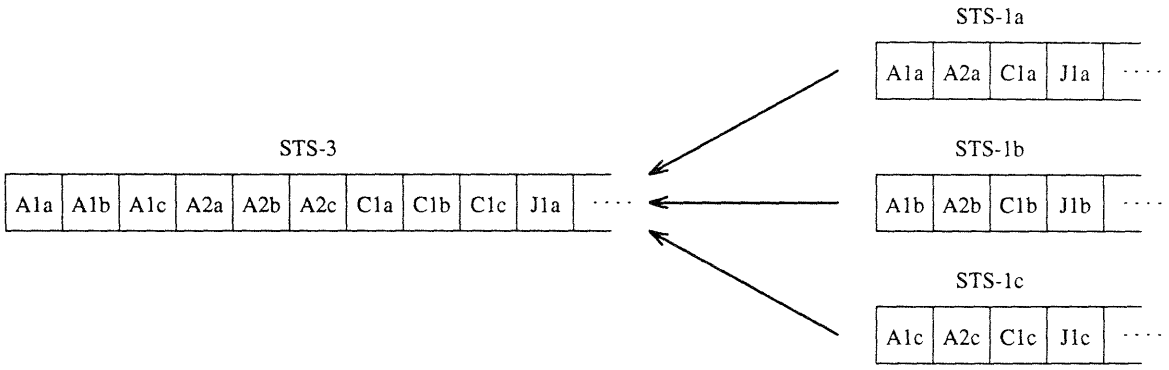


Figure 2-2: Time-Division Multiplexing of SONET frames

interleaved Time-Division Multiplexed (TDM) scheme. Figure 2-2 shows an example of this interleaving scheme for three input STS-1 streams.

Because of this multiplexing scheme, the first 2N bytes in any STS-N frame are always the A1 and A2 framing bytes—NxA1 bytes followed by NxA2 bytes. These bytes are used to synchronize the receiver to the incoming byte and frame boundaries. The A1, A2 and C1 (payload number) bytes are the only bytes that are not scrambled for transmission across the physical layer. A common technique is to use a frame detector which is designed to scan the serial data for the boundary between the A1 and A2 bytes.

Each STS-N/STM-N frame is a fixed 125μs in length. This frame time does not vary with the multiple number N. Take for example three STS-1 signals multiplexed up to an STS-3 rate—the frame contains three times more data but the signal rate is increased three fold, which gives the same frame time and effective throughput for each of the three STS-1 signals.

There are two specifications in the SONET/SDH standards which constrain the design and must be kept in mind, jitter tolerance and the pattern dependence test. Another specification, jitter transfer is not required because of the application, but we will see that it is useful in the design analysis. Appendix B has listing of these specifications.

Chapter 3

Phase-Locked Loop Basics

3.1 Definitions

Jitter can be thought of as the random phase variation on a waveform. In the context of a digital clock or data pattern, it describes the movement of an edge from its ideal location in time. This can be due to various sources of noise. Jitter is sometimes defined separately from *wander*, which can be thought of as low-frequency jitter. It is usually desirable to track wander due to its large, slow phase variations.

Jitter tolerance, in the context of a phase-locked loop (PLL), describes the maximum amount of input jitter that the PLL can track without losing lock. This is usually a specifications which states the minimum amount of phase jitter that a PLL should be able to track at various frequencies.

Jitter transfer is used to describe the transfer characteristic of phase input to phase output in a PLL. The maximum amount of jitter transferred from input to output is usually specified in order to reduce the total jitter in a system.

Jitter peaking occurs when there is a peak in the jitter transfer function. Excessive jitter peaking can cause large jitter build-up in cascaded systems. Because of this, jitter peaking is tightly controlled in the SONET/SDH specifications.

Pattern dependance describes the situation in which jitter is effectively generated in a PLL due to various input data patterns. For example, changes in the transition density of a data stream can affect the phase error of a PLL.

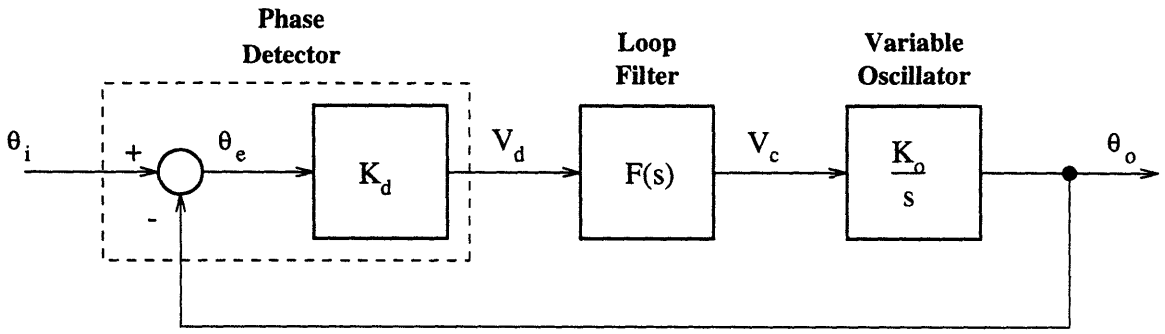


Figure 3-1: Linear phase-locked loop model

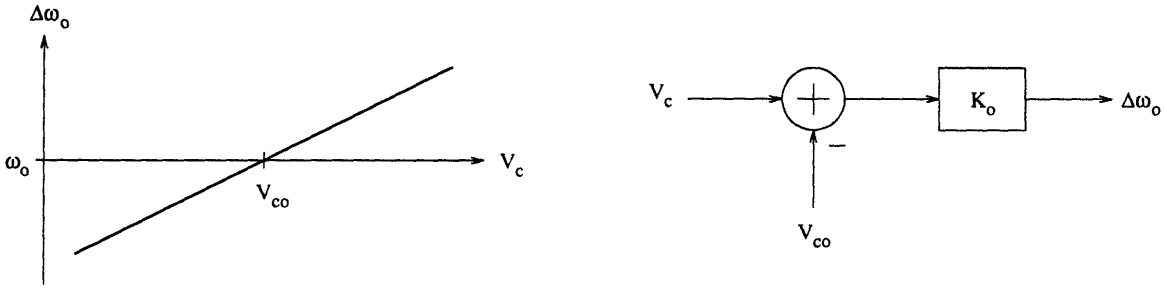


Figure 3-2: Ideal VCO model and characteristic

3.2 Phase-Locked Loops

A phase-locked loop is a system which aligns the phase of an input waveform to the phase of an internally generated reference waveform. The internal waveform generator is usually implemented as a voltage-controlled oscillator (VCO). To begin with, we will analyze the PLL in it's linear region, when it is frequency-locked to the input waveform.

Most PLLs usually consist of a phase detector (PD), a loop filter, and a VCO. See Figure 3-1 for a block diagram of a PLL. The VCO produces an output frequency which is proportional to the input voltage. Figure 3-2 shows a linear VCO characteristic and a functional model of a VCO. It is useful to define the VCO characteristic using a "constant" term ω_o and a deviation $\Delta\omega_o$. The frequency ω_o is defined as the average output frequency when the PLL is in lock, when the input frequency equals the output frequency, or $\omega_i = \omega_o$. The frequency deviation is

$$\Delta\omega_o = K_o(V_c - V_{co}) \tag{3.1}$$

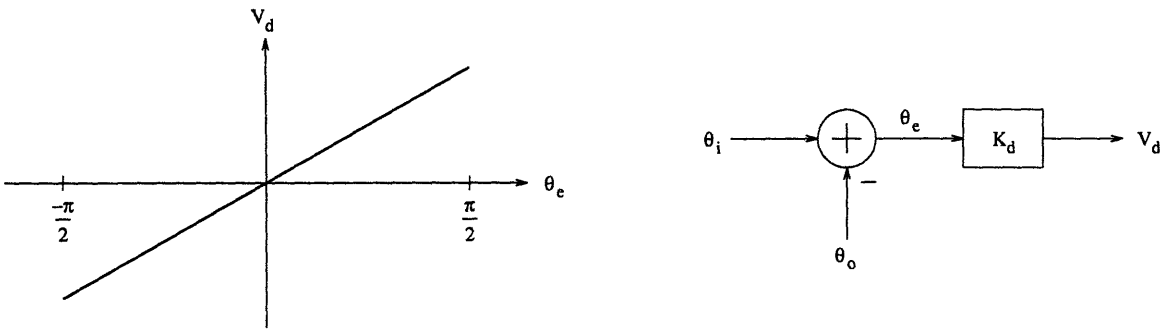


Figure 3-3: Ideal phase detector model and characteristic

where K_o is the VCO "gain" with units of radians per second per volt, V_c is the control voltage to the VCO and V_{co} is the control voltage which maintains the output ω_o.

Since we are interested in the phases and phase differences of various signals within the PLL, the transfer function of the VCO must be defined in terms of phase. Frequency is the derivative of phase, or stated differently, phase is the integral of frequency. Using the notation here

$$\theta_o = K_o \int_0^T \Delta\omega_o \, dt \tag{3.2}$$

Any deviation in frequency, which is defined as $\Delta\omega_o = \omega_o - \omega_i$ is integrated over time and scaled by K_o. Thus, the output phase of the VCO is

$$\theta_o = \frac{V_c K_o}{s} \tag{3.3}$$

From this equation we see that the VCO inherently adds one pole to the system.

The phase detector produces an output voltage V_d proportional to the difference in phase between its inputs. Figure 3-3 shows a linear phase detector characteristic and a functional model of a phase detector. The phase detector output voltage is

$$V_d = K_d(\theta_i - \theta_o) \tag{3.4}$$

where K_d is the phase detector gain with units of volts per radian and θ_i is the phase

of the input waveform.

The final block in the PLL is the loop filter. The filter output voltage is

$$V_c = F(s)V_d \tag{3.5}$$

where $F(s)$ typically has a low-pass characteristic. The output of the phase detector is filtered and used to tune the VCO.

A simple example illustrates the operation of the PLL. Assume that the PLL is perfectly locked to the incoming waveform so that $\omega_o = \omega_i$. If the frequency of the input waveform rises slightly, a phase error will begin to build up between the input and VCO waveforms. This phase error produces a voltage which in turn tunes the VCO to a higher frequency, eventually matching the input frequency. Similarly, if there is a phase change on the input waveform, the phase detector produces an error voltage which pushes the VCO waveform in the direction of the phase change, cancelling the phase error.

Continuing with the PLL analysis, we combining Equations 3.3, 3.4 and 3.5 and solve for the closed-loop phase transfer function $H(s)$

$$H(s) = \frac{\theta_o}{\theta_i} = \frac{K_o K_d F(s)}{s + K_o K_d F(s)} \tag{3.6}$$

It is also useful to solve for the closed-loop error transfer function $H_e(s)$ and the open-loop gain of the PLL, $G(s)$

$$H_e(s) = \frac{\theta_e}{\theta_i} = \frac{s}{s + K_o K_d F(s)} \tag{3.7}$$

$$G(s) = \frac{K_d K_o F(s)}{s} \tag{3.8}$$

All three of these functions are used to study the behavior of the PLL. There are three factors that define the PLL system: the VCO gain, the phase detector gain, and the filter transfer function. The VCO and phase detector gains can only be scaled, so the filter transfer function is used to alter the system behavior, which is analyzed next.

3.3 Loop Filter Selection

The loop filter is a very important piece of the PLL system. The type of filter used in a PLL dictates the overall PLL behavior, and the choice of filter time constants affects the performance.

As previouly stated, the VCO contributes a single pole to the system so that in the simplest PLL implementation in which no filter used, the PLL is a first-order system. The system under this condition can be analyzed using an equivalent filter transfer function $F(s) = 1$. Plugging this into Equation 3.6 gives the phase transfer function for the system

$$H(s) = \frac{K_o K_d}{s + K_o K_d} \tag{3.9}$$

This shows that effectively there is only one variable that the system designer can specify, $K = K_o K_d$. In this system the loop gain and bandwidth cannot be independently specified and thus there is a direct trade-off between specifications such as tracking performance and noise tolerance/rejection. To avoid this large constraint, first-order filters are typically used.

When a first-order loop filter is used, the PLL becomes a second-order system and thus there are two independent variables in the system—the designer can independently specify the bandwidth and loop gain. It is possible to use a second-order or higher-order loop filters, but this usually is undesired. One reason is that the second-order system is well understood and studied, and thus easier to implement. Also, a second-order system usually has enough flexibility with two parameters to satisfy design specifications. Third-order systems are more complex and can can have more complex stability issues that need to be addressed. A first-order loop filter is the most commonly used PLL filter and will be used in this design.

There are two types of second-order filters that can be used, either passive or active, each of which results in a different filter characteristic and PLL performance. Figure 3-4 shows an ideal model for both a passive lag filter and an active Proportional plus Integral (PI) filter, along with their respective transfer functions. The general

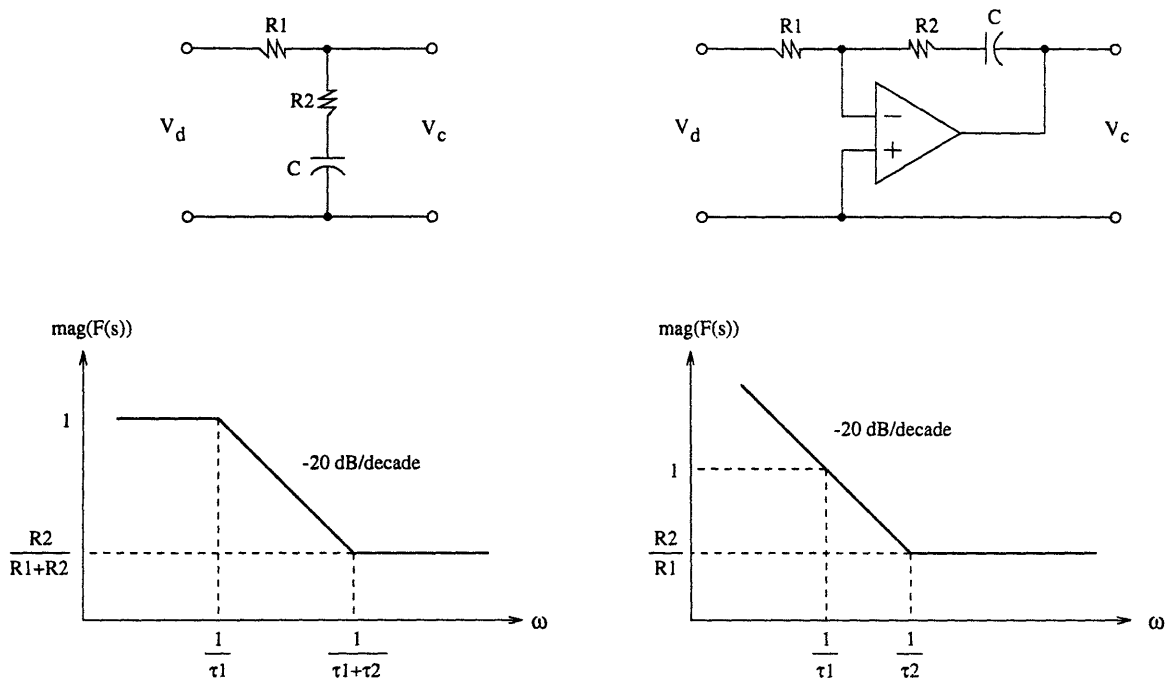


Figure 3-4: Passive and active filter models

passive and active filter transfer functions are, respectively

$$F_p(s) = \frac{s\tau_2 + 1}{s(\tau_1 + \tau_2) + 1} \tag{3.10}$$

$$F_a(s) = \frac{s\tau_2 + 1}{s\tau_1} \tag{3.11}$$

where $\tau_1 = R_1C$ and $\tau_2 = R_2C$. There are certain trade-offs between the active and passive filters. The passive filter is relatively simple to implement, while the active filter has added complexity and must be carefully designed. One advantage of the ideal active filter is that it has infinite DC gain which eliminates the static phase error normally associated with passive filters. The DC gain of the passive filter is unity, or $F(0) = 1$. Static phase error is not desirable because it reduces the amount of jitter that can be tolerated by the PLL while properly re-timing the data.

To analyze static phase error we assume the PLL is in frequency lock with the incoming data stream and that a passive filter with $F(0) = 1$. In the locked state, the input frequency equals the output frequency, or $\omega_i = \omega_o$. In order to produce this frequency, the VCO must be driven by a certain voltage V_{co}, which is the control

voltage of the VCO in lock. Since $F(0) = 1$, this DC control voltage is generated by the phase detector, and assuming an ideal phase detector, this means that there is a static phase error between the VCO and data waveforms. Another source of static phase error is due to the output offset voltage of a non-ideal phase detector. This type of offset voltage is present even if there is no phase error between the VCO and data waveforms. The static phase error in lock is θ_{eo}

$$\theta_{eo} = \frac{V_{co}}{K_d F(0)} \tag{3.12}$$

The static phase error in Equation 3.12 is due to the fixed offset voltage needed to hold the VCO so that $\omega_i = \omega_o$, which is the locked condition. This design uses an active filter with a very large DC gain—the second term of Equation 3.12 can thus be ignored. We will return to this equation later to make sure that this assumption is valid.

From this point on we assume a first-order active filter, and the active subscript is dropped from the notation. Substituting the active filter transfer function from Equation 3.11 into the phase transfer function Equation 3.6 gives the general second-order PLL phase transfer function

$$H(s) = \frac{K_o K_d \left(\frac{s\tau_2+1}{\tau_1}\right)}{s^2 + s\left(\frac{K_o K_d \tau_2}{\tau_1}\right) + \left(\frac{K_o K_d}{\tau_1}\right)} \tag{3.13}$$

Using standard servo terminology, the phase transfer function can be written as

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.14}$$

where ω_n is the natural frequency

$$\omega_n = \sqrt{\frac{K_o K_d}{\tau_1}} \tag{3.15}$$

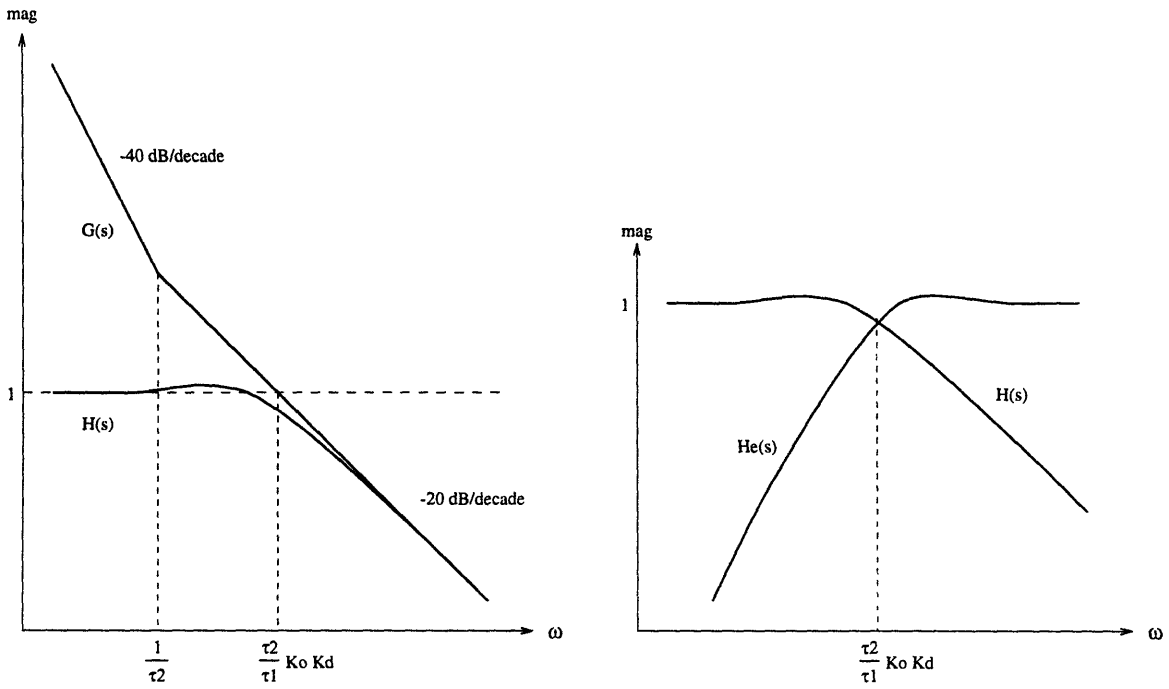


Figure 3-5: Second-order PLL open- and closed-loop response

and ζ is the damping ratio of the system

$$\zeta = \frac{\omega_n \tau_2}{2} \tag{3.16}$$

Using the same notation, the closed-loop phase error and open-loop forward gain transfer functions are, respectively

$$H_e(s) = \frac{\theta_e}{\theta_i} = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.17}$$

$$G(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2} \tag{3.18}$$

Figure 3-5a compares $H(s)$ and $G(s)$ for a second-order PLL. Figure 3-5b compares $H(s)$ and $H_e(s)$ for the same PLL. Notice that as $H(s)$ begins to roll-off at ω_{3dB}, the loop bandwidth, there is a slight peaking in the transfer function. This peaking is a normal phenomenon in second-order systems, and sometimes a bit of peaking is desired, but in communication systems it can have serious effects and is not tolerated. For example, in long fiber lines that contain hundreds of data regenerators, even

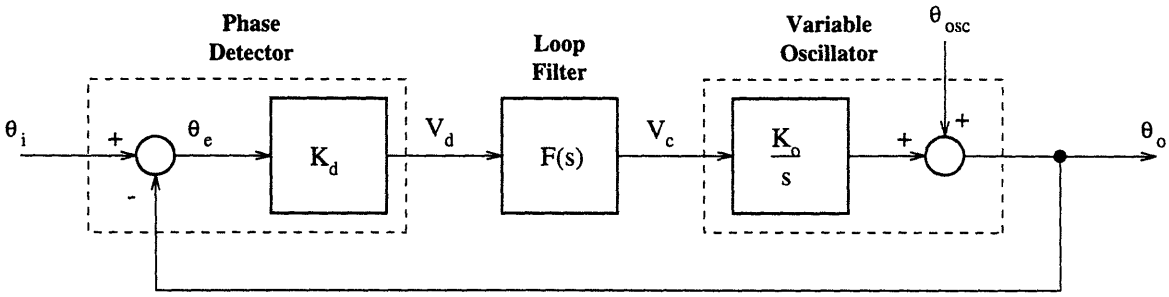


Figure 3-6: Linear phase-locked loop model with VCO phase noise

a small amount of jitter peaking from each regenerator can build up very quickly, eventually resulting in lost data.

3.4 PLL Tracking Performance

The PLL loop bandwidth divides the frequency response up into a high frequency region and low frequency region in which the tracking performance of the PLL must be analyzed. To make the analysis clearer a phase noise source internal to the PLL is included in the linear model. The VCO is a contributor to PLL phase noise as well as other physical noise generators. The resulting phase noise can be lumped together and modeled by VCO noise by adding in θ_{osc} into the system directly after the VCO—see Figure 3-6. The output phase θ_o from this model is

$$\begin{aligned} \theta_o &= \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\,\theta_i + \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\,\theta_{osc} \\ &= H(s)\,\theta_i + H_e(s)\,\theta_{osc} \end{aligned} \tag{3.19}$$

Within the loop bandwidth, when $s \ll j\omega_{3dB}$, Equation 3.19 reduces to

$$\theta_o = \theta_i + \frac{s^2}{\omega_n^2}\,\theta_{osc} \tag{3.20}$$

Equation 3.20 shows that in this region the system tracks changes in θ_i. Also, the VCO phase noise θ_{osc} below ω_{3dB} is attenuated.

Outside of the loop bandwidth, when $s \gg j\omega_{3dB}$, Equation 3.19 reduces to

$$\theta_o = \frac{2\zeta\omega_n}{s}\theta_i + \theta_{osc} \tag{3.21}$$

Equation 3.21 shows that outside the loop bandwidth the system does not track changes in θ_i, and the phase noise from the VCO, θ_{osc} passes through and appears at the output.

Referring back to Figure 3-5b, a few things can be summarized. The PLL response to the input phase θ_i determines the jitter transfer and jitter tolerance. These are important specifications in many system designs. To re-time the incoming data properly, the PLL must track slow changes of input phase. This type of phase change may be due to aging, temperature changes, and other wander at the transmitter. Even though the rate of change may be slow, the actual phase changes from these sources may be quite large and span multiple cycles. This is called jitter tolerance because the PLL must track or "tolerate" this jitter without losing lock. This topic will be covered further in later sections. When the rate of phase change is very low the PLL can track these changes perfectly, and $H(s) = 1$, and thus, the phase error transfer function $H_e(s)$ is negligible. This also applies to the VCO noise—the PLL can track and eliminate the VCO phase noise below ω_{3dB}.

When the rate of phase change from both the input and the VCO approaches and exceeds ω_{3dB}, the PLL is not fast enough to track the phase changes. At this point, $H(s)$ begins to roll-off and the phase error transfer function $H_e(s)$ approaches unity—the PLL effectively "ignores" the input phase changes, but as shown previously, the VCO phase noise is seen directly at the output. High-frequency jitter on the input is attenuated (as long as it can be tolerated) which is desirable, but VCO noise may be a problem. If the VCO noise is a problem, steps should be taken to reduce the VCO noise or increase the loop bandwidth. The PLL is usually not required to track changes in the input phase above certain frequencies, and the maximum jitter that must be tolerated is reduced.

Chapter 4

Phase-Locked Loop Parameter Calculation

There are two required specifications that constrain the PLL design parameters, jitter tolerance and pattern dependency. We will begin by analyzing the constraints due to the CCITT pattern dependence test. Next, we will analyze the constraints due to the SONET jitter tolerance requirements, then combine the constraints. After a brief analysis of jitter transfer, the PLL parameters will be calculated.

4.1 Pattern Dependence Test Constraints

In order to comply with the CCITT STM-N pattern dependence test, the PLL must not lose lock during extended periods of either all 0's or all 1's bits (See Appendix B, Section B.2). During a "dead" time such as this, unlikely as it may be, there will be no transitions on the serial data line and the phase detector will not be updated—the VCO can drift off frequency, causing a build-up of phase error which may cause the PLL to come unlocked.

It is fairly simple to get an estimate of the maximum phase error due to the required 72-bit long pattern of either all 0's or all 1's. In the worst case scenario the phase detector is railed at its maximum value during the entire 72-bit period, which corresponds to 28.9 ns. Because the filter transfer function is made up of both integral

and proportional gains, the resulting VCO control voltage has two components, both of which shift the VCO in frequency and thereby cause an accumulation of phase error.

Using the ideal active filter model from Section 3.3, the first contributing source to the filter output voltage is the integral gain of the filter—the filter capacitor integrates the phase error voltage. We define the maximum phase detector offset to be V_{dmax}—this is the maximum voltage output from the phase detector when it is railed, which is also K_d for the sample and hold phase detector. The basic capacitor equation is

$$\frac{dV_{cap}}{dt} = \frac{I}{C} \tag{4.1}$$

where I is the current switched through the capacitor, which is

$$I = \frac{V_{dmax}}{R_1} = \frac{K_d}{R_1} \tag{4.2}$$

Integrating both sides of Equation 4.1 and substituting in Equations 4.2 gives

$$V_{cap}(t) = \frac{K_d t}{R_1 C} \tag{4.3}$$

The second contributing source to the filter output voltage is the "zero" resistor R_2 which drops a constant voltage proportional to the input phase error voltage. The voltage from resistor R_2 is simply the current I through the resistor or

$$V_{res} = \frac{V_{dmax} R_2}{R_1} = \frac{K_d R_2}{R_1} \tag{4.4}$$

The sum of the capacitor and resistor voltages is applied to the VCO which causes a frequency change, Δf. This frequency chage is proportional to the VCO gain K_o.

$$\Delta f(t) = K_o \left(V_{cap}(t) + V_{res}\right) = \frac{K_o K_d}{R_1}\left(\frac{t}{C} + R_2\right) \tag{4.5}$$

Now, the phase change $\Delta\phi$ due to the frequency change $\Delta f(t)$ is calculated with the

following equation

$$\Delta\phi \;=\; 2\pi \int_0^T \Delta f(t)\; dt \tag{4.6}$$

Substituting $f(t)$ into this equation and evaluating the integral

$$\Delta\phi \;=\; 2\pi \frac{K_o K_d}{R_1} \int_0^T \left(\frac{t}{C} + R_2\right)\; dt \;=\; \pi \frac{K_o K_d}{R_1} \left(\frac{T^2}{2C} + R_2 T\right) \tag{4.7}$$

It is useful to know which PLL parameters influence this phase error. We know from the filter analysis in Section 3.3 that

$$\omega_n^2 \;=\; \frac{K_o K_d}{\tau_1} \;=\; \frac{K_o K_d}{R_1 C} \tag{4.8}$$

$$R_2 C \;=\; \tau_2 \;=\; \frac{2\zeta}{\omega_n} \tag{4.9}$$

Plugging these equations into Equation 4.7 gives

$$\Delta\phi \;=\; \pi\omega_n T\left(\omega_n T + 2\zeta\right) \tag{4.10}$$

Given a known maximum allowable $\Delta\phi$, Equation 4.10 becomes

$$\Delta\phi \;\leq\; \pi\omega_n T\left(\omega_n T + 2\zeta\right) \tag{4.11}$$

and we can solve this equation for a solution space. The result gives a curve which defines the allowed values on the ζ-ω_n plane. Solving Equation 4.11 for ω_n gives

$$\omega_n \;\leq\; \frac{-\zeta + \sqrt{\zeta^2 + \frac{\Delta\phi}{\pi}}}{T} \tag{4.12}$$

Figure 4-1 shows the constant error curves in the ζ-ω_n plane for phase error limits $\Delta\phi$ of $\frac{\pi}{2}$, $\frac{\pi}{3}$, and $\frac{\pi}{6}$. The allowed values for a given phase error limit are below and to the left of the curve. Now we proceed to find the constraints on the PLL parameters due to the jitter tolerance specification.

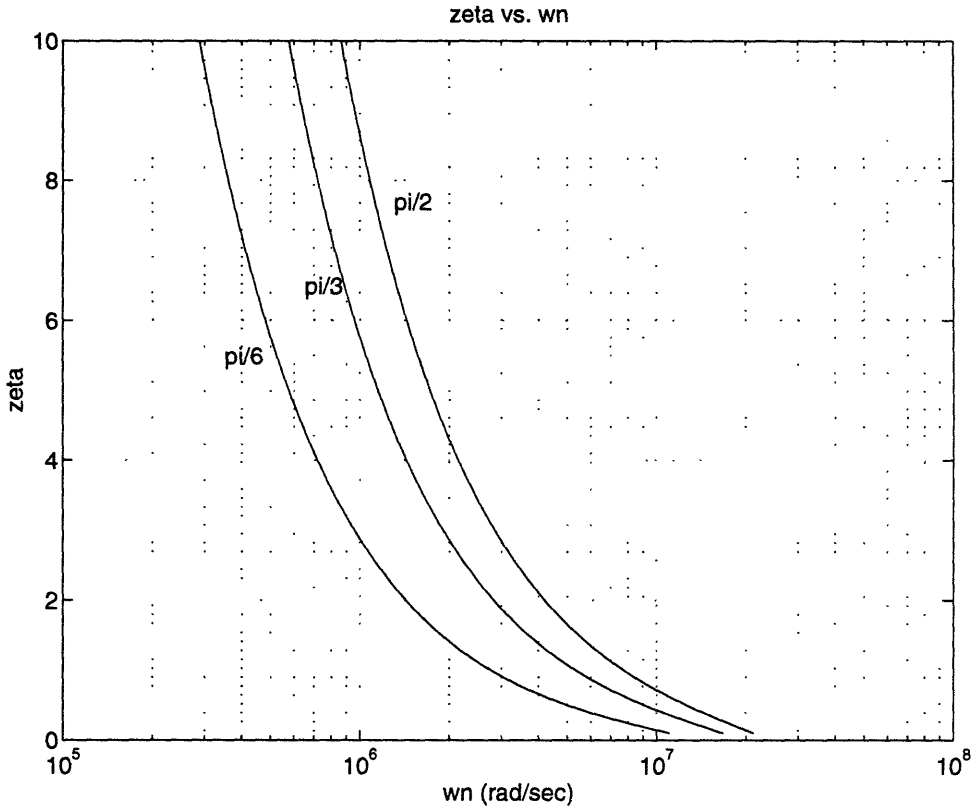


Figure 4-1: Constant CCITT phase error curves in the ζ-ω_n plane

4.2 Jitter Tolerance Constraints

As shown in Section 3.4 the jitter tolerance of the system is dependent in part upon the phase error transfer function $H_e(j\omega)$. Given a certain level of input jitter at a given modulation frequency ω_m, there will be a given amount of phase error on the VCO output. If this phase error is large enough the data will be mis-timed or the PLL might lose lock. The SONET standards specify a minimum jitter tolerance mask—a minimum amount of jitter that must be tolerated at the input of the PLL without losing data.

To analyze the constraints on the PLL parameters due to the jitter tolerance spec, we first calculate the magnitude of the phase error transfer function. Plugging $s = j\omega_m$ into Equation 3.17 gives

$$H_e(j\omega_m) = \frac{-\omega_m^2}{-\omega_m^2 + 2\zeta\omega_n j\omega_m + \omega_n^2} \tag{4.13}$$

The magnitude of $H_e(j\omega_m)$ is

$$|H_e(j\omega_m)| = \frac{\omega_m^2}{\sqrt{(\omega_n^2 - \omega_m^2)^2 + (2\zeta\omega_n\omega_m)^2}} \tag{4.14}$$

The phase error ϕ_e is simply the magnitude of the phase error transfer function multiplied by the input jitter ϕ_m

$$\phi_e = |H_e(j\omega_m)|\,\phi_m = \frac{\phi_m\omega_m^2}{\sqrt{(\omega_n^2 - \omega_m^2)^2 + (2\zeta\omega_n\omega_m)^2}} \tag{4.15}$$

The SONET jitter tolerance spec is a jitter vs. frequency graph as shown in Figure B-2. In order to construct a solution space and extract a ζ-ω_n curve similar to the one derived in Section 4.1, Equation 4.15 is evaluated over the SONET spec curve using a math analysis package. The SONET spec is entered into two vectors, a frequency vector ω_m containing all of the test points, and a phase vector θ_m which corresponds to the input jitter at these test points. The function created in the math analysis package has three arguements, ζ, ω_n, and the phase error *limit*. When

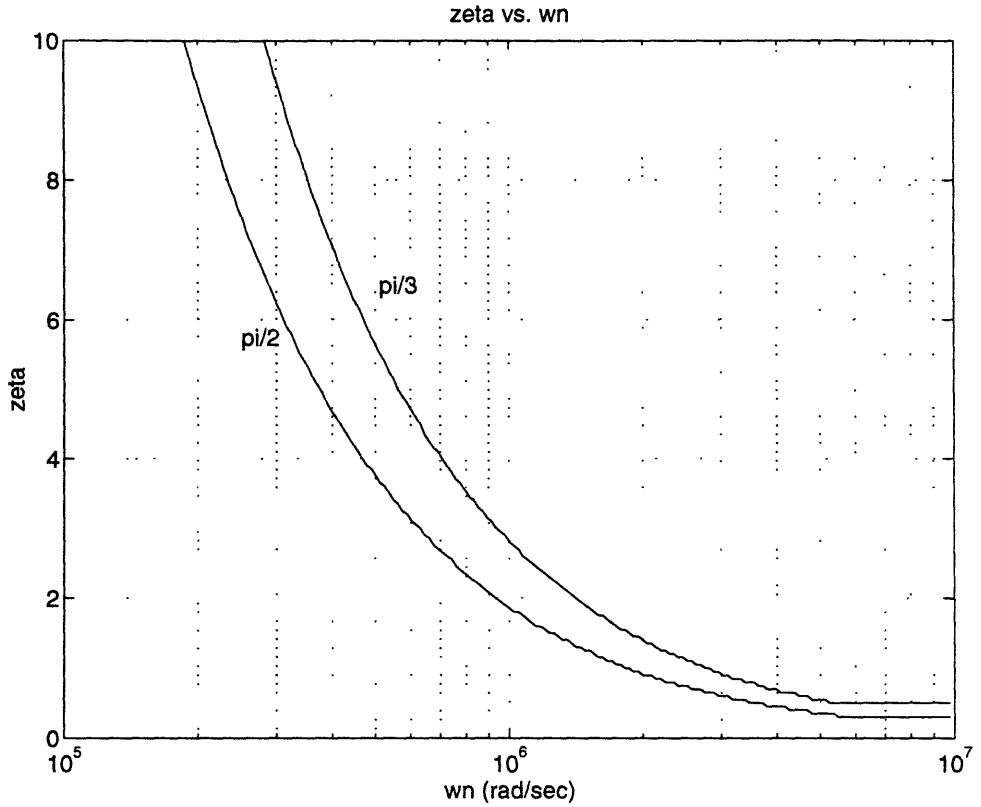


Figure 4-2: Constant SONET phase error curves in the ζ-ω_n plane

the function is called it simply calculates the phase error vs. frequency curve in response to the SONET spec input jitter at the specified (ζ,ω_n) point, and checks if the phase error exceeds the phase error *limit* at any point. If the phase error does not exceed *limit*, then that (ζ,ω_n) point is included in the solution space. The entire ζ-ω_n plane can be scanned with this function to construct the solution space, and constant maximum phase error curves can be extracted from this space.

Figure 4-2 shows the $\frac{\pi}{2}$ and $\frac{\pi}{3}$ constant phase error curves resulting from the SONET jitter tolerance analysis. The allowed values of ζ and ω_n are above the curve in this graph. The minimum possible value of phase error is $\frac{3\pi}{10}$. This is the high-frequency SONET jitter tolerance level, and corresponds to nearly the same curve as the one given by $\frac{\pi}{3}$ in Figure 4-2.

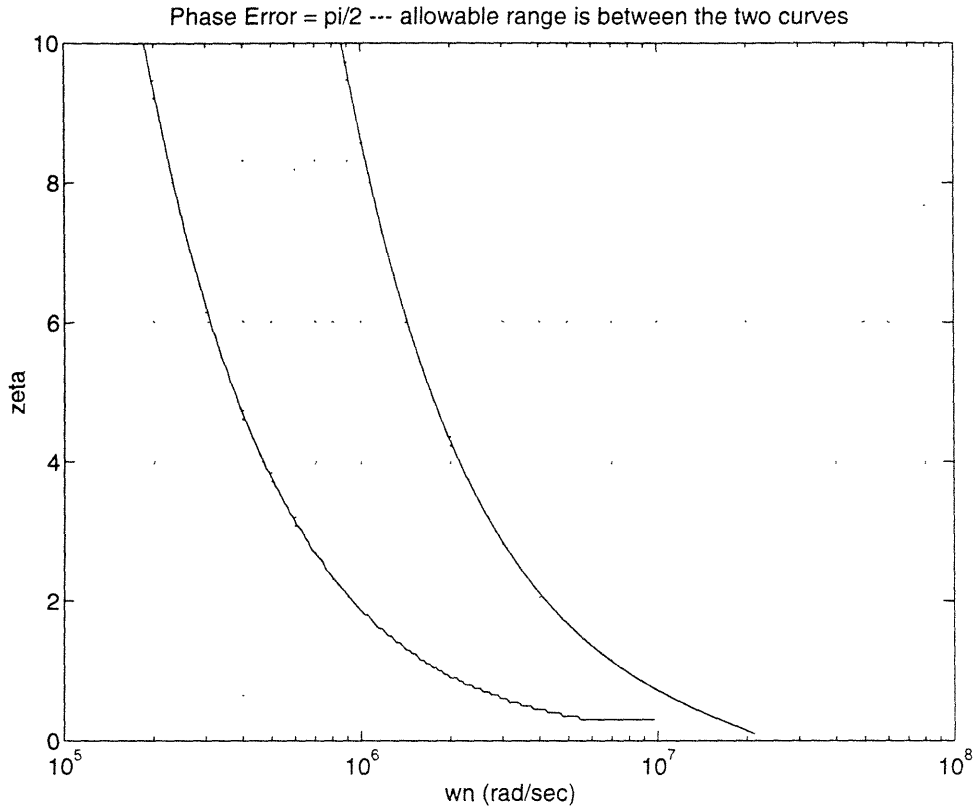


Figure 4-3: Allowed parameter range to guarantee a phase error of less than $\frac{\pi}{2}$

4.3 Phase-Locked Loop Parameter Curves

The curves from Figure 4-1 and Figure 4-2 are combined and the overall allowed values of ζ and ω_n are between the two curves. Figure 4-3 shows the allowed values of ζ and ω_n for a maximum phase error of $\frac{\pi}{2}$. The allowed values are between the curves. Similarly, Figure 4-4 shows the allowed values of ζ and ω_n for a maximum phase error of $\frac{\pi}{3}$. There is no advantage in choosing a point much higher than the $\frac{\pi}{3}$ SONET curve because it does not decrease the phase error after that point. For this reason we take the SONET curve at the minimum phase error to be the constraint curve. The (ζ,ω_n) point selected should be slightly above the $\frac{\pi}{3}$ SONET curve to ensure a bit of error margin. With this analysis we have essentially found a solution curve, which results in an infinite number of possible (ζ,ω_n) points. In order to further constrain the choices we now consider the SONET jitter transfer specification.

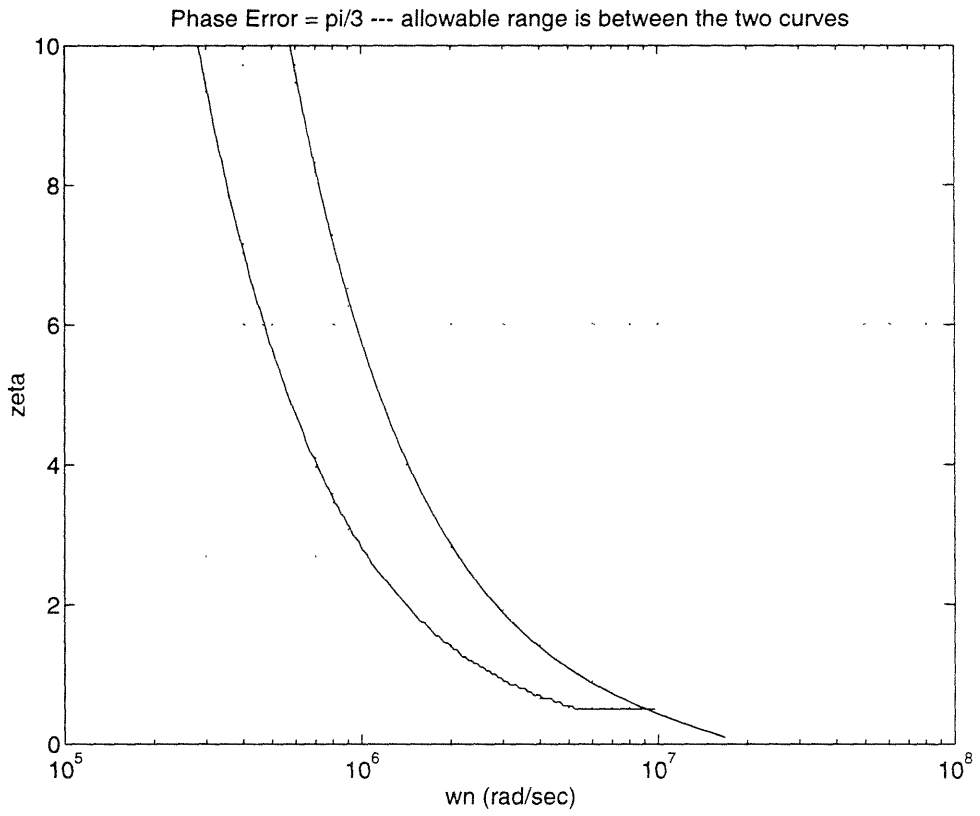


Figure 4-4: Allowed parameter range to guarantee a phase error of less than $\frac{\pi}{3}$

4.4 Jitter Transfer Constraints

The jitter transfer specification is not a requirement for this project because of the application, but it is useful to further constrain the PLL parameter selection and increase the performance of the system. The SONET standards call for a jitter transfer function that has a maximum peak of 0.1 dB and a maximum -3 dB point of 2 MHz. Given the jitter peak specification, we can solve for a minimum ζ.

The jitter transfer characteristic is simply the magnitude of the jitter transfer function, $H(j\omega)$

$$H(j\omega) = \frac{2\zeta\omega_n j\omega + \omega_n^2}{-\omega^2 + 2\zeta\omega_n j\omega + \omega_n^2} \tag{4.16}$$

$$|H(j\omega)| = \sqrt{\frac{(2\zeta\omega_n\omega)^2 + (\omega_n^2)^2}{(\omega_n^2 - \omega^2)^2 + (2\zeta\omega_n\omega)^2}} \tag{4.17}$$

To find the peak magnitude we take the derivative of $|H(j\omega)|$ and set it to zero. Since squaring is a monotonic function, we can simplify the math by taking the derivative of $|H(j\omega)|^2$ and set it to zero, which will give the same result.

$$\frac{d}{d\omega}|H(j\omega)|^2 = \frac{-4\omega\omega_n^2(2\zeta^2\omega^4 - \omega_n^4 + \omega^2\omega_n^2)}{(4\omega^2\omega_n^2\zeta^2 + \omega_n^4 - 2\omega^2\omega_n^2 + \omega^4)^2} = 0 \tag{4.18}$$

The value of ω that solves this is the frequency of the peak, ω_p

$$\omega_p = \frac{\omega_n}{2\zeta}\sqrt{\alpha - 1} \tag{4.19}$$

where

$$\alpha = \sqrt{1 + 8\zeta^2} \tag{4.20}$$

Substituting this result back into Equation 4.17 gives us the magnitude of the peak

$$M_p = 2\zeta^2\sqrt{\frac{2\alpha}{8\zeta^4\alpha + 4\zeta^2(2-\alpha) - \alpha + 1}} \tag{4.21}$$

Solving this for zeta gives

$$\zeta = \sqrt{\left(\frac{M_p}{2}\right)\frac{M_p - \sqrt{M_p^2 - 1}}{M_p^2 - 1}} \tag{4.22}$$

This shows that the magnitude of the phase transfer function peaking is a direct function of ζ. Using the SONET spec of $M_p = 0.1$ dB gives $\zeta = 4.32$, but given a 20% margin for error $\zeta = 5.18$ is chosen. Using the curves calculated in Section 4.3, a ζ of 5.18 gives a minimum ω_n of about 6.2×10^5 rad/sec or 95 kHz. Again allowing about 20% margin for error $\omega_n = 7.5 \times 10^5$ rad/sec or 120 kHz is chosen.

Now we can solve for the 3 dB point of the magnitude of the phase transfer function

$$|H(j\omega)| = \sqrt{\frac{(2\zeta\omega_n\omega)^2 + (\omega_n^2)^2}{(\omega_n^2 - \omega^2)^2 + (2\zeta\omega_n\omega)^2}} = \frac{1}{\sqrt{2}} \tag{4.23}$$

and now solving this for ω gives the 3 dB point in terms of ω_n

$$\omega_{3dB} = \omega_n\sqrt{1 + 2\zeta^2 + \sqrt{2 + 4\zeta^2 + 4\zeta^4}} \tag{4.24}$$

Substituting $\omega_n = 7.5 \times 10^5$ rad/sec gives a loop bandwidth $\omega_{3dB} = 7.8 \times 10^6$ rad/sec or 1.2 MHz. From this analysis we can see that the PLL can meet the SONET jitter transfer specification even though the intended application of this system does not require this specification.

Chapter 5

Clock Recovery and Data Re-timing Circuit Design

The clock recovery subcircuit is the heart of the receiver project. Without it, the correct phase and frequency clock cannot be extracted and the serial data cannot be recovered. The design of this block is very sensitive, and if it is not designed properly, the resulting bit error rate will be too large for practical use.

A block-level diagram of the clock recovery and data retiming system is shown in Figure 5-1. The inputs to this system consist of the high-speed serial data (either SDI or LBSDI), a low-speed reference clock (RCLKI), the Test Enable line (TESTEN) and the corresponding Test Clock In-phase and Quadrature lines (TCLKI and TCLKI), and a Loss-Of-Signal Indicator (LOSI) line. During normal operating conditions LOSI will be low and the PLL will lock to the high-speed data input, but when the LOSI line is asserted, the PLL switches over and locks onto the reference clock input to provide a stable clock output.

The high-speed PLL locks onto the 2.48832 Gb/s NRZ data, which can be thought of as a 1.24416 Ghz pseudo-sinusoid. The extracted clock signal is a phase-locked 2.48832 Ghz clock which is divided down to 1.24416 Ghz because the downstream circuitry (frame detect, demux, etc) uses both edges of the clock, instead of just a rising edge. To re-time the data, two flip-flops which operate on opposite clock edges are clocked from the VCOI signal divided down by 2.

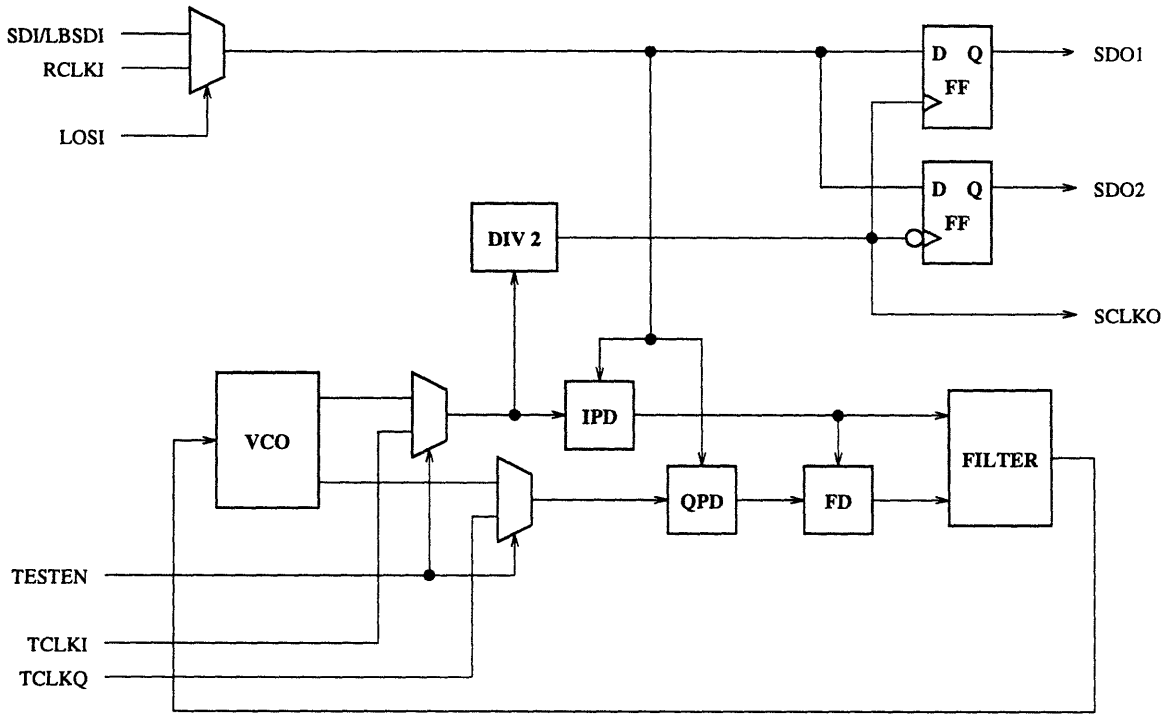


Figure 5-1: Block Diagram of the clock recovery and data retiming system

5.1 Phase and Frequency Detection

The FPLL uses an interesting phase and frequency detection scheme, derived from an article by Ansgar Pottbäcker, et al[5]. Pottbäcker states the benefits of using this scheme—it uses a relatively simple and low-speed phase and frequency detection system, no pre-processing of the NRZ data stream is needed, and a half-bit delay generator is not needed. See Figure 5-2 for a simplified diagram of the Pottbäcker FPLL.

The phase-frequency detector requires a VCO that produces in-phase and quadrature outputs. In Pottbäcker's paper, the two phase detectors (IPD and QPD) are essentially digital samplers which sample on both edges of the input clock. Instead of using the VCO waveform to sample the data stream, like typical digital phase detectors, the serial input data is used as a clock input to the phase detectors which sample the VCO waveforms—one phase detector samples the in-phase VCO waveform (VCOI) and the other phase detector samples the quadrature VCO waveform (VCOQ). First we assume that the VCO is not frequency-locked to the data. Fig-

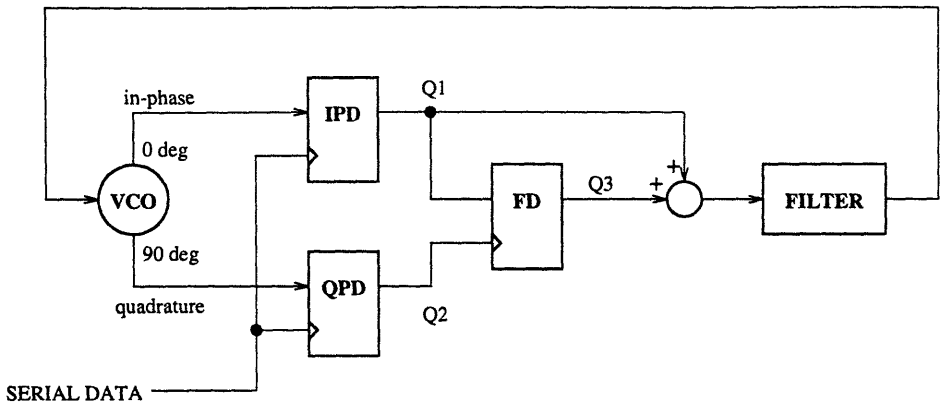


Figure 5-2: Block Diagram of the FPLL

Q1	Q2	Q3
Rising	1	0
Falling	1	0
Rising	-1	-1
Falling	-1	1

Table 5.1: Frequency detector state table

ure 5-3(a) shows a timing diagram for the phase detectors when the VCO frequency f_{vco} is higher than the bit rate f_b, and Figure 5-3(b) when the VCO frequency is lower than the bit rate. In the un-locked condition, the sampled Q1 and Q2 waveforms are the beat notes between the VCO and data frequency, and the Q1 and Q2 signals will always be in a quadrature relationship with one another—one signal will lag the other by 90 degrees—but which signal lags and leads depends on the sign of the frequency difference between the data and the VCO. Figure 5-3(a) shows that when $f_{vco} > f_b$, Q1 leads Q2 by 90 degrees, and in Figure 5-3(b) when $f_{vco} < f_b$, Q2 leads Q1 by 90 degrees. The frequency detector (FD) processes these quadrature beat notes, and depending on their quadrature relationship, the DC average of the Q3 output is the sign of the frequency difference. Table 5.1 shows the state table for the frequency detector which results in the proper beat note processing. The Q3 signal, after passing through the loop filter, drives the VCO towards lock. Notice that the FD output only changes state on a transition of Q1, either rising or falling, and that the FD has a ternary output, either 1, 0, or -1. Referring back to Figure 5-3, the FD output

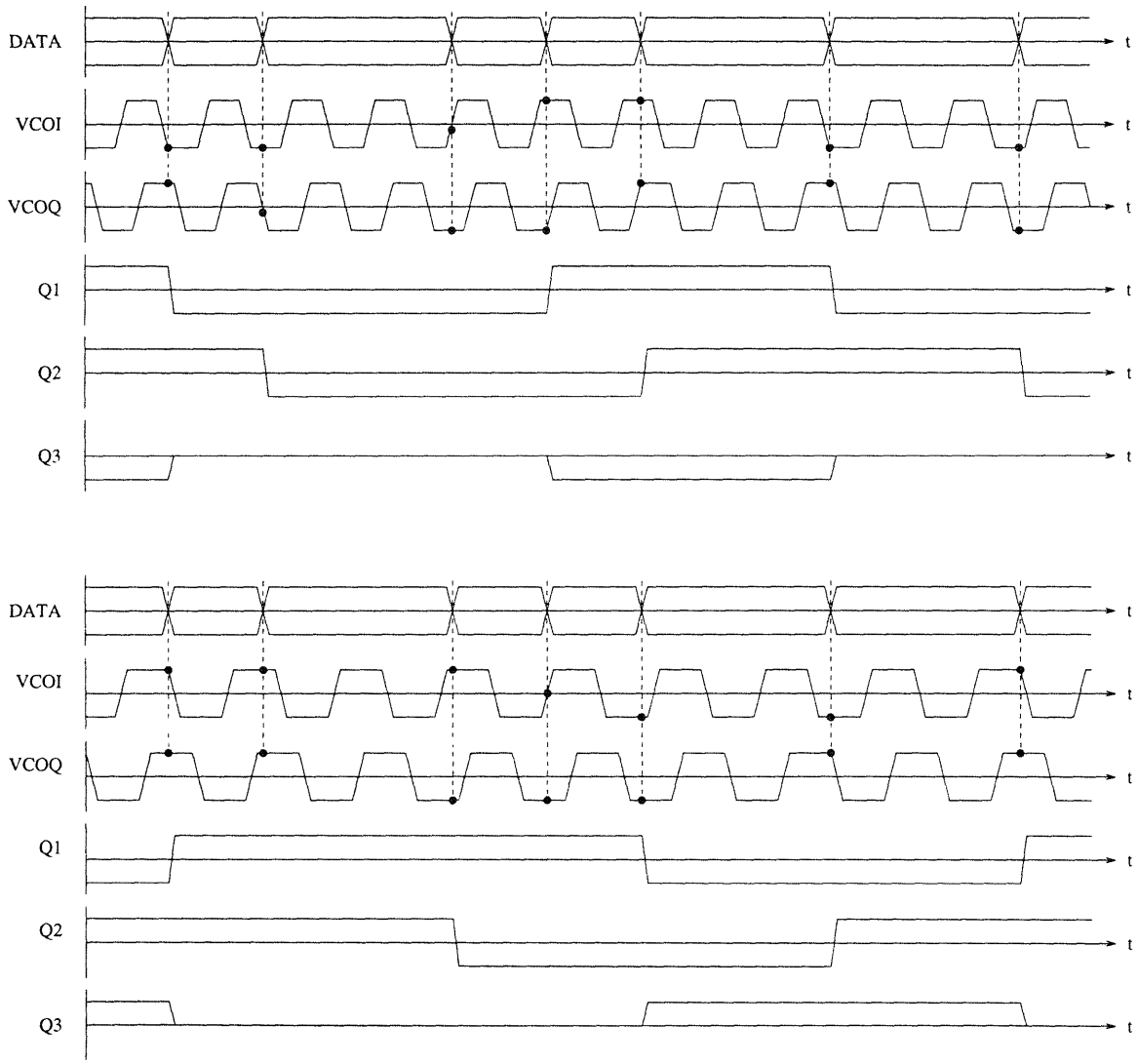


Figure 5-3: Timing diagram of the FPLL out of frequency lock

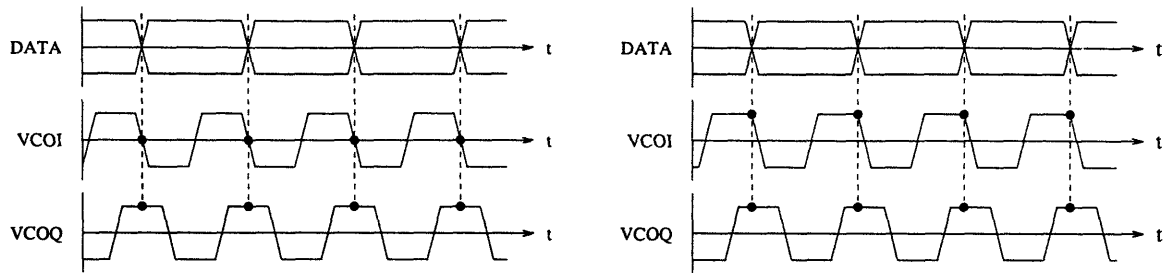


Figure 5-4: Timing diagram of the FPLL in frequency lock when the phase error is zero (a) and slightly off phase (b)

with these properties can be thought of as cancelling a half-cycle of the Q1 waveform when Q1 and Q3 are summed together, resulting in a DC component which drives the VCO towards lock. Figure 5-3 also shows that the proper waveform processing still occurs even if there are missing transitions in the data waveform.

Now we will look at the case when the FPLL is in both frequency and phase lock, or $f_{vco} = f_b$ and the phase error is exactly zero. See Figure 5-4(a). In this ideal case, the falling edge of the VCOI waveform is aligned exactly with each transition in the data stream—the in-phase PD (IPD) clocks in Q1 = 0 (or "metastable" state) and the quadrature PD (QPD) clocks in Q2 = 1. Again referring back to Table 5.1, this is the state in which the FD is not active (Q3 = 0) and contributes nothing to the phase error signal. This results in a stable operating point and a perfect lock when Q1 = 0 and Q3 = 0. In reality, jitter on both the data and VCO waveforms shifts the data and VCO edges slightly which toggles Q1 between -1 and 1 randomly. This is also a stable operating point because the DC average of Q1 is zero.

Now we will assume that the FPLL is in frequency lock but not in phase lock. This will be the case if the VCO begins to slide off-phase as shown in Figure 5-4(b). In this case the IPD responds as a phase detector whose output voltage corresponds to the sign of the phase difference. This type of phase detector is known as a "bang-bang" phase detector because of it's digital behavior. There are certain advantages and disadvantages of using a bang-bang type phase detector which will be discussed later. Notice that as long as the phase difference (or jitter) is less than 90 degrees of the VCO waveform, Q2 remains in state 1 and the FD does not become active. Referring to Table 5.1, the frequency detector remains inactive until the VCO drifts

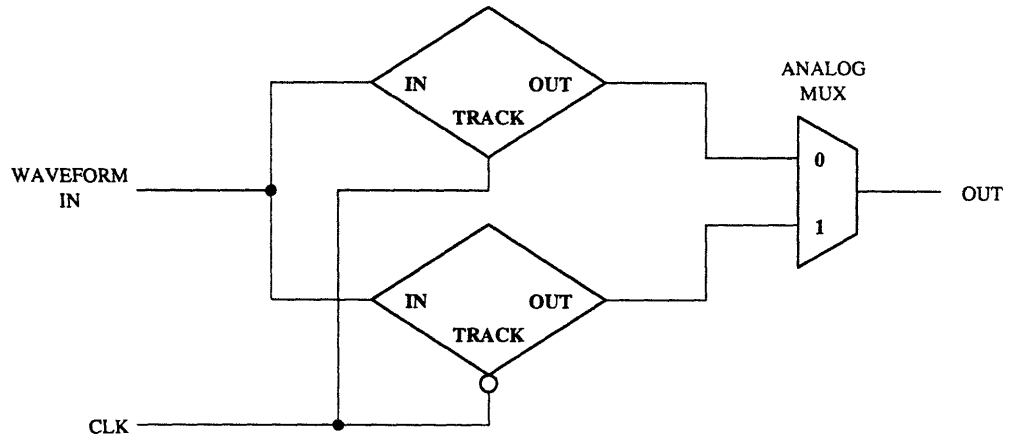


Figure 5-5: Sample and hold phase detector

180 degrees off phase which corresponds to a transition of Q1 when Q2 = 0.

Under normal frequency-locked operation, the transitions of the data waveform will be jittering around the VCOI falling edge slightly. This causes the appropriate phase error signal to be generated, but again, as long as the phase jitter is kept within a $\pm\pi$ range no frequency error signal is generated.

5.2 Phase and Frequency Detector Design

A high-performance analog sample-and-hold circuit is used as the phase detector. The sample-and-hold phase detector design is based on the same concept as the flip-latch. The circuit contains two track-and-hold analog "latches" and a 2:1 multiplexer, as shown in Figure 5-5. Appendix D covers the basic operation of this type of sample-and-hold phase detector. Each track-and-hold block is based on the diode bridge topology. Because the diode bridge is a single-ended topology, each of the track-and-hold blocks consist of two separate diode bridge circuits, one for each of the differential lines. Because of this symmetry, only one of the two diode bridge circuits from each track-and-hold is analyzed.

Figure 5-6 shows the basic diode bridge topology. The main sampling diode bridge consists of diodes D1 through D4. Transistors Q1 and Q2 form a current switch which switches the diode bridge from hold mode to track mode and vice versa, and diodes D5 and D6 are clamping diodes for operation in the hold mode. Inputs *trackh* and *trackl*

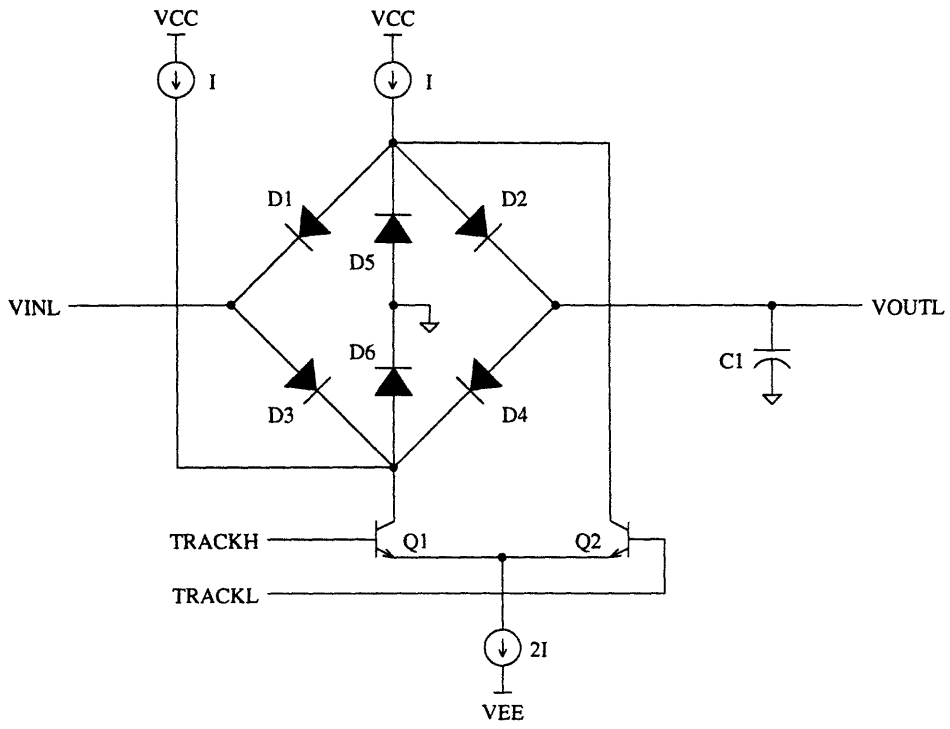


Figure 5-6: One half of a differential track and hold circuit

are the high and low sides of the digital differential track signal, respectively. First, lets analyze the circuit in track mode, when $(trackh - trackl) > 0$, so that current switch transistor Q2 is cut-off and Q1 conducts a current of $2I$. When the input is at about the same voltage as the output, the current is split equally between the right and left hand sides of the diode bridge, as shown in the simplified Figure 5-7(a). In this state all of the diodes are conducting and have equal voltage drop, resulting in zero offset from input to output. Also, in this state the input supplies zero current. Because the input and output signals are one half of a differential signal, input to output offset is not a problem within certain limits.

The analysis of the diode bridge dynamics in track mode must be broken into two parts, the first when the input is approximately equal to the output, and second when there is a large difference between the input and output (when the circuit switches from hold to track mode for instance). When the input within a few $\frac{kT}{q}$ of the output, the diodes can be modeled simply as resistors of value $r_d = \frac{kT}{qI_D}$. In this case the time constant associated with the holding capacitor C_1 is due to the resistance of the

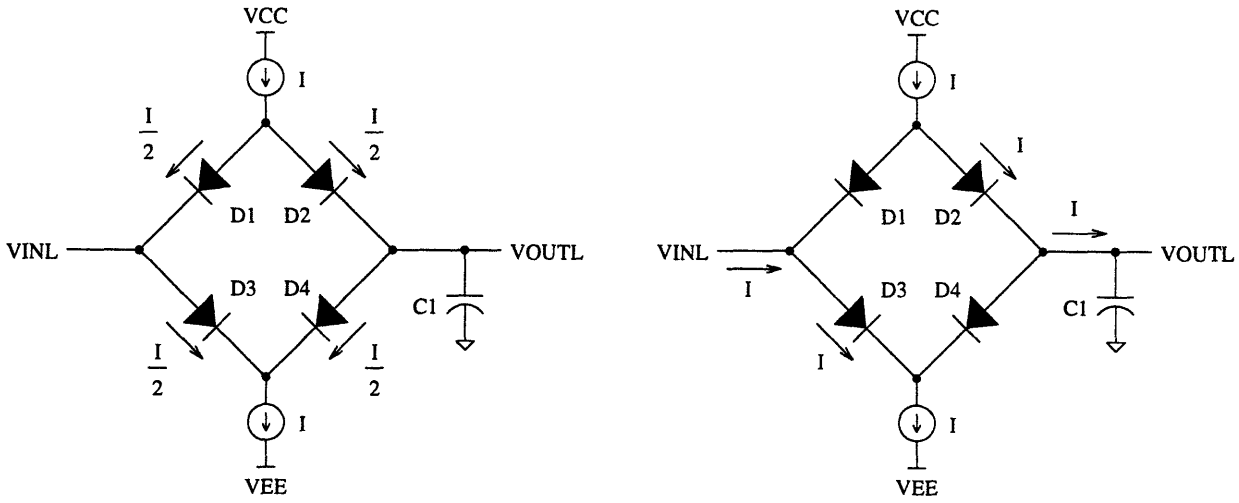


Figure 5-7: Simplified diode bridge in track mode

"resistor network" seen by $C1$.

$$\tau = r_d C_1 = \frac{2kTC_1}{qI} \tag{5.1}$$

When the input is greater than (or less than) a few $\frac{kT}{q}$ from the output, the limiting factor in charging time is the slew rate due to the maximum current I available at the output. Figure 5-7(b) shows the diode bridge when the input is at a much higher voltage than the output. In this case diodes D2 and D3 carry nearly all the current and the capacitor is charged with a constant current I, resulting in a slew rate of I/C. One disadvantage of this is that the input must also provide a current of I, therefore a unity-gain buffer with a high current drive capability is needed to drive the track-and-hold circuits.

Referring back to Figure 5-6, when the circuit is in the hold mode, or when $(trackh - trackl) < 0$, transistor Q1 is cut-off and Q2 conducts a current of $2I$. In this mode, current I flows through diodes D5 and D6 which reverse biases diodes D1 through D4 by $-V_{BE}$. The back-to-back reversed biased diodes D1-D2 and D3-D4 effectively disconnects the input and output. Leakage currents from the diodes will tend to charge or discharge C_1 and cause errors. Similarly, any input current needed for the output buffer which follows the track-and-hold will tend to charge or discharge the capacitor. If a standard emitter follower is used, current will be drawn

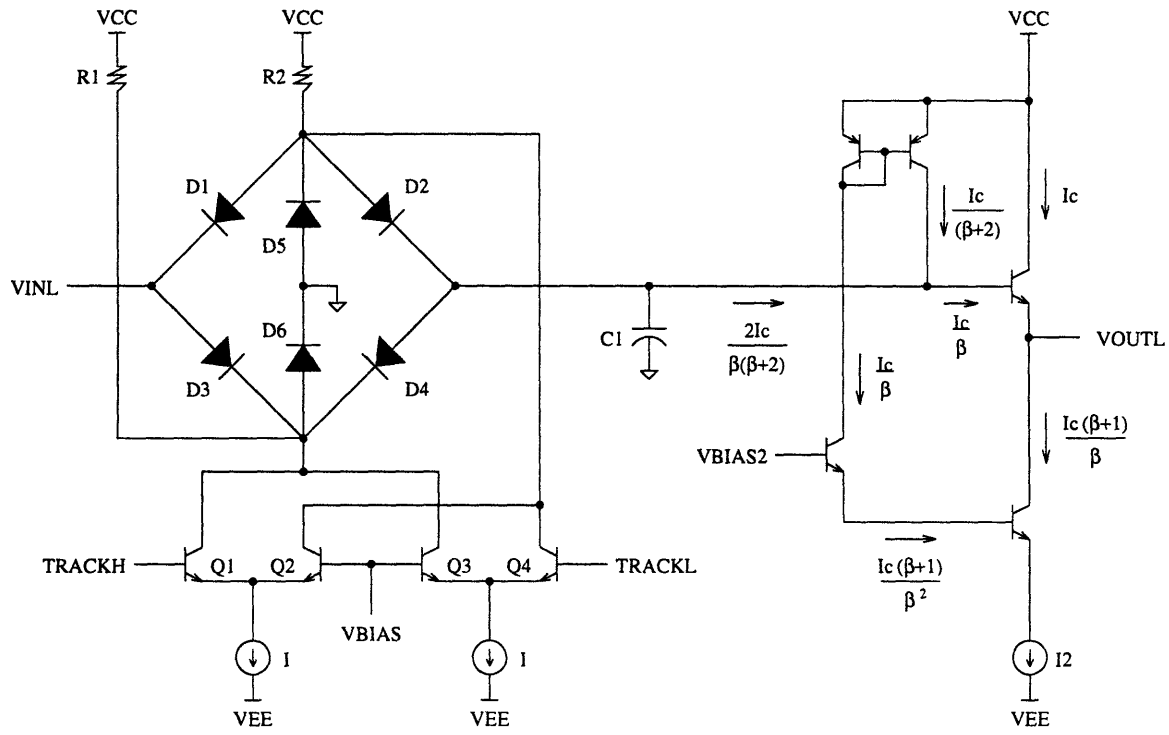


Figure 5-8: Modified track and hold circuit

out of the capacitor, causing the output voltage to decrease linearly with time. As long as matching between devices is good the resulting errors will be seen only as a common-mode error which will be rejected by the following stages.

In order to improve the tracking capability, it is desirable to use a small charging capacitor along with high bias current. When a small capacitor is used however, the capacitor will discharge faster in the hold mode due to the input current required for the following buffer stage. As previously stated, this results in a common-mode offset, which is not a problem, but the capacitor voltage should not drift significantly over a length of time when there are no data transitions. For example, with random data input into the phase detector there is no guarantee that the next data edge will occur within a bit period or two, and thus the hold mode could last for many bit periods before switching back into track mode.

A few modifications to the track and hold circuit are shown in Figure 5-8. In this diagram the output buffer is included, and the bias currents are shown. With the current cancelling circuitry, the input current to the buffer is reduced by about a

factor of β. There is a total offset of about one V_{BE} drop from input to output—the input is centered around zero volts so the output will be about one V_{BE} drop below zero, a standard ECL level one output.

Figure 5-8 shows that an f_T doubler is used in the current switch to increase the switching speed. Also, resistors R_1 and R_2 replace the current sources at the top. This reduces the complexity of implementing a true current source, and the offset side effects of this change are not important. Mismatches between the current sources and the current sink cause direct offset errors. Using resistors in place of the current sources decreases the linearity of the circuit because the finite resistance of the resistive "current sources." The current through the diode bridge changes as a function of the input and output voltage, but with good diode matching the current I is split between the bridge diodes equally and the input to output offset is uneffected.

The bandwidth of the diode bridge and hold capacitor in track mode is about 2.4 GHz in order to track the input waveform from the VCO. The bandwidth of the output buffer does not need to be this high because the output is the beat note between the VCO frequency and the data rate, which is about 100 MHz at most when out of frequency-lock and much less than this in normal in-lock operation. The major constraint to the bandwidth is that the effects of the limited bandwidth contribute another pole to the overall open-loop transfer function. To be safe, the higher-order poles such as these should be at least one order of magnitude higher frequency than the crossover frequency of about 1 MHz. With a bandwidth greater than 10 MHz to 50 MHz the higher-order poles do not significantly contribute to the phase margin of the overall loop.

The unity-gain input buffer for the phase detector is shown in Figure 5-9. Not only does the input buffer provide sufficient current drive capability for the diode bridge track-and-hold circuits, it also level shifts the input from ECL level-1 to ground so that the diode bridge can also be ground centered. Transistors Q1 and Q2 along with resistors R1 and R2 make up the input analog differential pair. Transistors Q3 and Q4 are cascode devices which increase the amplifier bandwidth, and resistors R3 and R4 are the amplifier load resistors which are chosen to set the gain to unity.

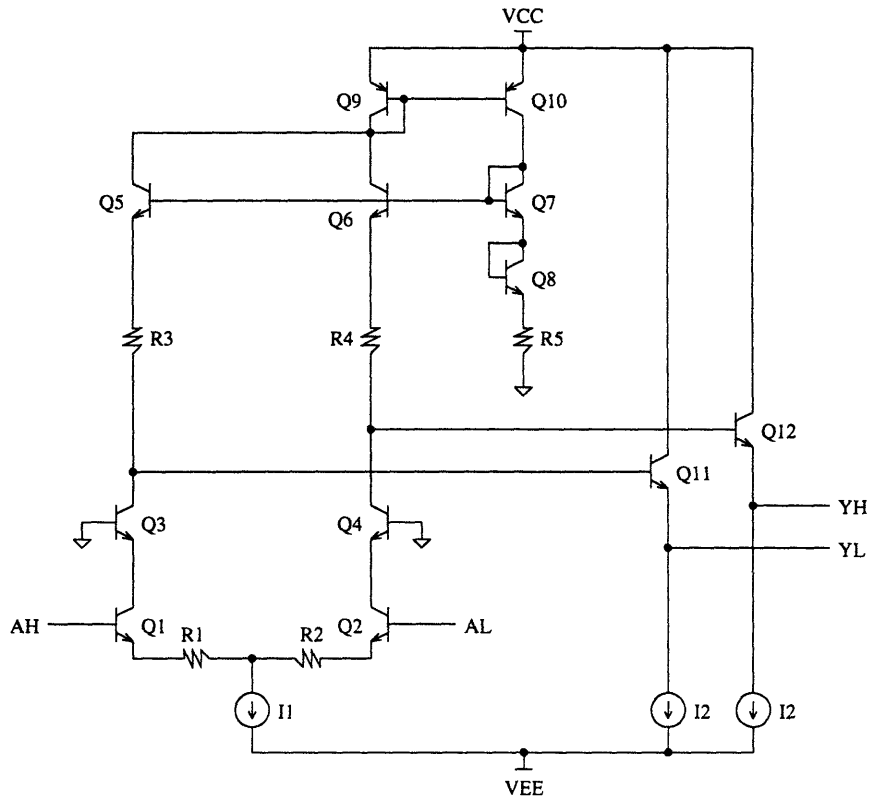


Figure 5-9: Track-and-hold input buffer

Transistors Q5 through Q8 along with resistor R5 set the output level at approximately ground by matching the number of V_{BE} drops and resistive drops between the output and the bottom of R5. Transistors Q9 and Q10 mirror the current from the differential pair into the Q7-Q8-R5 string. Finally, transistors Q11 and Q12 are emitter-followers which buffer and provide the necessary current drive. The input buffer has a bandwidth of about 2.5 GHz.

To finish up the analysis of the phase detector, the analog multiplexer is now presented. Figure 5-10 shows the analog mux. Both of the inputs to the mux are at level-one. The select lines are digital, so no emitter degeneration is used in this switch. Similar to the track-and-hold output buffer, the bandwidth of the multiplexer is only about 100 MHz.

The entire track-and-hold phase detector (THPD) has an internal delay of 60 ps from clock edge to sample time, which is about 54 degrees or 0.94 radians. This must be taken into account later during the data re-timing analysis. The phase detector

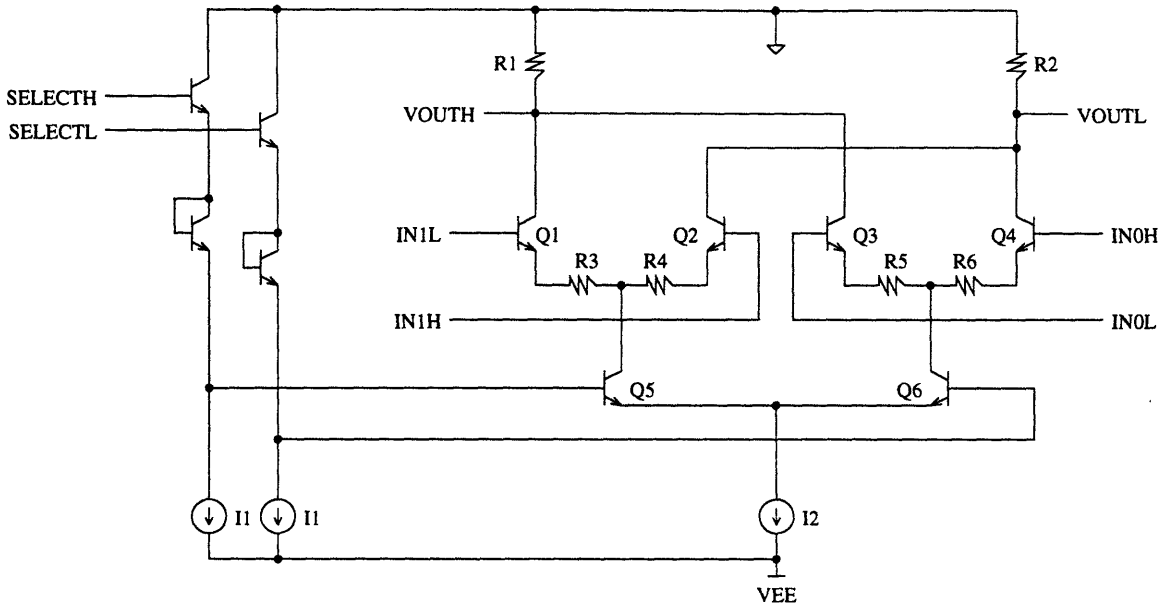


Figure 5-10: Analog Multiplexer

gain is simply the slope of the phase detector characteristic at zero degrees phase error. Since this phase detector is a sample-and-hold, it's characteristic shape is the same as the input waveform shape, a sine wave, as shown in Figure 5-11. The equation for this characteristic is

$$V_d = 0.2 \sin(\theta_e) \tag{5.2}$$

The linearized phase detector gain $K_d = 200$ mV/rad.

The frequency detector (FD) is implemented digitally, similar to Pottbäcker's frequency detector. Figure 5-12 shows the diagram of the FD without the input and output level shifters for simplicity. The IPD and QPD inputs are from the in-phase and quadrature phase detectors respectively. The FD is based on the flip-latch concept with two latches and a 2:1 multiplexer. In this case the multiplexer output is modified to produce the correct ternary output as given in Table 5.1. First, using a flip-latch analogy, the output can only change when there is a transition on IPD, regardless of the polarity of the transition. Now, if QPD is low during an IPD positive transition, current I1 is routed through mux transistors Q12 and Q10 so that VOUTH is low and VOUTL is high, resulting a "-1" at the output. If QPD is low during an IPD negative transition, current I1 is routed through Q11 and Q7 so that VOUTH

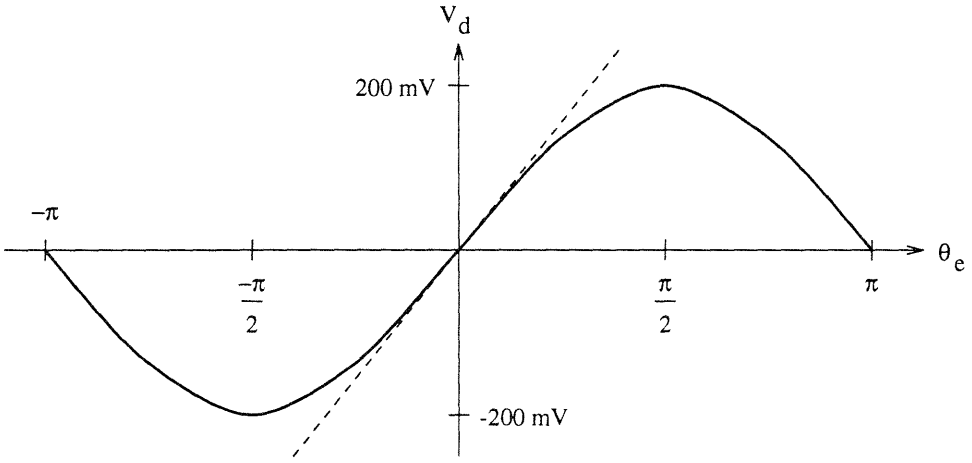


Figure 5-11: Sample-and-hold phase detector characteristic

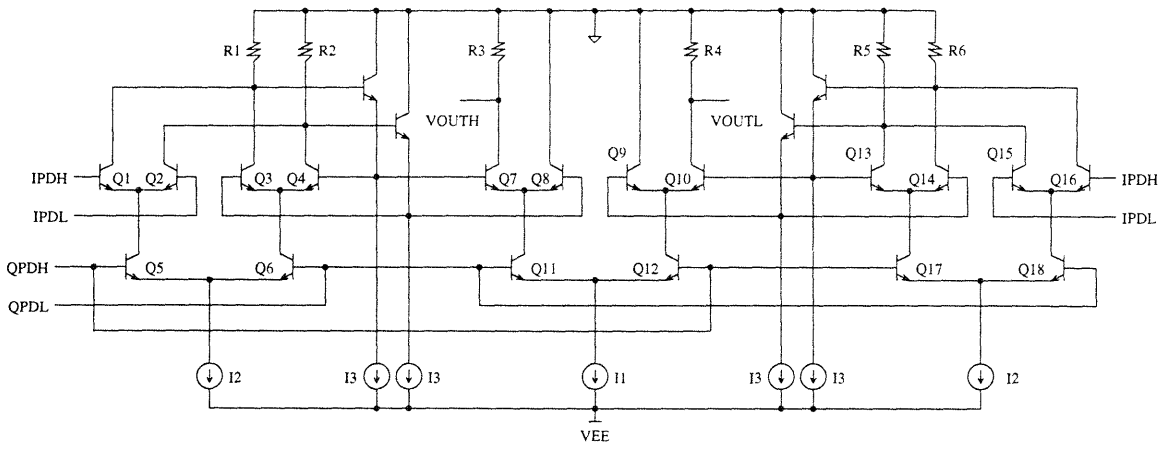


Figure 5-12: Frequency Detector

is high and VOUTL is low, resulting a "1" at the output. If QPD is high during an IPD positive transition, current I1 is routed through Q12 and Q9 which is dumped to ground—no current flows through R3 or R4 resulting in a differential "0" at the output. Similarly, if QPD is high during an IPD negative transition, current I1 is routed through Q11 and Q8, also resulting in a differential "0" at the output. This operation implements the functionality described in Table 5.1.

5.3 Signal Multiplexers

There are various multiplexers internal to the module, both digital and analog. All of these muxes have a standard ECL mux topology.

There are two digital muxes directly following the on-chip post-amp in order to select the input data source, and one digital mux inside the clock recovery block which selects the alternate RCLKI source in case of an LOS condition. Also, three digital muxes are used to enable the high-speed serial data outputs SDO1 and SDO2. These muxes are all high-speed custom digital circuits that operate with ±200 mV differential swing.

There are two analog muxes inside the clock recovery block which select either the VCOI and VCOQ sources or the TCLKI and TCLKQ sources. These are used to test the frequency and phase and detectors at low speeds. Since these multiplexers operate at a single frequency in normal operation, 2.48832 GHz, and a fixed 200 mV input and output amplitude, they are designed for unity-gain under these conditions. Since these analog muxes are never switched dynamically, the AC characteristics for the switch transistors need not be analyzed. In this case the mux consists of two independent amplifiers, one of which is selected at a given time. The analog muxes were characterized with an output cload of 200 fF and estimated internal parasitic capacitances. The linear input range is about 250 mV, and the bandwidth is about 2.8 GHz. Since these muxes will never have the two inputs active at the same time, deselected input-to-output isolation is not a factor in the design.

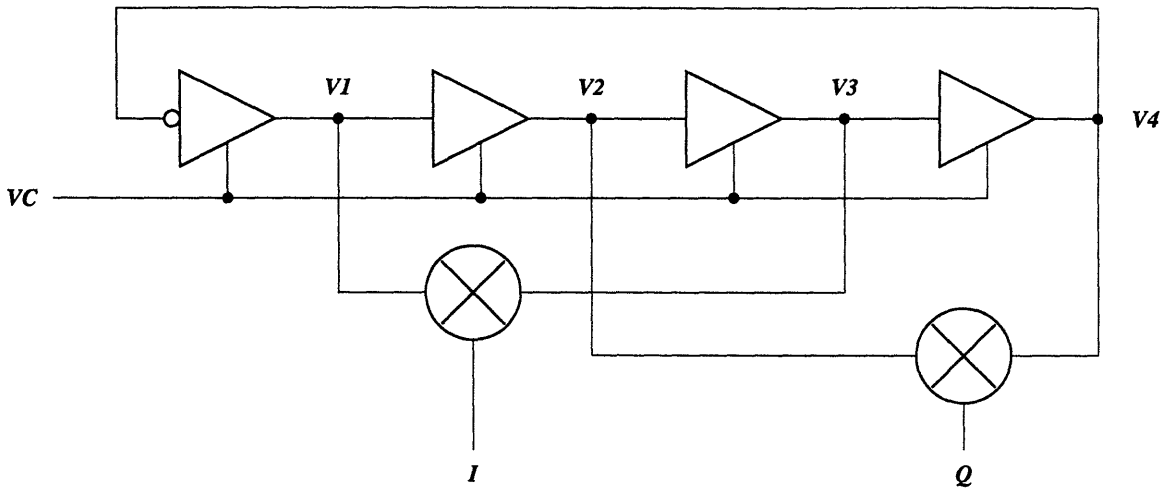


Figure 5-13: Ring VCO with in-phase and quadrature frequency-doubled outputs

5.4 VCO Design

A ring oscillator was selected to be used as the VCO because of the ease of integration, the wide tuning range possible, and the availability of in-phase and quadrature outputs, which is required for the frequency and phase detectors. While a wide tuning range is not necessary in some respects—the input data rate cannot change more than ± 20 ppm—the variation in center frequency over process variations is substantial, and thus a wide tuning range is used.

The oscillator uses four differential, variable-delay stages with a single inversion. See Figure 5-13 for a diagram of the ring VCO. Signal taps are taken off between each delay stage, V_1 through V_4, and the propagation delay through each stage is t_{pd}. The period of the oscillator output waveform is $8t_{pd}$ which is a full cycle, or 360 degrees. Therefore, 180 degrees of the output waveform is equal to $4t_{pd}$, and 90 degrees is equal to $2t_{pd}$. Observing a waveform at any tap along the line, for example V_1, the waveform at a tap which is $2t_{pd}$ (two stages later in the ring), or V_3 in this case, lags by 90 degrees. Thus, V_1 and V_3 are in quadrature, and also V_2 and V_4 are in quadrature. Now each pair of in-phase and quadrature taps are multiplied together—V_1 and V_3 are multiplied together to produce V_I, as well as V_2 and V_4 to produce V_Q. The amplitude of each sinusoid is the same, as well as the frequency, and we can define V_1 as the phase reference waveform. The result of the multiplication of

the sinusoids is given by the following equations.

$$\begin{aligned} V_1 &= A\sin(\omega t) \\ V_2 &= A\sin\left(\omega t + \frac{\pi}{4}\right) \\ V_3 &= A\sin\left(\omega t + \frac{\pi}{2}\right) = A\cos(\omega t) \\ V_4 &= A\sin\left(\omega t + \frac{3\pi}{4}\right) = A\cos\left(\omega t + \frac{\pi}{4}\right) \end{aligned}$$

$$V_I = V_1 V_3 = A^2\sin(\omega t)\cos(\omega t) = \frac{A^2}{2}\sin(2\omega t) \tag{5.3}$$

$$V_Q = V_2 V_4 = A^2\sin\left(\omega t + \frac{\pi}{4}\right)\cos\left(\omega t + \frac{\pi}{4}\right) = \frac{A^2}{2}\sin\left(2\omega t + \frac{\pi}{2}\right) \tag{5.4}$$

Equation 5.3 shows that when V_1 and V_3 are multiplied together, the resulting waveform V_I is twice the frequency. Furthermore, Equation 5.4 shows the result of multiplying V_2 and V_4 together produces a waveform V_Q that is not only twice the frequency, but also in quadrature with V_I. The ring oscillator runs at 1.24416 GHz, and the two outputs I and Q give the required "in-phase" and "quadrature" 2.48832 GHz waveforms.

The delay stage used in the VCO is based on a patent by Art Metz[4]. The circuit gives a very wide linear delay vs. input voltage characteristic. See Figure 5-14 for the schematic of this variable delay cell. The delay cell is variable by steering current through either Q1 and Q4 or Q2 and Q3. The subcircuit that contains transistors Q9 and Q10, and diodes D1 and D2 is the input VCO control voltage generator which is common to all of the delay cells—this subcircuit linearizes the VCO tuning curve and will be expanded upon later. The delay cell is most easily analyzed for two separate cases, one when $(steerh - steerl) > 0$ and the other when $(steerh - steerl) < 0$.

Figure 5-15 shows the simplified equivalent variable delay cell when the current I1 is steered through Q5 and Q8. In this state, transistors Q6 and Q7 are cut off, and Q5 and Q8 simply act as cascode transistors with their bases at a constant V_{bias}. Because of this, the bottom plate of capacitors C1 and C2 are fixed, and thus the input voltage is seen directly across the capacitors. When a positive transition occurs

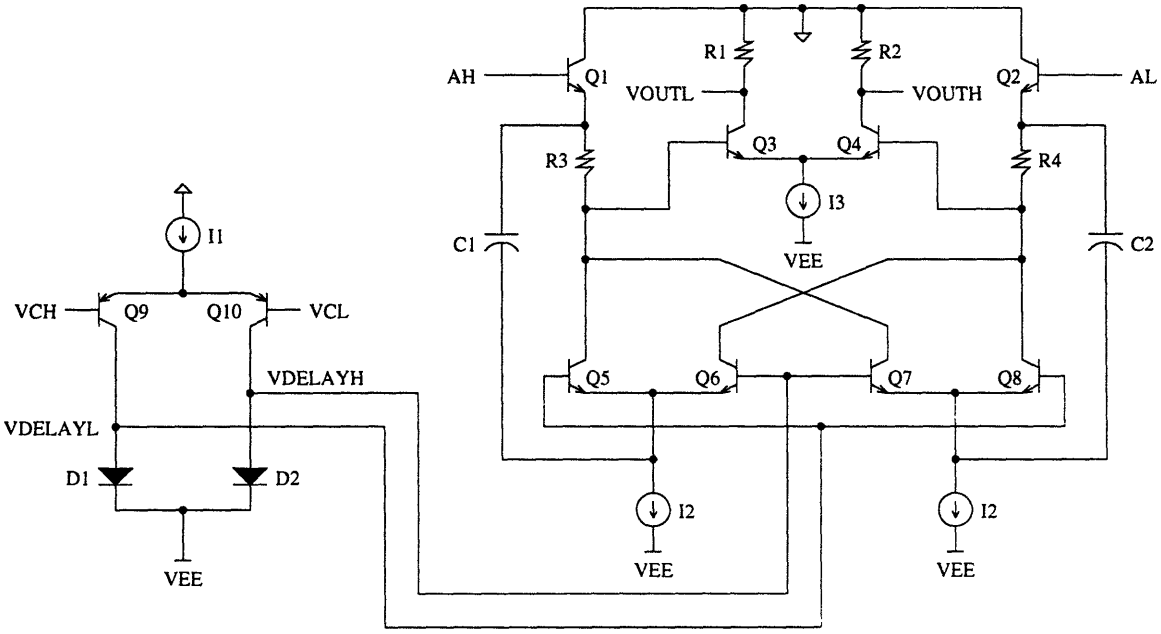


Figure 5-14: Variable delay cell schematic

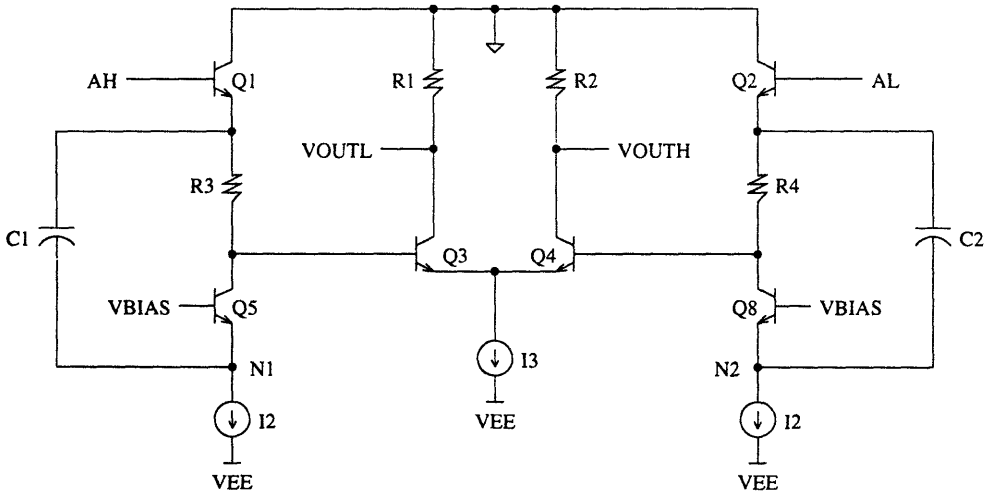


Figure 5-15: Variable delay cell in the up-state

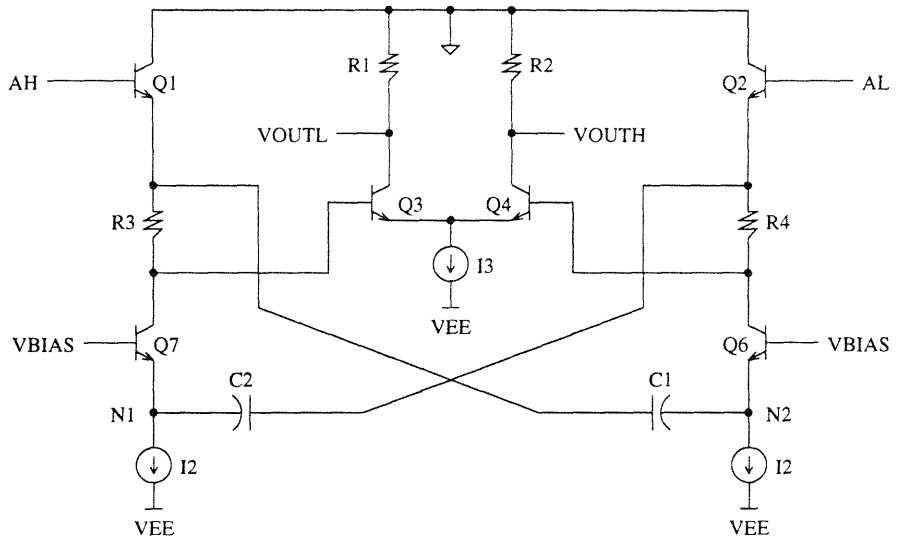


Figure 5-16: Variable delay cell in the down-state

on AH, a small current ΔI is injected into node N1 to charge capacitor C1. This ΔI flows through Q5 and reduces the voltage drop across R3, effectively "helping" the differential pair Q3-Q4 switching. At the same time, the negative transition which occurs on AL discharges C2 and pulls the same ΔI from node N2 and down through R4. This increases the voltage drop across R4 and also "helps" the differential pair Q3-Q4 switching. Since the circuit is symmetric the same "helping" action occurs when transitions of the opposite polarity occur.

Figure 5-16 shows the simplified equivalent variable delay cell when the current I1 is steered through Q6 and Q7. In this state transistors Q5 and Q8 are cut off, and Q6 and Q7 simply act as cascode transistors with their bases at a constant V_{bias}. Similar to the previous case, the bottom plates of capacitors C1 and C2 are fixed and the input voltage is seen directly across the capacitors. When a positive transition occurs on AH, current ΔI is pulled from node N2 to charge capacitor C1. This ΔI flows through Q6 and reduces the voltage drop across R4, which "hurts" or opposes the differential pair Q3-Q4 switching. At the same time, the negative transition which occurs on AL injects the same ΔI into node N1 and through R3. This reduces the voltage drop across R3 and also "hurts" the differential pair Q3-Q4 switching. Again, because of the circuit symmetry the same "hurting" action occurs when transitions of the opposite polarity occur.

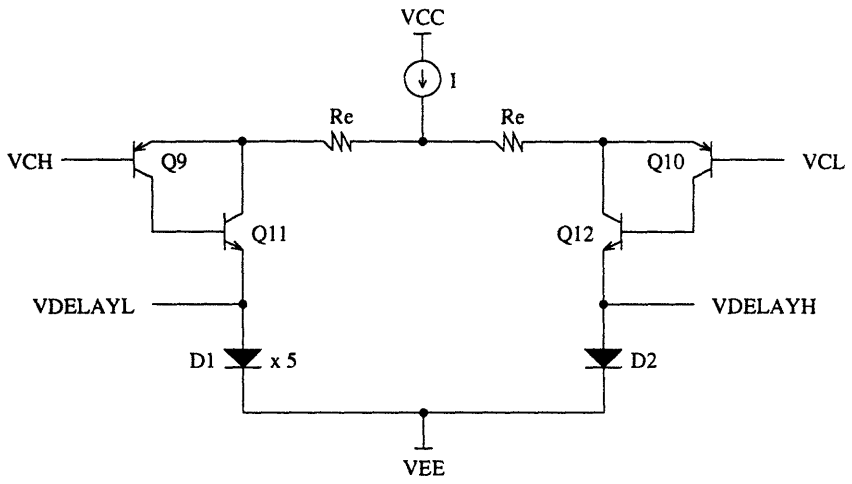


Figure 5-17: Modified VCO control voltage generator

As stated before, the variable delay cell has a very linear delay vs. voltage characteristic. This may seem good, but it actually does not give the desired VCO tuning characteristic. The delay vs. voltage characteristic can be modeled with the following equation

$$T_{pd} = T_{pdo} - K_t V_c \tag{5.5}$$

where T_{pdo} is the nominal propagation delay, K_t is the tuning "gain" factor and V_c is the control voltage. The frequency of the VCO is then

$$f_{vco} = \frac{1}{8(T_{pdo} - K_t V_c)} \tag{5.6}$$

which is far from a linear characteristic. Figure 5-17 shows includes a modification of the subcircuit that fixes this problem, along with other modifications. Diode D1 is five times larger than D2 which shifts the tuning characteristic so that there is a higher VCO gain at low V_c and a lower VCO gain at high V_c. This modification gives a very linear frequency vs. V_c characteristic. The emitter resistors R_E increase the linear input range to accept about ± 2 V, and transistors Q11 and Q12 improve the current handling of the input PNPs. The bandwidth of the VCO control voltage generator is about 50 MHz, due to the PNPs at the input. This high-frequency roll-off is desirable because it reduces the noise into the VCO which causes jitter.

The final simulated VCO tuning characteristic is shown in Figure 5-18. With

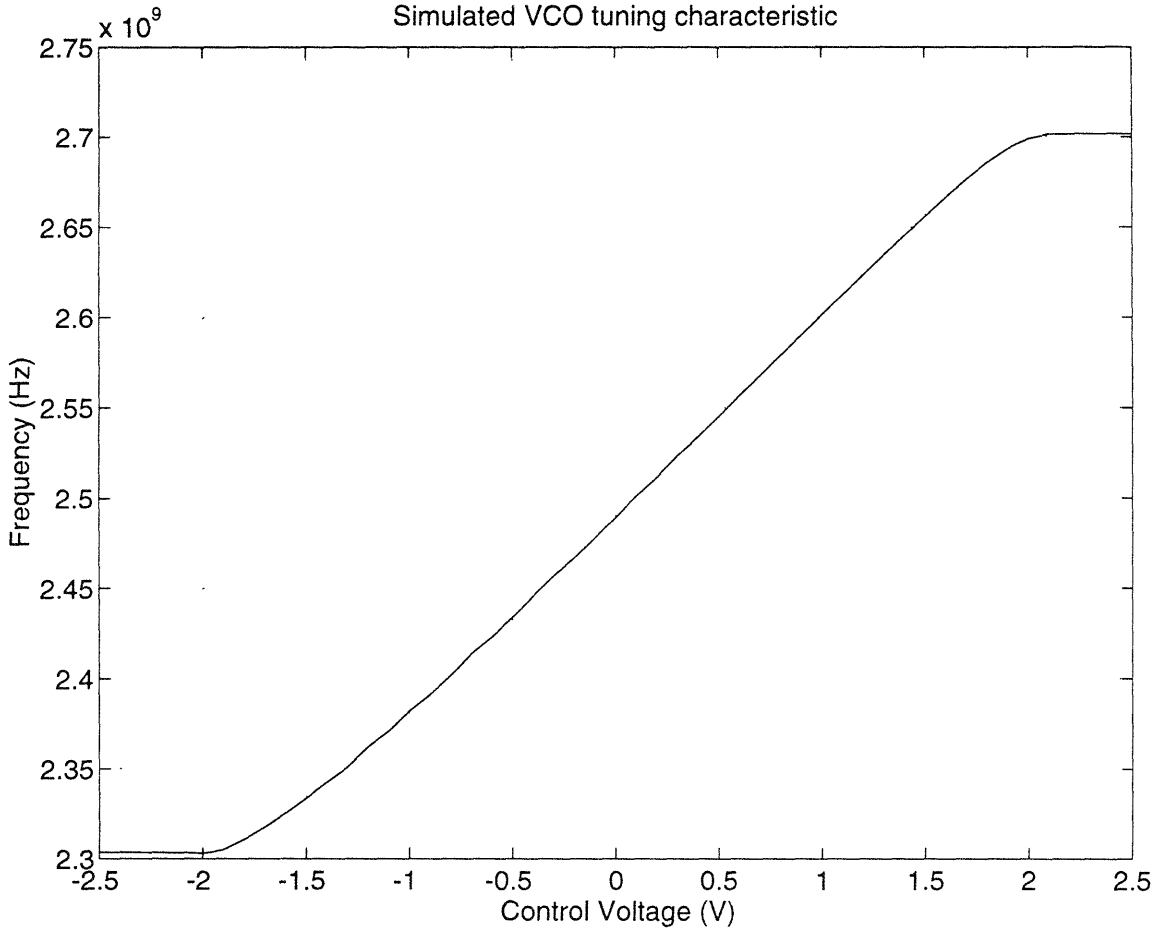


Figure 5-18: Simulated VCO tuning characteristic

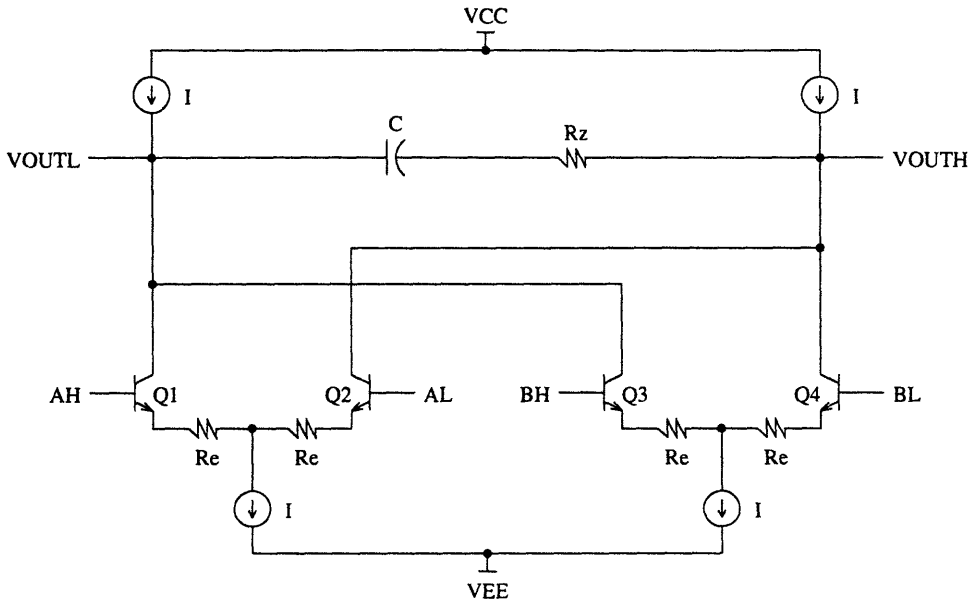


Figure 5-19: Basic circuit diagram for the charge pump

nominal process models, the VCO has a tuning range from 2.3 GHz to 2.7 GHz over a ±2 V steering voltage input. The resulting VCO gain is

$$K_o = 2\pi \frac{2.7GHz - 2.3GHz}{4V} = 628.3 \times 10^6 \frac{rad}{V} \tag{5.7}$$

5.5 Active Filter Design

The Filter block is implemented as an active, fully differential integrator or charge pump with an included zero in the transfer function. Figure 5-19 shows the basic charge pump circuit. There are two inputs to the charge pump, AH and AL from the IPD and BH and BL from the FD. All of the current sources are matched. To analyze the charge pump, first assume that the inputs AH, AL, BH, and BL are all balanced at the same potential—all of the transistors are conducting I/2, and there is no current flowing through the capacitor. Now if AH is slightly higher than AL, transistor Q1 carries I+ΔI current, Q2 carries I-ΔI current, and ΔI flows from right to left through the capacitor C, charging it such that the output voltage increases, effectively integrating the voltage across AH and AL. Similarly, if AL is slightly higher than AH, transistor Q2 carries I+ΔI current, Q1 carries I-ΔI current, and ΔI flows

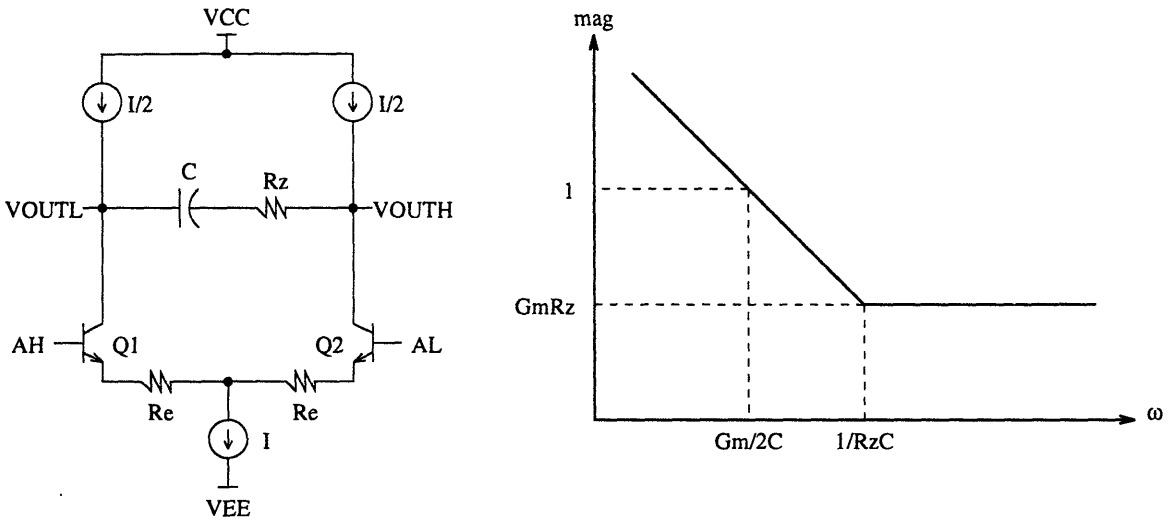


Figure 5-20: Simplified charge pump circuit

from left to right through the capacitor C, charging it such that the output voltage decreases, again integrating the voltage across AH and AL. The BH and BL inputs work in the same manner, and the differential currents are effectively summed with the differential currents due to the AH and AL inputs.

To further analyze this circuit, we will look only at one set of inputs, AH and AL, and assume that the other inputs, BH and BL are balanced and contribute no differential currents—this will be the case when the PLL is in frequency lock. In this case the simplified equivalent circuit is shown in Figure 5-20 The transfer function of the filter is given by the following equation

$$A_v = G_m Z = G_m \left(\frac{1}{2Cs} + \frac{1}{2} R_z \right) \tag{5.8}$$

where G_m is the transconductance of the current switches and Z is the load impedance seen at the collectors of the switching transistors. There is a factor of one-half on the impedance terms because the circuit is differential—from a single-ended point of view, the collector of Q1 sees half of the total differential impedance. The transconductance G_m can be calculated using half-circuit techniques to be

$$G_m = \frac{1}{\frac{1}{g_m} + R_e} = \frac{1}{\frac{V_T}{I_C} + R_e} \tag{5.9}$$

At low frequencies, when $s < 1/R_zC$, the impedance of the capacitor dominates, and

$$A_v = \frac{G_m}{2Cs} = \frac{1}{s\tau_1} \tag{5.10}$$

where τ_1 is the integrator time constant. This is the transfer function of an ideal integrator—a single pole at the origin, a -20 dB/decade slope, and a crossover frequency of

$$\omega_c = \frac{G_m}{2C} \tag{5.11}$$

At high frequencies, when $s > 1/R_zC$, the resistance R_z dominates and the transfer function simply becomes a constant.

$$A_v = \frac{1}{2}G_mR_z \tag{5.12}$$

The breakpoint in the curve is due to the zero in the transfer function which has a time constant of

$$\tau_2 = R_zC \tag{5.13}$$

The transfer function is shown in the graph in Figure 5-20.

Now we will explore the two major non-idealities of the charge pump. First, the charge pump has a limited DC gain due to the finite resistance seen at each of the output nodes VOUTH and VOUTL, due to the finite output resistance of the top current sources, the finite output resistance of the transistor switches, and the finite input resistance of the output buffer. This effect can be modelled as a single resistor R_o across the top current sources. At frequencies from $\omega = 1/R_oC$ down to DC, the impedance of the capacitor is greater than the output resistance which is effectively in parallel, thus the output resistance dominates. The maximum DC gain is then given by the following equation

$$A_{vo} = G_mR_o \tag{5.14}$$

where R_o is the output resistance at either node VOUTH or VOUTL. The effect of

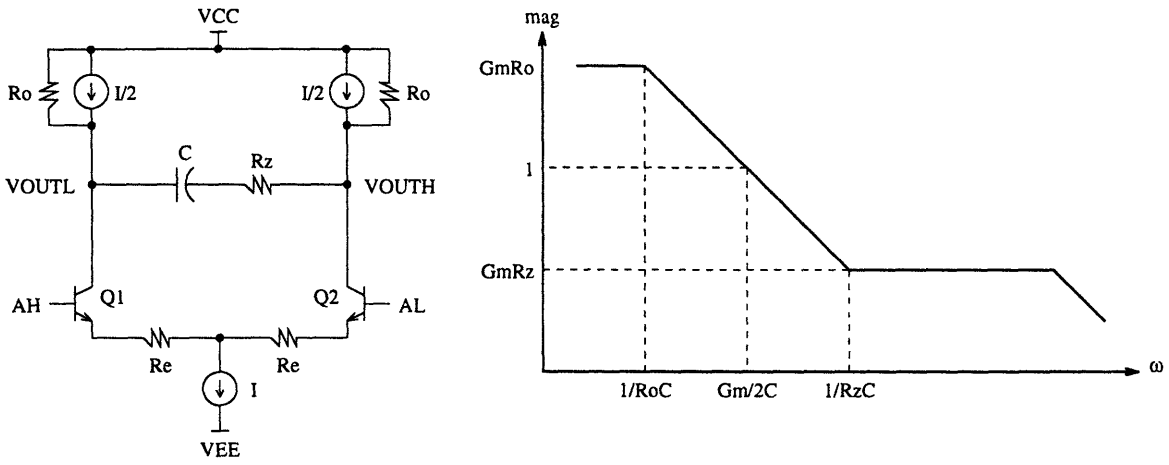


Figure 5-21: Simplified charge pump with output resistance

this non-ideality limits the minimum static phase error that can be obtained.

The second important non-ideality to consider is the finite bandwidth of the switching transistors. At very high frequencies, the bandwidth of the transistors due to C_π and C_μ limits the gain just as in any amplifier, and the transfer function begins to roll off. This introduces another pole to the overall PLL transfer function at ω_3. The resulting filter transfer function is

$$F(s) = \frac{s\tau_2 + 1}{s\tau_1(s + \omega_3)} \tag{5.15}$$

If this pole is located near the loop gain crossover frequency, the stability of the system is affected and it must be taken into account. It is recommended[12] that this pole be at least four times the crossover frequency, or $\omega_3 \geq 4\frac{\tau_2}{\tau_1}K_oK_d$. The high-frequency roll-off of the filter occurs at about 1 GHz, which is well above the loop gain crossover frequency of about 1.5 MHz. Many times additional poles are intentionally introduced into the system to roll-off the response earlier, helping to decrease the high-frequency noise.

In Chapter 4 the PLL parameters where calculated to be $\zeta = 5.18$ and $\omega_n = 7.5 \times 10^5$ rad/sec. Given $K_d = 200$ mV/rad and $K_o = 6.28 \times 10^8$ rad/sec as calculated in Sections 5.2 and 5.4 respectively, we can solve for the filter time constants using

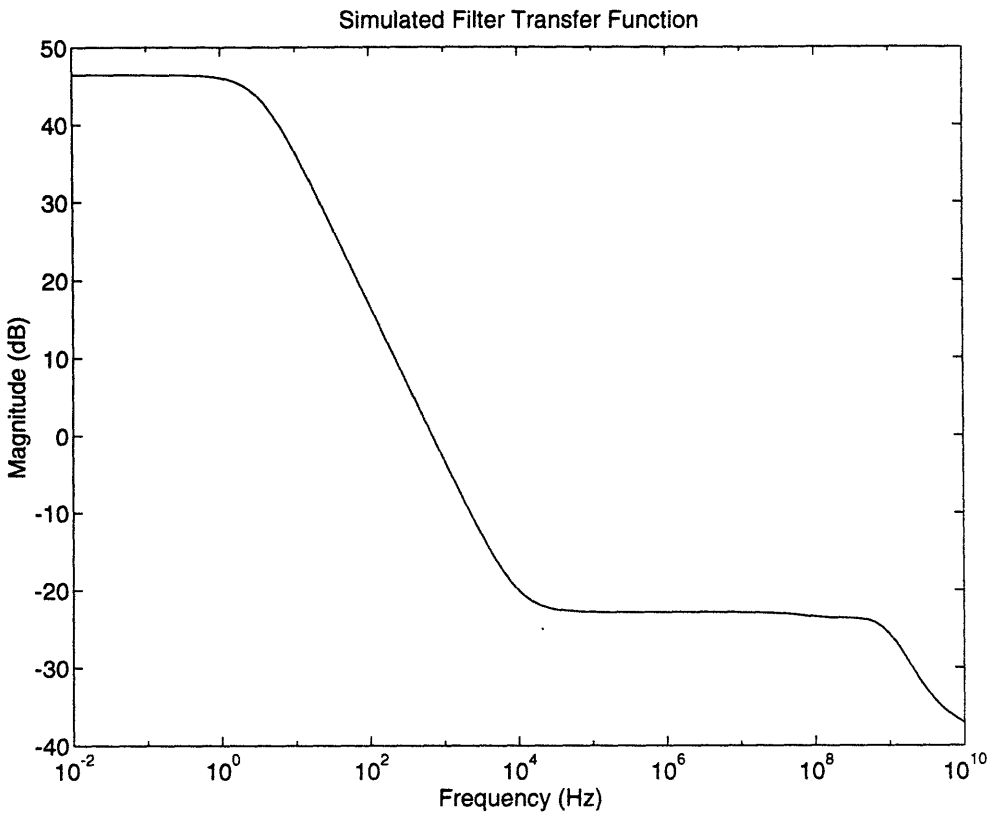


Figure 5-22: Simulated filter frequency response

Equations 3.15 and 3.16.

$$\tau_1 = \frac{K_o K_d}{\omega_n^2} = \frac{(6.28 \times 10^8)(0.2)}{(7.5 \times 10^5)^2} = 223\mu s \tag{5.16}$$

$$\tau_2 = \frac{2\zeta}{\omega_n} = \frac{2(5.18)}{7.5 \times 10^5} = 13.8\mu s \tag{5.17}$$

These time constants, which are simply RC products, are too large to integrate on chip. For example, to obtain the 223 μs required for τ_1, even integrating a huge 1 MΩ on chip would require a capacitor value of 223 pF, which is unreasonable. Selecting an external filter capacitance of 0.01 μF results in reasonable resistor values of $R_e = 9.87$ kΩ and $R_z = 1.38$ kΩ, using $I_C = 20\mu$A. For these values $g_m = 769\mu$S and $G_m = 89.5\mu$S.

The final simulated filter transfer function is shown in Figure 5-22. The simulated filter response and the calculated filter response match very closely. The maximum

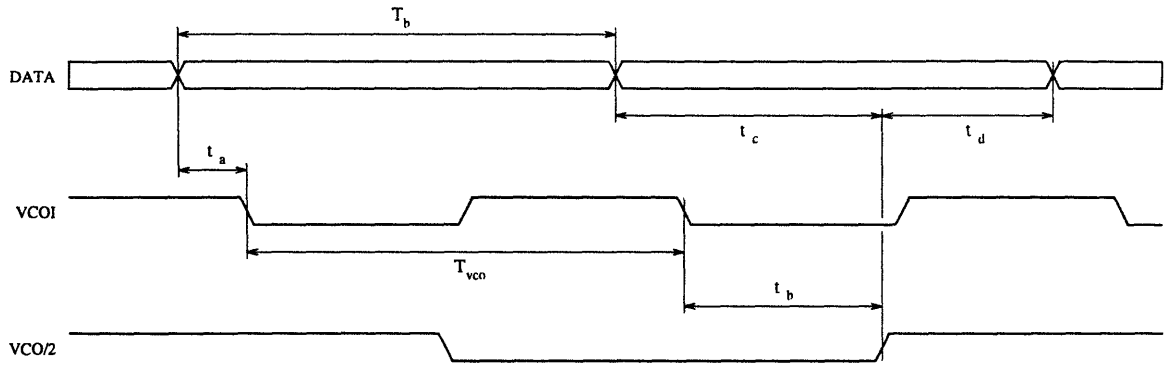
DC gain from the curve is 46.4 dB or 209, which corresponds to an effective $R_o = 2.34$ MΩ.

5.6 Data Re-timing

The first purely digital function after the clock recovery section is a flip-flop that re-times the data and regenerates it up to digital levels. Simulations must be performed to insure that the VCO waveform is centered in the data eye and that the data will be re-timed properly.

The flip-flop is tested to find how much the VCO waveform (divided by 2) can vary from the center position and still correctly re-time the data. In this SPICE test, both the clock and data waveforms are 1.24416 GHz sinusoids, which represents a somewhat "worst case" shaped waveform and a data stream of a repeated 1010 sequence. The swept parameter in the simulation is DELAY, which is the amount of delay between the rising edge of the clock (which is fixed) and the data. The data is properly re-timed when it is swept from 32 ps through 385 ps. "Good" here means that there is at least a 100 mV swing at the output of the flip-flop over the up/nom/down models. This simulation shows that the clock edge can vary approximately ±176 ps from the center of the bit period and still re-time the data properly.

Figure 5-23 shows the timing diagram of the data re-timing signals. To begin with, the top trace is the 2.48832 Gb/s data waveform which has a bit period T_b of approximately 402 ps. When in phase and frequency lock, the phase detector will place the falling edge of the VCO in-phase waveform (the second trace) exactly at edge of each bit period. Unfortunately, there will be some phase detector induced static phase error, t_a, so that the VCOI waveform will lag about 165 ps from the ideal time. The VCOI waveform is divided by two, which is shown as the third trace as VCO/2. The VCO divider has a CLK-Q time of about 183 ps, which is about half of a bit period. At this point it is shown that the minimum times t_c and t_d—the data transition to clock edge and clock edge to data transition minimum times respectively—are easily kept within the limits.



1/19/94

	Description	min	nom	max	units
T_b	Bit period		402		ps
T_{vco}	VCO waveform period		402		ps
t_a	Phase Detector induced static phase error		60		ps
t_b	Div 2 clock to Q delay	157*	183	216*	ps
t_c	Minimum required setup time	16			ps
t_d	Minimum required hold time	32			ps

Figure 5-23: Timing diagram of data re-timing.

One further thing that must be calculated is the resulting jitter tolerance of the data re-timing flip-flops. The jitter tolerance based on the worst-case setup and hold times are easily calculated. The maximum setup jitter tolerance Φ_{su} is equal to the static phase error t_a, plus the minimum Div2 clk-Q delay t_b, minus the required setup time t_c.

$$\begin{aligned} \Phi_{su} &= t_a - \frac{T_{vco}}{2} + t_{bmin} - t_c \\ &= 60\,ps + 157\,ps - 16\,ps = 201\,ps \\ &= \pi\ rad = 180\ deg \end{aligned} \tag{5.18}$$

The maximum hold jitter tolerance Φ_h is equal to the bit period T_b, minus the static phase error t_a, minus the maximum Div2 clk-Q delay t_b, minus the required hold time t_d.

$$\begin{aligned}\Phi_h &= T_b - t_a + \frac{T_{vco}}{2} - t_{bmax} - t_d \\ &= 402ps - 60ps - 216ps - 32ps = 94ps \\ &= 0.47\pi\ rad = 84\ deg \end{aligned} \tag{5.19}$$

This lesser of these two results constrains the peak jitter tolerance, which turns out to be $\Phi_{su} = 94$ ps. The peak-to-peak jitter tolerance is twice this value, or 188 ps, 0.94π radians, or 168 degrees. The SONET spec calls for a high-frequency (above 1 MHz modulation frequency) jitter tolerance of 0.15 UI, which is 0.9425 radians or 60.28 ps peak-to-peak jitter. This level of jitter tolerance is easily obtainable as long these calculated numbers are valid. It must be kept in mind that these calculations assume that no noise generators within the receiver are significantly contributing to the jitter at the input. To improve this estimate, the VCO jitter should be taken into account, as well as the jitter from the divider and the flip-flop.

Chapter 6

Signal Detect Circuit Design

The Signal Detect block asserts the Loss-Of-Signal Indicator (LOSI) line if there are no transitions in the serial input data stream within a certain period of time, or if the appropriate loss-of-power line is asserted—either the External Loss-Of-Signal (ELOP) or Loss-Of-Power Indicator (LOPI) depending on if the internal post-amp is used. Also, if the Clock Recovery block is bypassed the Signal Detect output is held high in order to enable the RCLKI line. This block is designed to fulfill the SONET requirement in Section B.1.1—a "blank" period anywhere from 2.3μs to 100μs must trigger a LOS condition. The required functionality is relatively low-speed, simple and easy to build with the Tektronix GST-1 digital standard cells.

The Signal Detect block is implemented as a 9-bit ripple counter which is clocked by the 77.76 MHz reference clock. The Signal Detect block diagram is shown in Figure 6-1. The ripple counter is a series of 9 divide-by-2 flip-flop stages. The last flip-flop is not used as a counter—it simply detects a rising edge from the previous stage and triggers the LOSI line—therefore it does not contribute to the overall terminal count. The Set/Reset Generator asserts a set/reset signal when there is a transition on the SDI line—this sets the first 8 counter flip-flops and resets the last flip-flop. If the terminal count of 256 is reached before a reset pulse is generated, the rising edge on the 8th flip-flop toggles the last flip-flop, which sets the LOSI line. Since the 77.76 MHz reference clock has a period of 12.86 ns, the LOSI line is asserted 3.3μs after the set/reset line is cleared.

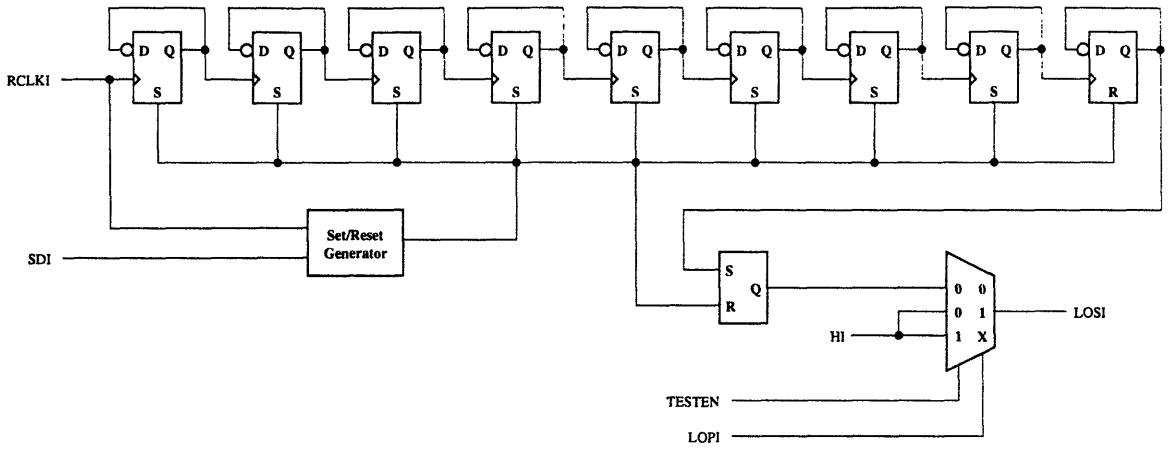


Figure 6-1: Block Diagram of the Signal Detect Block

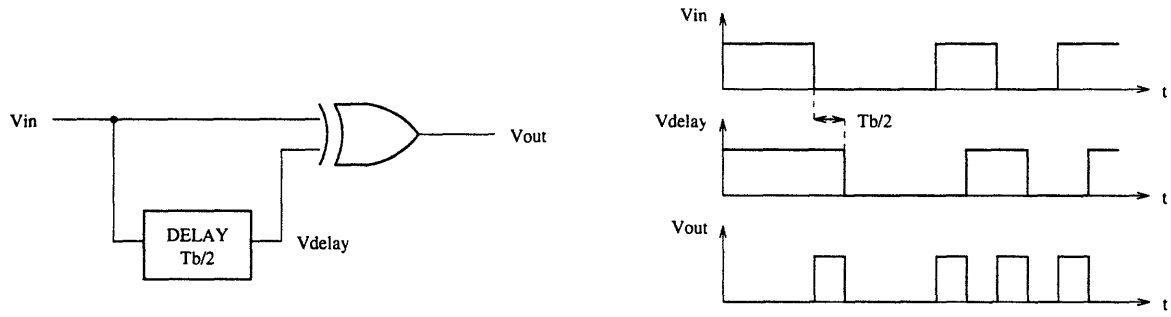


Figure 6-2: Delay-XOR circuit and timing diagram

The 2:1 multiplexer selects either the LOPI or ELOP line, depending on the state of BPPAE. The 3:1 multiplexer is used as a complex AND-OR logic gate. When TESTEN is asserted it overrides all other inputs to the mux and forces LOSI high. If TESTEN is low and the selected loss-of-power line is asserted, the counter input to the mux is overridden and LOSI is forced high. Only when both the TESTEN line and the selected loss-of-power line are low does the counter input signal propagate to the LOSI line.

The set/reset pulse generator behaves somewhat like a differentiator, which is typically realized as a delay-XOR in many circuits. See Figure 6-2 for a diagram of a delay-XOR circuit and its associated timing diagram. The delay block has a delay of half a bit period or $T_b/2$. The data input at V_{in}, when XORed with the delayed data at V_{delay} produces a pulse of width $T_b/2$ at every transition of the input data. This functionality is exactly what is needed for the set/reset generator, but unfortunately

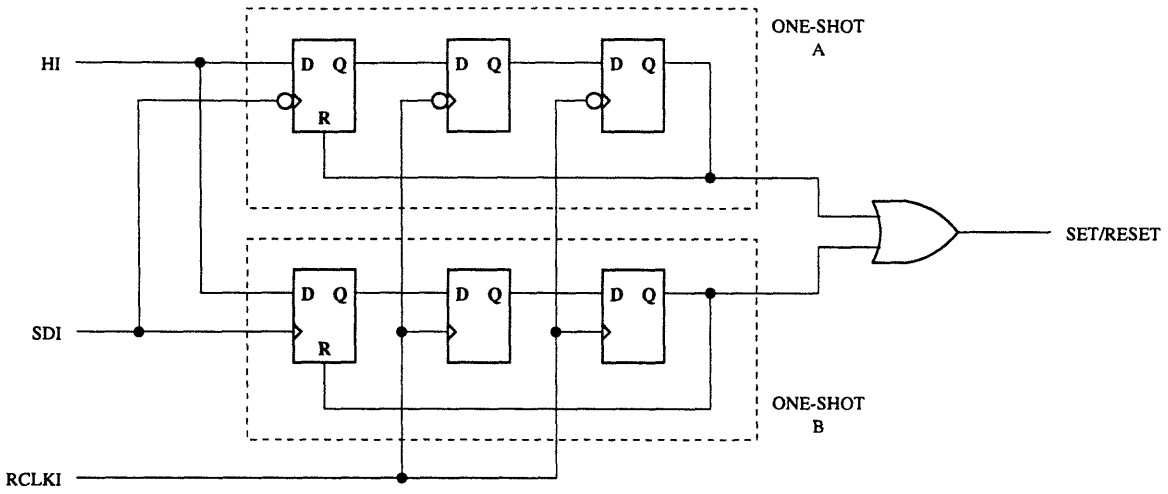


Figure 6-3: Block Diagram of the Set/Reset Generator

it is a high-speed circuit which introduces two problems. The GST-1 standard cells are not capable of performing an XOR function with an output signal of 2.48832 GHz, and also the low-speed counter cannot be reset with such short pulses. Upon closer inspection of the signal detect block, the set/reset pulses do not need to be generated at every transition of the input data. The only constraint is that the spacing between set/reset pulses is shorter than the LOS countdown time.

A new design for the set/reset pulse generator is shown in Figure 6-3. This design is based on two one-shots which are low-speed devices. The upper three flip-flops in the diagram make up one-shot "A" which is triggered by a negative edge on SDI. Similarly, the lower three flip-flops make up one-shot "B" which is triggered by a positive edge on SDI. The first flip-flop in each one-shot is the edge sensitive device, and the other two flip-flops are used as delay elements and also to resynchronize the reset pulse and reduce chances of metastability on the set/reset line.

Refer to the one-shot "B" in Figure 6-3. Assume for a moment that all three flip-flops are reset to "0". When a positive edge occurs on the SDI line, the first flip-flop loads in a "1" due to the high level on the flip-flop's data input. On the next two RCLK cycles, this "1" is propagated down through the next two flip-flops—on the third RCLK cycle the "1" triggers the set/reset line. The same signal that triggers the set/reset also resets the first flip-flop to "0". In two RCLK cycles this

"0" propagates down through the flip-flops and clears the set/reset line to "0". Thus the set/reset pulse is a minimum of 2 RCLK cycles long, or 25.72 ns. If another SDI positive edge occurs after the first flip-flop is reset to "0" but before the next RCLK positive edge, the first flip-flop is set to "1" once again and the set/reset pulse is extended for another 2 RCLK cycles. The upper one-shot "A" functions in the same way except that it is triggered off of a negative edge on the SDI line—in this way, either a rising or falling transition of SDI clears the Signal Detect block.

This Signal Detect block implements all of the necessary functionality with relatively low-speed and low-power digital cells.

Chapter 7

Frame Detect and 1:16 Demux Design

The receiver uses the A1 and A2 framing bytes to align the serial data on proper byte and frame boundaries. In an STS-N signal, there will be NxA1 bytes followed by NxA2 bytes that mark the beginning of the frame. Because of the interleaving format for higher rate signals, the boundary between the A1 and A2 bytes always occurs at the same time relative to the beginning of an STS-N frame. An effective way to perform byte and frame alignment is to search for this A1-A2 boundary and gives a frame pulse signal (FP) when it is detected.

This receiver detects an STS-48 frame, so the frame begins with 48 A1 bytes followed by 48 A2 bytes. The most straight forward and simple-minded implementation might be to check all of the frame bytes. A fully parallel implementation would use a shift register that is 2x48 bytes long (768 bits long) with a huge amount of parallel decoding logic. A fully serial implementation would use a 10-bit state machine which checks all 768 bits, one bit at a time. While the parallel implementation is excessively large in die area and power, both schemes are impractical due to speed and worst case bit error rates (BER). In terms of bit rate problems, if the BER is higher than 1:768 (about 1.3×10^{-3}) then statistically a frame boundary will never be detected. A typical worst case BER is 10^{-3}, so a value of 1.3×10^{-3} is nearly as great as the worst case BER.

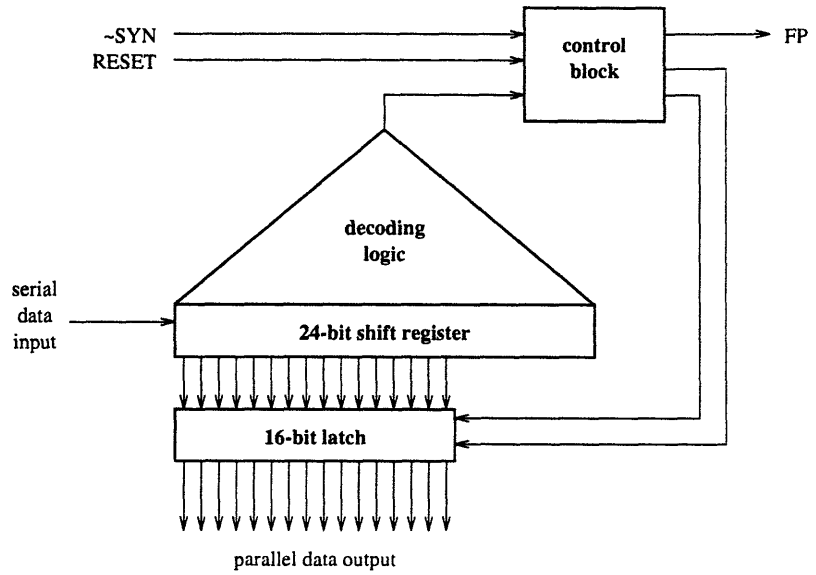


Figure 7-1: Basic parallel frame detect diagram

Not all of the bytes need to be decoded—only the *transition* between the A1 and A2 bytes is needed for synchronization. In the SONET framing scheme, only one frame pulse needs to be detected in order resynchronize the frames, and when synchronized, four invalid or missed frame pulses constitutes a loss of frame. Enough bits should be decoded to insure that the A1-A2 transition is not just a false alarm (possibly due to "random" data transmitted in the payload which happens to contain a framing pattern), but too many bits can prevent rapid synchronization due to high BERs. In his paper, Kong [3] states that searching for just the three byte sequence such as A1A1A2 gives a false alarm probability of 2^{-24}.

7.1 Shift and Compare

The Frame Detect block is a relatively high-speed, parallel/pipelined implementation. This block simply shifts bits through the 24 bit shift register and compares the bits to a hardwired A1A1A2 pattern in the decoding section. See Figure 7-1 for a general block diagram of the parallel frame detect block. As mentioned before, the decode process is pipelined because of the relatively large gate delay compared to the clock period. The pipeline stages are set up so that FP will occur on the byte boundary

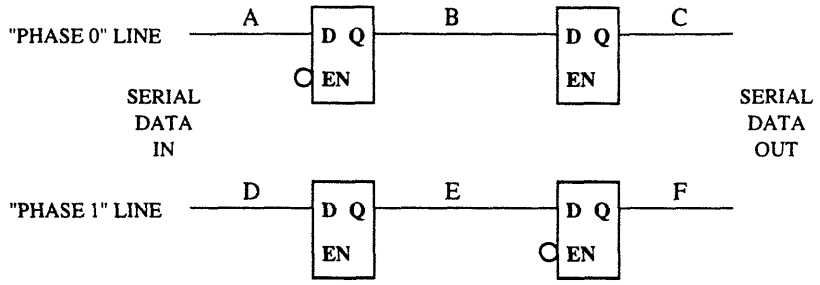


Figure 7-2: Simplified two bit shift register cell

precisely one byte after an A1A1A2 sequence has occurred.

For the 24-bit shift register, 24 of these flip latches can be cascaded. (See Appendix D) Unfortunately, the decoding logic needed to detect the A1A1A2 framing sequence is now doubled because there are essentially two separate serial lines to decode now, either of which can hold the "even" bits or the "odd" bits. As we will see, though, the decoding logic can also be half-speed and half-power.

Before discussing the decoding logic, the order of bits along the shift register must be explained further. Figure 7-2 shows the 2-bit cell that is used as a subcircuit to construct the shift register. This block can be cascaded twelve times to make up the 24-bit shift register (the data lines on the first 2-bit cell must be connected together). The data lines have been "untwisted" and the clock line has been left off for simplicity—the bubble on the clock inputs indicates the phase of the latch. It is essential to remember that the clock signal is a "half-speed" clock (relative to a flip-flop clock at the same data rate). The signal flow down the lines can be thought of exactly like a string of flip flops, but one of the lines has the opposite phase. For this reason, the top string of latches is labelled "phase 0" and the bottom string of latches is labelled "phase 1." With the clock running at half speed and the first 2-bit cell input data lines tied together, the incoming data bits are sampled back and forth between the two phase strings. Depending on the phase of the clock, the even bits are on the phase 0 line and the odd bits are on the phase 1 line, or vice versa.

At any one instant in time, a single bit of data is "visible" at two points on a string because of the "master-slave" action of the latches. Think of the 2-bit cell in Figure 7-2 as one of the middle cells in the cascaded string. If the phase 0 line

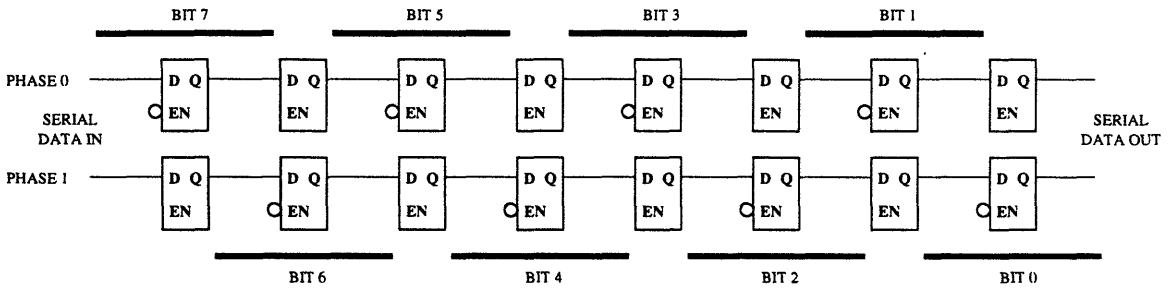


Figure 7-3: Bit positions along the shift register at a given point in time

sampled the first incoming serial bit, then that bit at some later point in time will be visible at points B and C, and the second incoming bit at that same point in time will be visible at points D and E. Alternatively, if the phase 1 line sampled the first incoming serial bit, then that bit at some later point in time will be visible at points E and F, and the second incoming bit at that same point in time will be visible at points A and B. Figure 7-3 shows the positions of bits at some instant in time for a longer string of two bit cells. In this figure, the first incoming serial bit was latched by the phase 1 line—the odd bits are on phase 0, and the even bits are on phase 1. We can also see that the inverted clock latches are holding data, and the non-inverted clock latches are transparent, so the clock signal must be high. Notice that each bit of data is "visible" at two points along the string. Now, on the next clock edge (clock falling edge), the data bits in each string will simultaneously shift one latch to the right. While it has been shown that a single bit is "visible" at two points in a string at one instant in time, it is also true that a single bit is "visible" at a single point in a string for two consecutive clock phases. Both of these facts are important in understanding the proper operation of the frame detector.

Now we proceed with the decoding discussion. In order to correctly decode the 24-bit framing sequence, both the phase 0 and phase 1 lines must be decoded together. Before the framing sequence is found, it is not known whether the even bits are on phase 0 or phase 1, so the decoding logic must be doubled in order to check the phase of the clock in reference to the incoming bits. It should be pointed out here that this decoding scheme must check for two different framing patterns—one "real" pattern and also a "false" pattern which has the even and odd bits interleaved—both of which

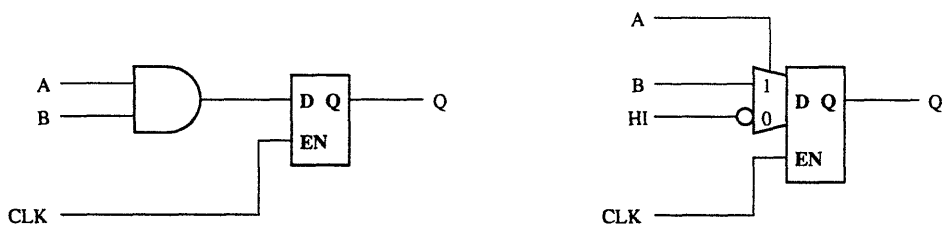


Figure 7-4: Equivalent functionality (a) 2-input AND with latch (b) 2:1 mux incorporated into latch cell

appear to be a valid framing sequence to the frame detector. Thus, the probability of getting a false frame is doubled, and is now at 2^{-23}.

The decoding logic is simply a pipelined tree of AND gates. Using the standard cell library, the clock period is too short to incorporate a separate AND gate between two pipeline stages. Up to 3 levels of logic can be used in a standard cell ECL gate, which means that a simple latch can accomodate one additional level of logic. This means that a 2:1 multiplexer can be included within each latch, and less logic needs to be implemented outside of these circuits. This implementation actually gives a number of benefits. A 2:1 mux can be used as a universal two-input logic gate—any two-input logic function can be performed by a 2:1 mux, such as an OR gate, AND gate, XOR gate, etc. Figure 7-4 shows two different implementations of a AND and latch combination, both are functionally equivalent. Additionally, there is also both a speed and power savings associated with this implementation. If a mux and a latch are designed separately, two logic trees and four emitter follower stages are needed, but designed together, only one tree and two emitter followers are needed, saving half the power and nearly half the propagation delay.

The first pipelined decoding logic stage is contained in the 2-bit cell as shown in Figure 7-5. If the decoding mux is placed across points B and E to detect a 2-bit pattern, the decision circuit must be high-speed—the latches before points B and E are clocked on opposite phases, so the overlapping "valid" data time is only 402 ps. Also, the odd/even bit phase information is lost in this configuration—the decision circuit does not know if the bit at point B lags or leads the bit at point E. A change in mux placement as shown in Figure 7-5 fixes these problems. Because each bit of data

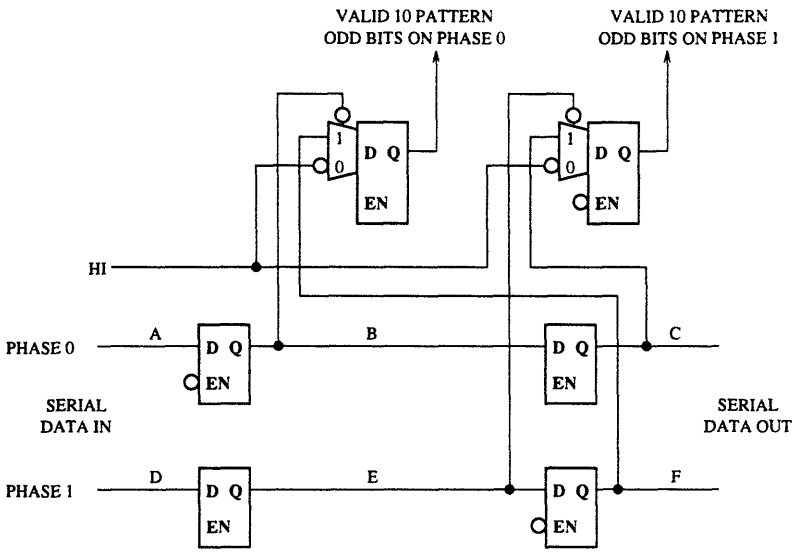


Figure 7-5: A 2-bit shift cell with 10 decoding logic

is "visible" at two points along a string, the mux can be placed across points B and F, or similarly across points E and C. The latches before these pairs of points have the same clock phase and the decision circuit has 804 ps. The odd/even bit phase information is retained—if the valid pattern appears across points B and F, then F is the leading bit and the odd bits are on phase 0. If the valid pattern appears across points E and C, then C is the leading bit, and the odd bits are on phase 1. Notice again that there are two valid pattern outputs, one for each phase, both are clocked on phases opposite to the preceding latch. Each pipelined decoding stage must have opposite clock phase latches in order for the signals to propagate correctly.

Since all of the standard cells used here have fully differential inputs and outputs, pattern detection is simplified by hardwiring the mux inputs either inverted or non-inverted by changing switching the leads. Four cells were created that are hardwired to detect the four possible 2-bit patterns, binary 00, 01, 10, or 11. The most significant bit of a serial byte is transmitted first, so in Figure 7-5, the most significant bit is non-inverted, and the next bit is inverted, so the valid outputs will detect a binary 10 serial data input. The shift register and first level of pipelined decoding logic is simply a selection of the appropriate decoding cells. The SONET framing byte A1 detector—the A1 byte is binary 11110110—is a cascade of a 01 cell, a 10 cell, a 11

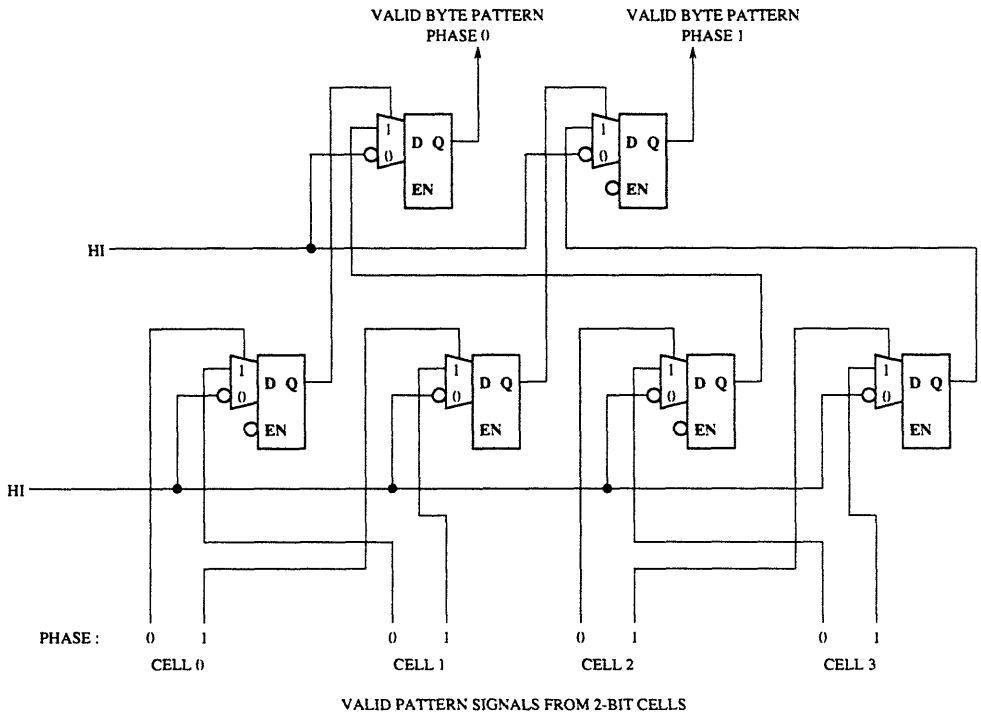


Figure 7-6: The byte compare block

cell, and a 11 cell.

The next stage of pipeline decoding is a byte compare block, shown in Figure 7-6. When valid 2-bit patterns appear either on all of the phase 0 or all of the phase 1 inputs to the byte compare block, the corresponding phase valid byte pattern output will go high one clock cycle later. Notice that the input latches have opposite clock phases with respect to the 2-bit cell output latch clock phases.

The last stage of decoding logic, the final compare block, is similar to the byte compare block except that there are only three pairs of inputs. See Figure 7-7 for a diagram of the final compare block.

7.2 1:16 Demultiplexer

The 1:16 demultiplexer is simply a 16-bit latch that takes data off of the 24-bit shift register at the appropriate time. Both the architecture and timing are closely tied with the shift register.

The serial data in the shift register is separated with the even bits on one string

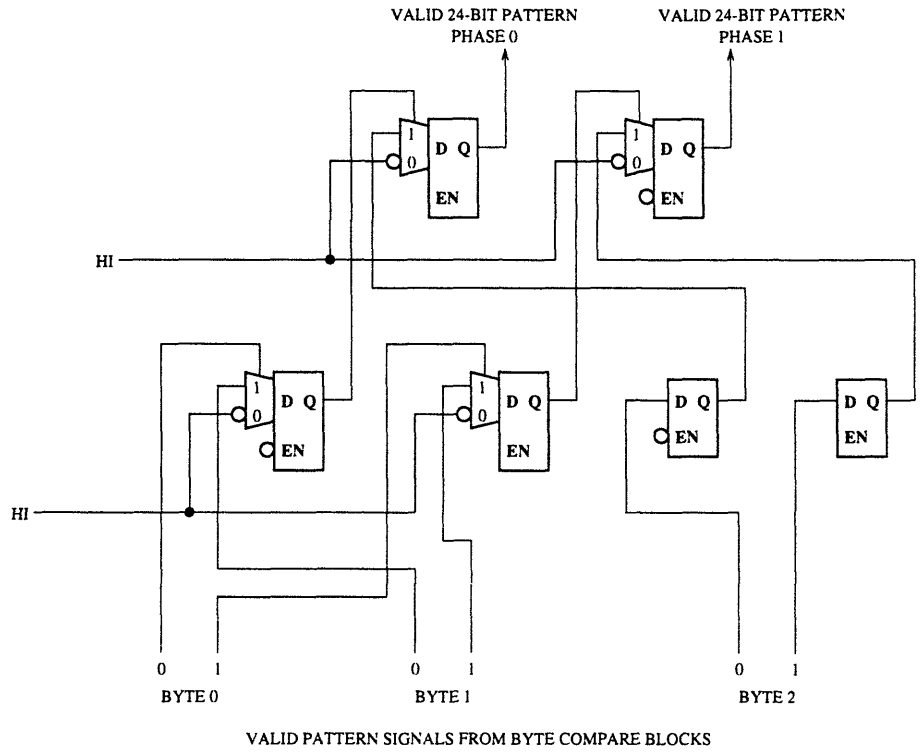


Figure 7-7: The final compare block

and the odd bits on the other string. Depending on the phase of the clock relative to the first bit sampled, the 16-bit latch must interleave the even and odd bits. To accomplish this, a 2:1 mux-latch selects which string to take that particular output bit from. Refer to Figure 7-8. When SELECT is low, the LSB is taken directly off of phase 0 (point B) and the MSB is taken directly off of phase 1 (point F). Similarly, when SELECT is high, the LSB is taken off of phase 1 (point E) and the MSB is taken off of phase 0 (point C), but in this case through a latch which delays the data by one half-cycle—the data at points G and H are the same clock phase as the data at points B and F. Since the "select" mux-latch is enabled from a non-inverted clock, the output of this mux-latch is updated when it goes transparent on every rising edge of the clock. The mux-latches at the bottom of Figure 7-8 serve as internal holding registers. These mux-latches are enabled from an inverted clock, so that they are latched when the clock line is high. When the STROBE signal is low the Q output is fed back into the mux 0 input, holding the data. To write a bit of data the STROBE signal must be raised high to select the mux 1 input while the clock is low, making

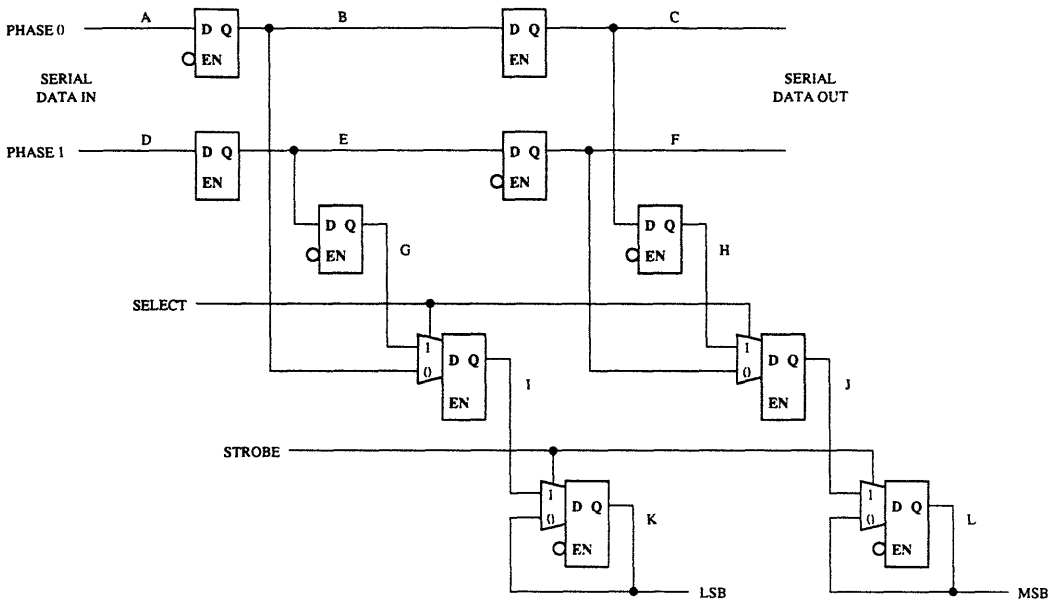
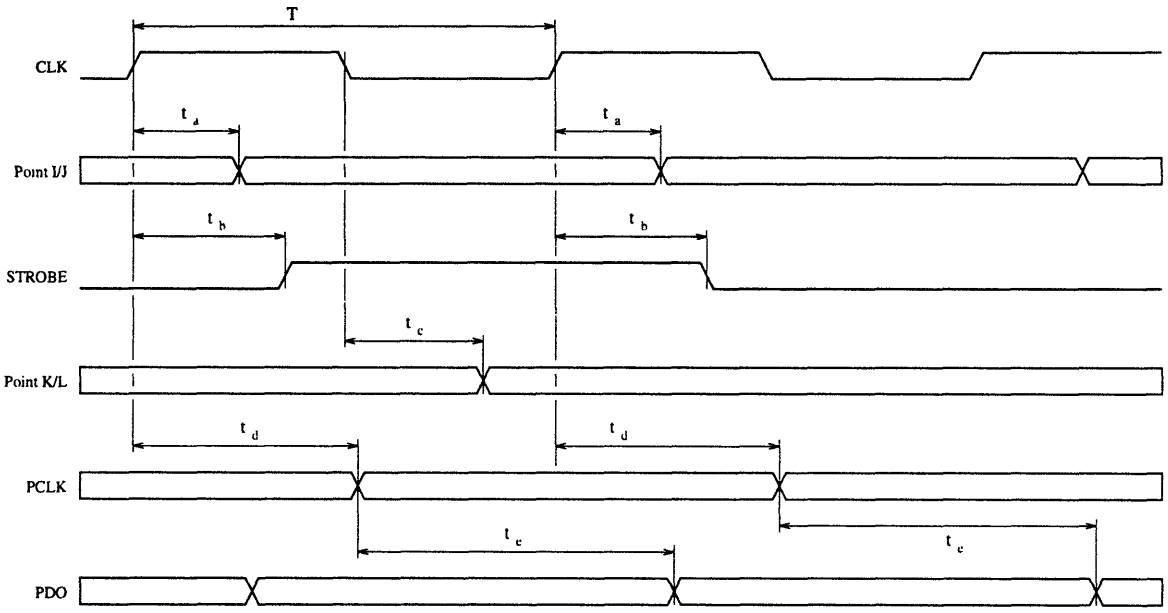


Figure 7-8: The parallel output circuitry attached to a 2-bit shift cell

the latch transparent, then either the STROBE or the clock lines can latch the data. If the STROBE signal goes low, the mux 0 input is selected and the data is held regardless of the clock input. If the clock goes high, the mux-latch latches the input data. The output of these mux-latches are updated on every falling edge of the clock signal.

Because of the implementation of the mux-latch holding register, the STROBE signal must overlap the low half-cycle of the clock input by a certain amount. The timing diagram is shown in Figure 7-9. In this timing diagram, the top trace is the clock signal, the second trace is the output of the SELECT mux-latch, and the third trace is the STROBE signal. The data output of the SELECT mux-latch is triggered by the rising edge of the clock signal, and time $t_a = 341$ ps. The STROBE signal must overlap the low half-cycle clock when the data is valid—the time from the falling edge of the clock to STROBE high is $t_b = 495$ ps and the clock to data time for the output of the 2-bit cell is $t_c = 404$ ps. The CLK to PCLKO time is $t_d = 606$ ps and the PCLKO to PDO time is $t_e = 879$ ps.



1/6/94

	Description	min	nom	max	units
T	CLK period		804		ps
t_a	CLK to valid data at points I/J	148		260	ps
t_b	CLK to STROBE delay	192		373	ps
t_c	CLK to valid data at points K/L	161		357	ps
t_d	CLK to PCLK delay	275		576	ps
t_e	PCLK to PDO delay	345		856	ps

Figure 7-9: STROBE timing diagram

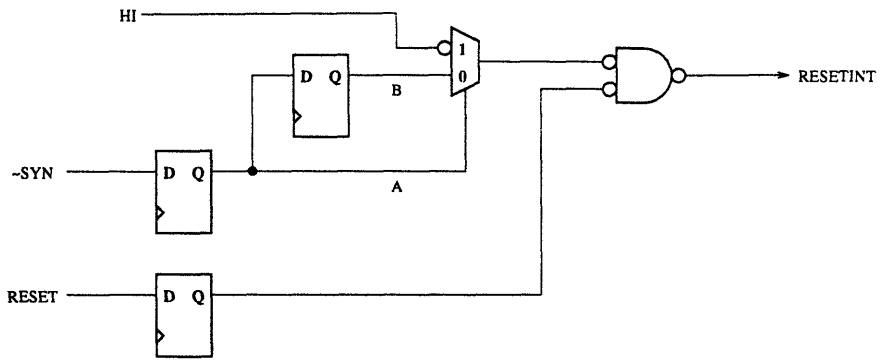


Figure 7-10: Control block reset logic

7.3 Control Block

The control logic uses the phase 0 and phase 1 signals to control the timing of the 16-bit output latch, the timing of the clock output, and the FP output signal.

The frame detect block is reset by either a falling edge of ~SYN or RESET asserted high. Figure 7-10 shows the reset logic that generates the internal reset signal. Both the ~SYN and RESET signals are routed through synchronizing flip-flops. The delay-XOR is a common technique which creates a single pulse from a single edge transition, which is similar to what is needed here. The only difference is that the delay-XOR gives a pulse output for both rising and falling edges, and the control circuitry should be reset only on a falling edge. When ~SYN is asserted (goes low) the frame detect block will be reset and begins looking for a valid framing pattern. Later, the ~SYN signal is deasserted (goes high) after a framing pattern has been found and received by the downstream external circuitry. If resets occur on both the rising and falling edges of ~SYN, this would once again reset the frame detect block and reframing would occur.

Instead of an XOR function, a custom function is wired up using a mux. With an XOR, any difference in the data between points A and B (See Figure 7-10) result in a pulse output from the XOR. In order to catch only the falling edge, the custom function must give a pulse output when A is low and B is high. The mux in the figure is wired up to give this output characteristic. Finally this pulse and the RESET line are ORed together to form the final internal reset signal, RESETINT.

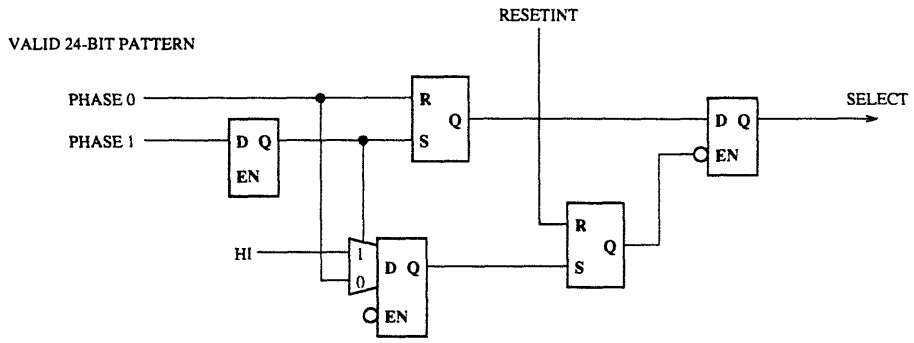


Figure 7-11: Control block select logic

Figure 7-11 shows the portion of the control logic that selects the interleave mode of the 16-bit output latch. The phase 0 and phase 1 input lines come from the decoding logic section. They are normally low, but when a valid framing pattern has been found, either the phase 0 or phase 1 signal will pulse high for one full clock cycle. Because there is a finite chance of a misframe due to a framing pattern in other portions of the SONET frame, the control block will select the bit interleaving mode of the output on only the first phase signal that arrives after a RESETINT signal. Any subsequent phase signals will be ignored so that the frame or byte alignment will not be altered.

As with the 16-bit parallel output circuitry, the phase 1 valid pattern signal is delayed one half-cycle so that the following decision circuits can be low-speed devices. The mux-latch in the diagram is wired to perform an OR function on the phase 0 and phase 1 lines. Since the output from the phase 0 and phase 1 lines is a pulse, the output of the mux-latch is also a pulse, indicating that a valid framing pattern has been found. The lower-right RS-latch, which is set by the OR mux-latch and reset by the internal reset line, "remembers" that a framing pattern has been found since the last RESETINT signal. The upper-left RS-latch holds the value of the phase pulse that occured last. A phase 0 pulse resets the latch and a phase 1 pulse sets the latch.

When a phase pulse occurs, the phase value is held on the upper-left RS-latch and the output latch is transparent. The output signal is called SELECT because it selects the demux bit interleaving mode. On the next clock half-cycle, the output latch enable is brought low and the SELECT value is held. The SELECT value is

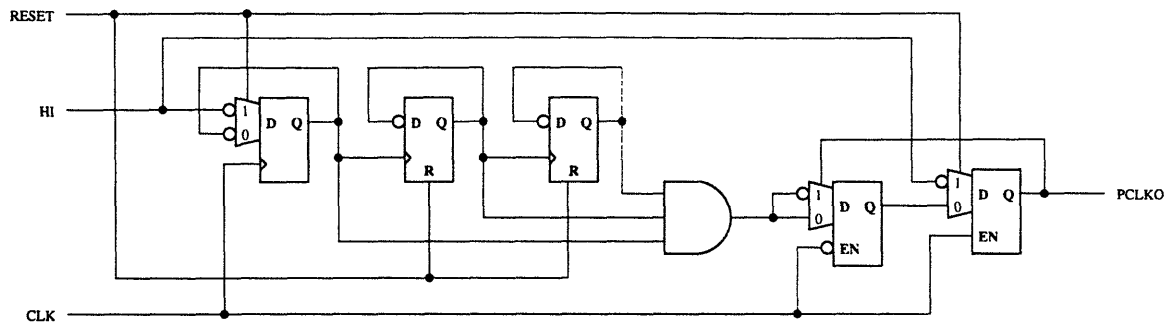


Figure 7-12: Diagram of the DIV16 divider

held until RESETINT is asserted, which clears the lower-right RS-latch, pulling the output latch enable high again.

There are two separate clock dividers used in the control logic, DIV16 and DIV8ST. The DIV16 is a divide-by-16 block which outputs a 50% duty cycle clock at 77.76 MHz. This clock divider is used to drive the external PCLKO line and drive the PDO0–PDO15 data output flip-flops, as well as the FP signal output flip-flop. The DIV16 divider is synchronous with the serial clock, but its phase with respect to the serial clock is arbitrary. The DIV8ST is a divide-by-8 strobe block which outputs a byte-aligned 800 ps strobe which is used to load the data from the shift register onto holding latches at appropriate times. This clock divider is reset when an inital framing pattern is found and is synchronized with the serial clock.

The DIV16 divider is essentially a 3-bit ripple counter with a final synchronizing divide-by-2 stage. See figure 7-12 for a diagram of the DIV16 divider. The divider is resettable for testing purposes—when RESET is high, the second and third flip-flops are reset internally, and the mux-flip-flop and last mux-latch are reset because the mux input 1 is selected which is tied low. Under normal operation, RESET is low so that the first three flip-flops operate identically, and together they form a 3-bit ripple counter. To analyze the operation of the last synchronous divide-by-2 stage, assume that all the bits are initially reset to "0" values, and the divider is counting up. The outputs of the first three flip-flops are ANDed together so that a ripple carry pulse is generated at binary 111. Since the last bit is a "0" the mux-latch 0-input is selected, and the AND output is a "1". On the next cycle the last bit is set to a "1" so that

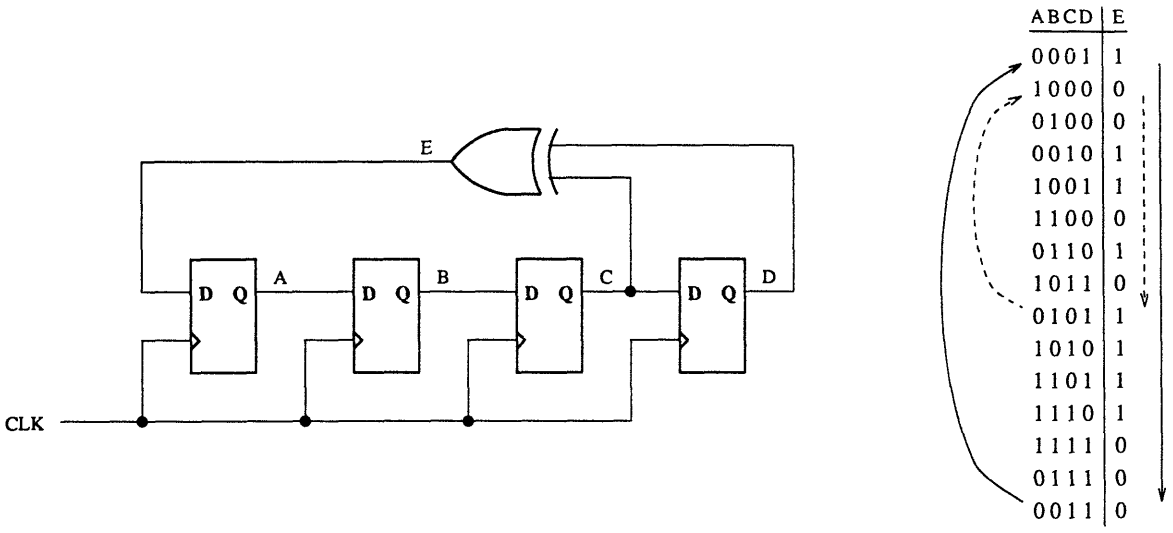


Figure 7-13: A 4-bit pseudo-random sequence generator and its sequence table

the mux-latch 1-input is selected, and the AND output goes low. Now the input to the mux-latch is inverted, but since the AND output is low a "1" is still clocked into the mux-latch until another ripple carry pulse appears on the output of the AND.

The DIV8ST divider is derived from a pseudo-random sequence bit generator because it takes fewer gates to implement than in the standard synchronous counter implementation. To step through the design process, we first analyze a 4-bit pseudo-random sequence generator which gives a 15-state sequence. Figure 7-13 shows the sequence generator and its associated sequence table. The outputs of each flip-flop are labeled with letters A through D which hold the current state ABCD, and the output of the XOR gate is labelled with the letter E. The states will be referred to by their binary number. The generator can be initialized at any desired state except the 0000 state. For example, the first state in the state table is shown as 0001, which means that the XOR output is 1. On the next CLK cycle the bits in the generator are shifted to the right by one position, and the bit E is shifted into the A flip-flop, so that the next state will be 1000. With this shift-and-feedback scheme, the sequence in the table in Figure 7-13 will occur beginning at the top and continuing down to the bottom, then wrapping around to the top again in an infinite loop shown by the solid arrows.

For the DIV8ST divider, we only need an 8-state sequence generator. The dashed

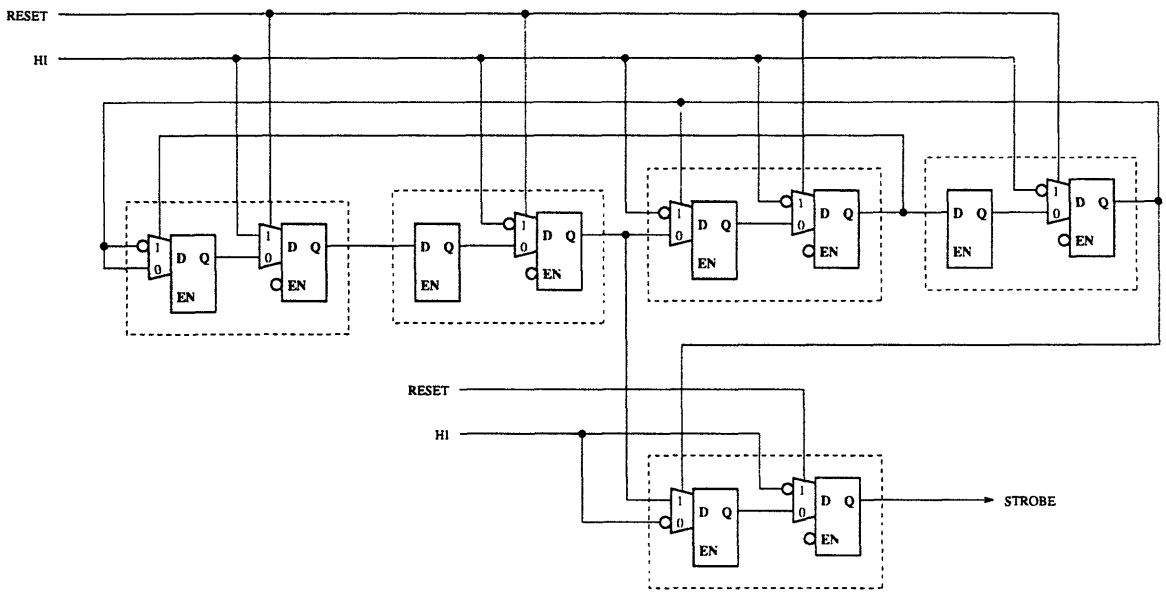
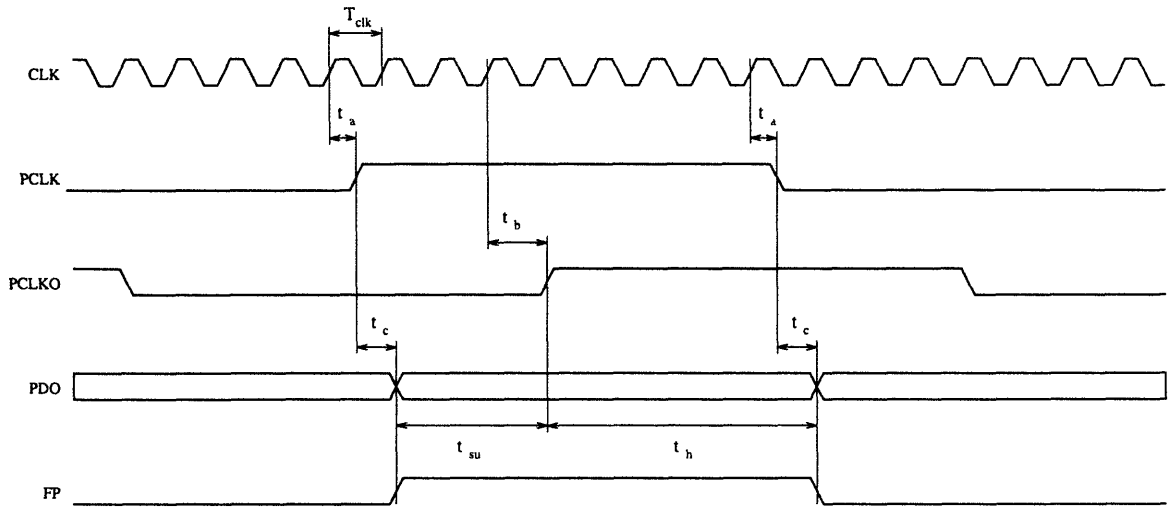


Figure 7-14: Circuit for the DIV8ST divider

arrow lines on the state sequence table show a possible 8-state sequence that could be used. There are a few nice properties about this new sequence that make it desirable to use. First of all, the first state of this new sequence, 1000, has only one bit different from the state 1010 which is directly after the last state in the new sequence. Because of this, only bit C needs to be toggled differently in order to make this counter loop back to state 1000, realizing the new 8-state sequence generator. This is easily accomplished—notice that in the first half of the old 15-state sequence, when bit D is a 1, bit C in the next state is a 0 *except* after the 0101 state. In order to make our new sequence, the only thing that needs to be done is to force bit C to always be 0 after bit D is 1.

Next, it would be nice if a single bit could be used as the STROBE signal, but there are no single bits within the 8-state sequence that are unique to that state. There is a two-bit pattern that is unique to the last state—bits B and D are only both 1s in state 0101, which is easily detected with a single logic gate. Figure 7-14 shows the modified DIV8ST circuit. In this figure the flip-flops have been broken down into two separate latches—the first latch of each pair is the latch or mux-latch that performs a logic function (such as XOR or AND) and the second latch is the reset mux-latch. When RESET is brought high for a full clock cycle, the value on



1/6/94

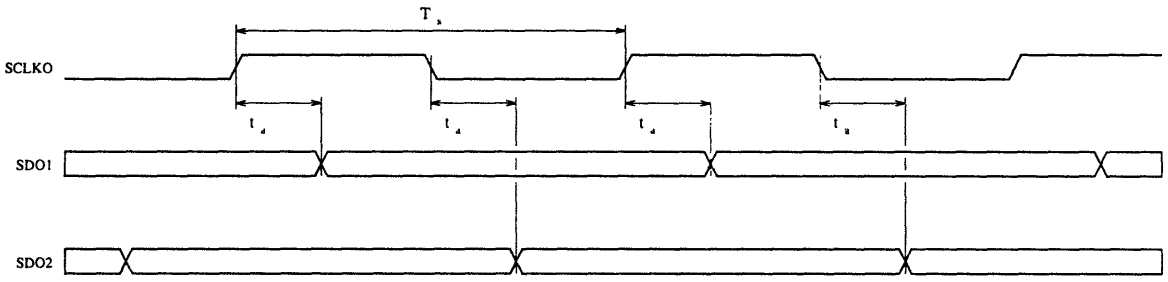
	Description	min	nom	max	units
T_{clk}	CLK period		804		ps
T_{pclk}	PCLK period		12860		ps
t_a	CLK to PCLK delay	275		576	ps
t_b	CLK to PCLKO delay	557		1181	ps
t_c	PCLK to PDO / FP delay	345		856	ps
t_{su}	Required PDO set-up time	1000			ps
t_h	Required PDO hold time	3000			ps

Figure 7-15: Timing diagram for parallel output clock and data

the 1-input of all the reset mux-latches is loaded in—in this case, the 0101 state is loaded into the main divider, and the output mux-latch is loaded with a 1 so that the STROBE line goes high immediately.

Figure 7-15 shows the timing relationship between the parallel clock output, the parallel data output, and the frame pulse.

Figure 7-16 shows the timing relationship between the serial output clock and the two output data streams.



	Description	min	nom	max	units
T_s	SCLKO period		804		ps
t_a	SCLKO to SDO delay		183		ps

Figure 7-16: Timing diagram for serial output clock and data

Chapter 8

Conclusion

8.1 Summary

A receiver module was designed which extracts a clock signal from a 2.5 Gb/s NRZ serial data stream, re-times the data, synchronizes to the data frames, and outputs the data as a 16-bit parallel word. The module should meet and exceed the required SONET and CCITT specifications for jitter tolerance and pattern dependence, and in addition it should also meet the jitter transfer specification which is not required for this application. Detailed SPICE simulations were performed using estimated parasitic capacitance to ensure successful operation over a variety of conditions.

8.2 Future Considerations

8.2.1 Analog/Digital Phase Detector Trade-offs

There are two phase detectors and one frequency detector in Pottbäcker's FPLL, all of which are digital emitter-coupled logic (ECL) circuits. Each of these blocks is relatively simple and is based on the flip-latch concept—see Appendix D. The flip-latch phase detectors can be thought of as digital sample and hold circuits—on every clock edge, the phase detector samples and holds the digital value at the input. The frequency detector is a small modification of the phase detector. There are advantages

and disadvantages to using a digital phase detector, but there are also advantages and disadvantages to using an analog version of the phase detector.

In typical analog phase detectors, a small phase error produces a corresponding small error voltage, and the phase detector has a finite "gain" near zero phase error. Even most digital phase detectors have a finite "gain" near zero, despite their switching behavior, because the phase error voltage is taken to be the average or DC component of a waveform whose duty cycle varies with phase error. One disadvantage of using the Pottbäcker digital phase detector is because a small phase error produces a full digital-level DC logic swing, resulting in essentially a "infinite" phase detector gain—the gain is only limited due to the decision-making ability of the latch. When there is zero phase offset, the latch will toggle between ± 200 mV due to jitter on the input data and the VCO, and this will DC average to zero. A small phase offset (which is larger than the input data and VCO jitter) produces a full-swing digital signal—this is a true DC phase error voltage, not an averaged phase error voltage.

On one hand, this "DC" output reduces the noise in the system—typical digital phase detection schemes produce harmonics which must be filtered out, but this scheme gives a DC output which needs no filtering. On the other hand, the "infinite" gain of the PD makes the PLL an inherently unstable system. This type of PLL is sometimes called a "bang-bang" PLL because it "bangs" back and forth around the center frequency. This action can add jitter directly to the VCO waveform unless the loop bandwidth is very small. Since this "jitter" is in the GHz range and the loop bandwidth is in the MHz range, it is relatively simple to construct a filter that substantially reduces this noise while not affecting the loop dynamics.

In his article, Pottbäcker does mention a method to make the latch into a semi-analog part. With this modification, the phase detector performs like a sample and hold circuit and the phase detector characteristic curve takes the shape of the VCO waveform—the output is simply a sampled version of the input. The phase detector gain in this case is equal to the maximum output voltage.

Because the die is pad-limited, both the analog and digital phase detectors can be fabricated on the same die for testing purposes. To test the digital PD, the analog

PD can be effectively removed by laser trimming the wire runs, and vice versa for the analog PD. If the digital phase detector performs well, the analog phase detector can either be laser trimmed from the circuits, or "permanently removed" with a single mask change. This would result in a substantial power savings because the digital PD is much less power hungary than the analog PD. If the digital phase detector fails to perform well, the digital phase detector can be similarly removed.

Appendix A

The Tektronix GST-1 Process, Modeling, and Standard Cells

A.1 The GST-1 Process

The Tektronix GST-1 process[10] is a self-aligned, double-polysilicon, Si-bipolar process with an NPN $f_T > 12$ GHz. GST-1 is designed for the purpose of building high density, high performance circuits. Three layers of metal are available, as well as schottky diodes, lateral PNPs, and trimmable Nichrome resistors. SPICE models are available which accurately model variations in process and temperature for all devices. These models are extracted from actual devices, and their accuracy is verified through the measured performance and high-yield of production designs.

A.2 The GST-1 Digital Standard Cell Library

The Tektronix GST-1 digital standard cell library[11] provides a resource for designing high-performance digital circuits that gives products a unique technological advantage. The library offer very high density with modest power dissipation in comparison to ECL gate array products. Emphasis is placed on differential logic as this gives the greatest leverage in terms of raw speed-power-product over single-ended alternatives.

The library contains 14+ generic cell families, each one including a subset of

functionally identical cell that offer a variety power and speed options. There are also 14+ peripheral cells which are not used in this thesis. Each cell is based on three-level differential ECL with 200 mV logic swings. The cell families include the standard gates such as latches, flip-flops, AND gates, multiplexers, level shifters, etc. In addition, more complex functions are implemented such as combined mux-latch and mux-flip-flop gates, asynchronously loadable latches and flip-flops, and others. These cells take advantage of the three level logic capabilities, saving both power and propagation delay.

For flexibility in design and simulation, the standard cell library includes transistor-level models with layout-extracted parasitics, as well as functional models, both of which are characterized over process, temperature, supply voltage, and loading variations. Simulation with the transistor-level models give the most accurate results, while the functional models allows fast simulations of large-scale and complex circuits. Worst-case models are typically used during design and evaluation which results in high yield and high reliability. Both the transistor and functional level models were used in designing this IC.

The propagation delay through a standard cell is characterized over loading conditions. For each cell there are two components to the delay—a fixed delay through the cell and a load dependent delay. For accurate timing analysis, both the fixed and load dependent delays must be taken into account. After layout, the parasitics that contribute to the load delay (interconnect) can be extracted and incorporated back into the simulation tools to refine the timing analysis.

There are other significant advantages in using the standard cells. Custom digital gates need not be designed, and so the design process can progress quickly. Also, since they have been used in numerous ICs, there is lower risk.

Appendix B

Applicable SONET/SDH Requirements

B.1 SONET Requirements

The following technical specifications are taken from Bellcore TR-NWT-000253 [6] and Bellcore TA-NPL-000436 [7].

B.1.1 Loss Of Signal (LOS)

Taken from Bellcore TR-NWT-000253 section 6.3.1.1.1

> To detect a failure that occurs at the source (e.g., laser failure) or the transmission facility (e.g., fiber cut), all incoming SONET signals are monitored for the loss of physical layer signal (optical and electrical). The detection of LOS must take place within a reasonably short period of time for timely restoration of the transported payloads. It is also important to distinguish between LOS and other signal failure conditions such as Loss of Frame (LOF), especially for intermittent failures, for trouble isolation purposes.
>
> **(R) 6-130** All incoming SONET signals (optical and electrical) shall be monitored for LOS before descrambling by detecting "all-zeros pat-

terns", where an "all-zeros pattern" corresponds to no light pulses for OC-N optical interfaces and no voltage transitions for STS-1 and STS-3 electrical interfaces. An "all-zeros pattern" on the incoming SONET signal shall be considered LOS if it lasts $100\mu s$ or longer and shall not be considered LOS if it lasts $2.3\mu s$ or shorter. SONET equipment shall enter the LOS state within $100\mu s$ of the onset of the incoming "all-zeros pattern."

The treatment of "all-zeros patterns" lasting between $2.3\mu s$ and $100\mu s$ for the purpose of LOS detection is not specified and is therefore left to the choice of the equipment designer. For testing compliance with the LOS detection requirement, it suffices to apply an "all-zeros pattern" lasting at most $2.3\mu s$ and to apply an "all-zeros pattern" lasting at least $100\mu s$.

(R) 6-131 Once in an LOS state, the SONET NE shall determine that the signal is acceptable (i.e., the SONET equipment shall exit the LOS state) when two consecutive valid frame alignment patterns have been detected and, during the intervening time (one frame), no "all-zeros pattern" considered as LOS was detected.

Taken from T1M1-3/93-005R1 section 8.1.2.1.1

An LOS defect is the occurance of no transitions on the incoming signal (before descrambling) for time T, where $2.3 \leq T \leq 100\mu s$.

The LOS defect is terminated after a time period equal to the greater of $125\mu s$ or $2.5T'$ containing no transition free intervals of length T', where $2.3 \leq T \leq 100\mu s$.[1]

B.1.2 Pull-in/Hold-in

Taken from TR-NWT-000253 section 5.4.3.2.3

[1]Note that termination of a LOS defect involves the reoccurance of genuine signal or clock transitions (not false transitions due to noise).

Hold-in range is defined as the measure of the maximum input frequency deviation from the nominal clock rate through which a synchronized clock can maintain a continuous phase-locked operation.

Pull-in range is defined as a measure of the maximum input frequency deviation from the nominal clock rate which can be overcome by a clock to pull itself into synchronization with another clock.

(R) 5-59 If a SONET Network Element (NE) is timed from an OC-N reference, the NE clock shall pull-in and hold-in to an OC-N reference that is off frequency by ±20 ppm.

B.1.3 Jitter

Taken from TR-NWT-000253 section 5.6

Jitter is defined as the short-term variation of the significant instants (e.g., rising edge or level crossing) of a digital signal from their ideal positions in time (short-term implies phase oscillations with spectral components greater than or equal to 10 Hertz).

Jitter Transfer

For Category II interfaces, the jitter transfer function must be under the curve in Figure B-1, when input a sinusoidal jitter up to the mask level in Figure B-2 is applied.

Jitter Tolerance

For Category II SONET interfaces, jitter tolerance is defined as the peak-to-peak amplitude of sinusoidal jitter is applied on the input OC-N/STS-N signal that causes a 1-dB power penalty. This is a stress test intended to ensure that no additional penalty is incurred under operating conditions.

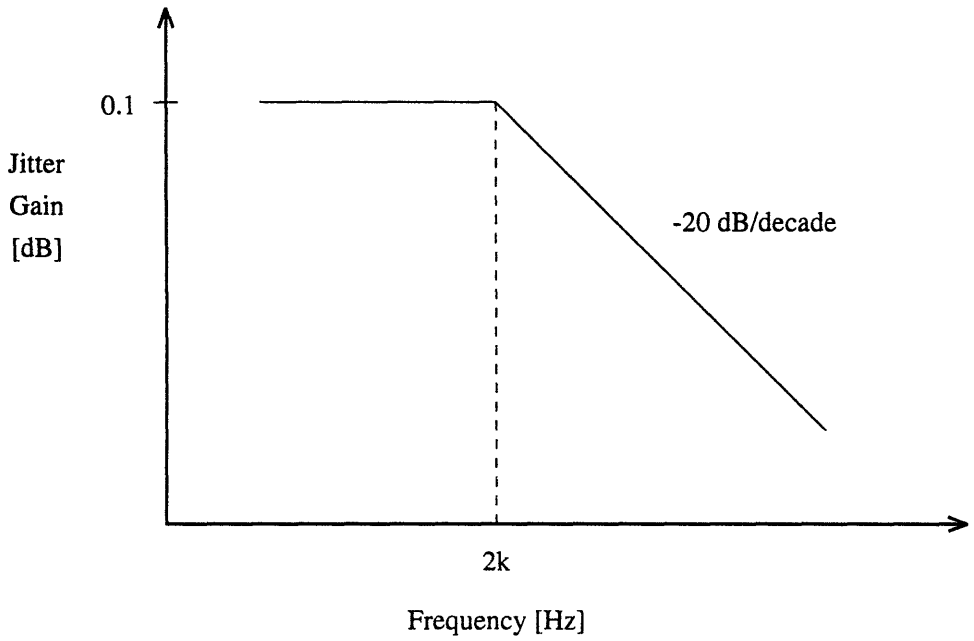


Figure B-1: SONET Category II Jitter Transfer Characteristics

(O) 5-10 OC-48 Category II SONET interfaces should tolerate, as an objective, the input jitter applied according to the mask in Figure B-2.

B.2 CCITT Requirements

B.2.1 STM-N Pattern Dependence Test

Taken from COM XV-R 42-E G.958

The duration of the zero-timing-content periods of A and C is made equal to the longest like-element sequences expected in the STM-N signal. A value of 9 bytes (72 bits) is provisionally proposed for this.

The duration of the pseudo-random periods should allow recovery of both the zero base line offset of the signal and of the timing recovery circuit following occurrence of the A and C periods. Therefore it should be longer than the longest time

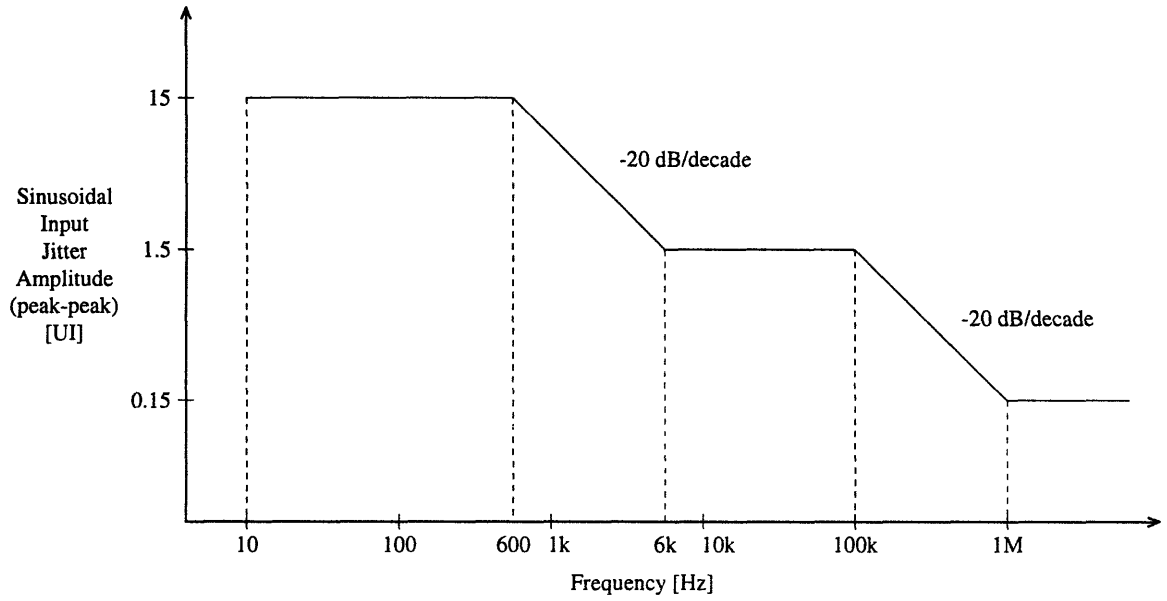


Figure B-2: SONET Category II Jitter Tolerance Mask

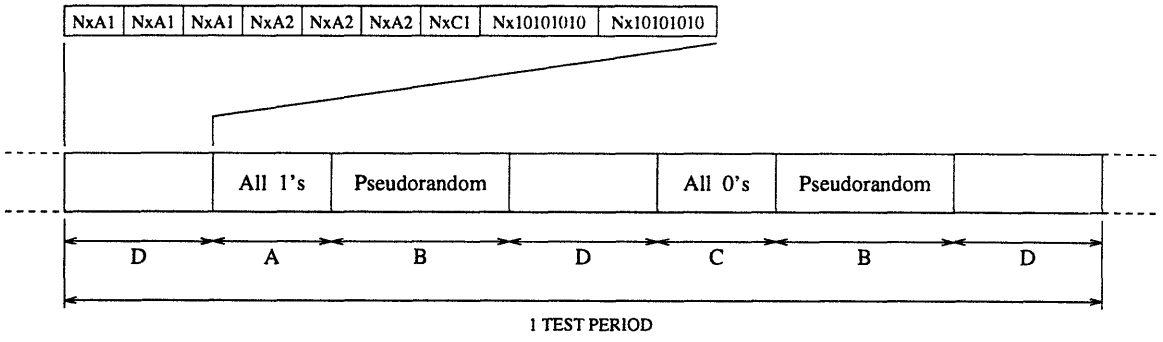


Figure B-3: CCITT pattern dependence test sequence

constant in the regenerator. In the case of a PLL based clock extraction this could give a value of the order of 10000 bits. Taking into account possible limitations of test equipment a minimum of 2000 bits is considered acceptable.

The content of the pseudo-random section should be generated by a scrambler having the same polynomial as defined in CCITT Recommendation G.709. Ideally, the scrambler should "free-run," i.e. the beginning of the pattern should be uncorrelated with the frame alignment section. This arrangement will ensure that the system experiences the worst possible phasing of the PRBS at some point during the course of the test. However it is recognized that test equipment limitations may preclude the use of a free running scrambler. Hence it may be necessary to specify a worst-case phasing of the PRBS. This is for further study.

The D period is defined as the first row of the section overhead of the STM-N signal, including valid C1 bytes (consecutive binary numbers).

Appendix C

Receiver I/O summary

All high-speed signals coming into the module are differential for minimum noise and maximum speed. This virtually eliminates asymetrical power supply current spikes normally associated with single-ended inputs, which reduces the on-chip noise. This also reduces the voltage swings across loads which increases speed and reduces power consumption. All inputs and outputs are 200 mV differential ECL "level 1" type signals. "Level 1" signals are offset from ground by one V_{BE} drop.

C.1 Inputs

SDI	Serial Data Input	The 2.48832 Gb/s incoming serial data, low-power, single-ended analog line.
BPSDI	Bypass Serial Data Input	A 2.48832 Gb/s serial data input used when the on-chip post amp is bypassed.
BPPAE	Bypass Post Amp Enable	Used to bypass the on-chip post amp and select the BPSDI input.
LBSDI	Loopback Serial Data Input	A 2.48832 Gb/s serial data input for in-system diagnostic loopback testing.
LBE	Loopback Enable	Used to select the LBSDI serial data input.

RCLKI	Reference Clock In	A precision 77.76 MHz reference clock which is used to provide a steady output clock during a LOSI condition.
TCLKI	Test Clock In-phase	Used for low-speed testing of the phase detector and frame detect circuits.
TCLKQ	Bypass Clock Quadrature	Used for low-speed phase detector testing.
TESTEN	Test Enable	Selects the alternate clock and data source for the Frame Detect block for Frame Detect/Demux testing.
RESET	Reset	System master reset line.
~SYN	Synchronize	Signals an Out-Of-Frame (OOF) condition. Active falling edge. Resets the Frame Detect block and disables the demux data output.
ELOP	External Loss-Of-Power Indicator	Signals that the input power to the pre-amp or limiting amp has fallen below a specified threshold and that the data is not valid.
SOE	Serial Output Enable	Enables the high-speed serial data outputs to allow in-system line loopback testing.

C.2 Outputs

LOSI	Loss-Of-Signal Indicator	Asserted when there is excessive inactivity on the selected data input.
LOPI	Loss-Of-Power Indicator	Asserted when the input power to the on-chip post amp is below a specified threshold.
PCLKO	Parallel Clock Out	A 77.76 MHz clock output for the parallel data lines. Both edges of this clock must be used to latch in the parallel data.
PDO[0..15]	Parallel Data Out	An 16-bit parallel 155.52 Mb/s data output.
SCLKO	Serial Clock Out	A 1.24416 GHz clock output for the serial output lines. SOE must be asserted to activate SCLKO.
SDO1	Serial Data Out 1	The re-timed serial data at 1.24416 Gb/s. SOE must be asserted to activate SDO1.
SDO2	Serial Data Out 2	The re-timed serial data at 1.24416 Gb/s. SOE must be asserted to activate SDO2.
FP	Frame Pulse	A double byte-wide pulse that occurs when a valid A1A1A2 framing sequence is found.
CF1	Capacitor Filter pad 1	Loop filter capacitor must be attached across CF1 and CF2.
CF2	Capacitor Filter pad 2	Loop filter capacitor must be attached across CF1 and CF2.

Appendix D

Flip-latch and Latch-latch Logic

D.1 The Flip-Latch Concept

The basic flip-latch consists of two latches and a 2:1 multiplexer. See Figure D-1. The operation of the flip-latch is simple. Assume at first that the clock is low. In this state, the bottom latch is transparent, the top latch is holding data, and the mux outputs the data data held at point A. On the next rising clock edge, the bottom latch holds the current data input onto point B, the mux outputs this data, and the top latch becomes transparent. On the next falling clock edge, the top latch holds the current data input onto point A, the mux outputs this data, and the

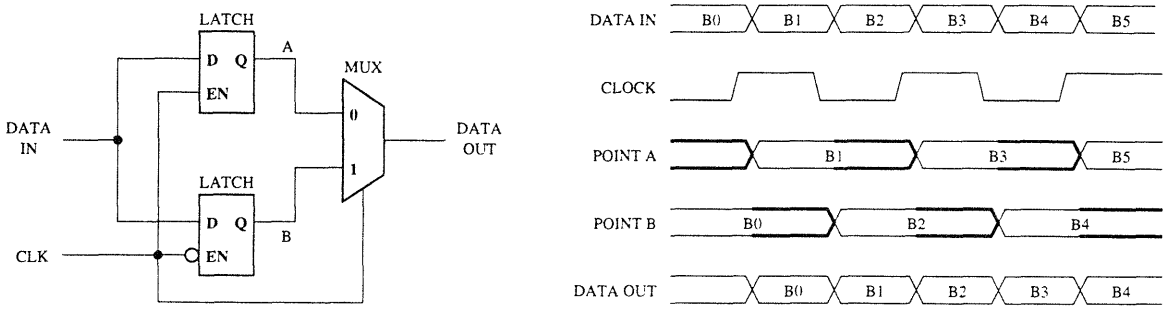


Figure D-1: Basic flip-latch architecture

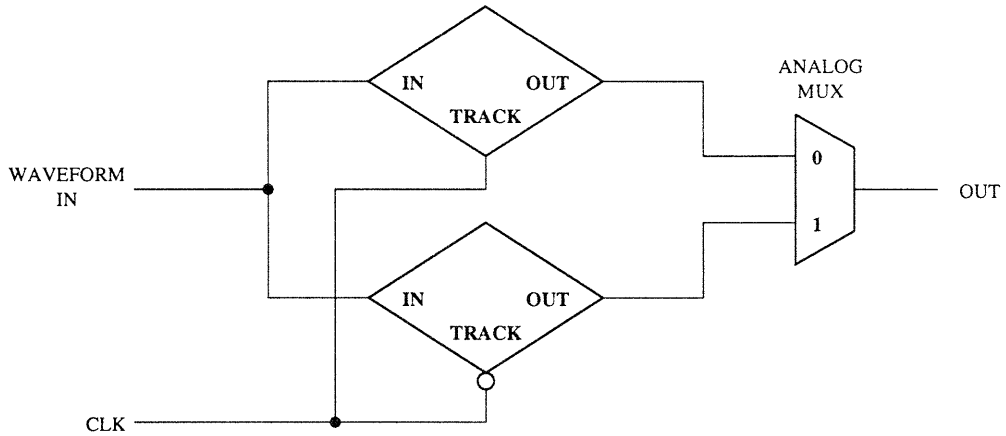


Figure D-2: Analog Sample and Hold Phase Detector

bottom latch becomes transparent. Assuming the setup time for the latch is not violated, the flip-latch clock-to-Q time is simply the select-to-Q time of the mux, which is relatively fast.

One advantage of the flip-latch is that at the same data rate, the flip-latch uses a clock that is half the rate of a flip-flop clock. This means that a flip-flop that has a 2.48832 GHz clock can be replaced with a flip-latch running with a 1.24416 GHz clock—the clock period is extended from 402 ps to 804 ps.

The flip-latch can be thought of as a digital sample and hold circuit—on each edge of the clock the circuit samples current input and holds this value until the next edge. Using this type of thinking, the flip-latch concept is extended into the analog domain and used in the analog phase detector. In this case the latches are replaced with their analog equivalents, track and hold circuits, and the digital multiplexer is replaced with an analog multiplexer. Figure D-2 shows the analog equivalent circuit. The functionality is equivalent, but the signals are fully analog.

D.2 Latch-latch Logic

Latch-latch logic is based on the same concept as flip-latches, and the following development will be derived using flip-latches. To begin, refer to

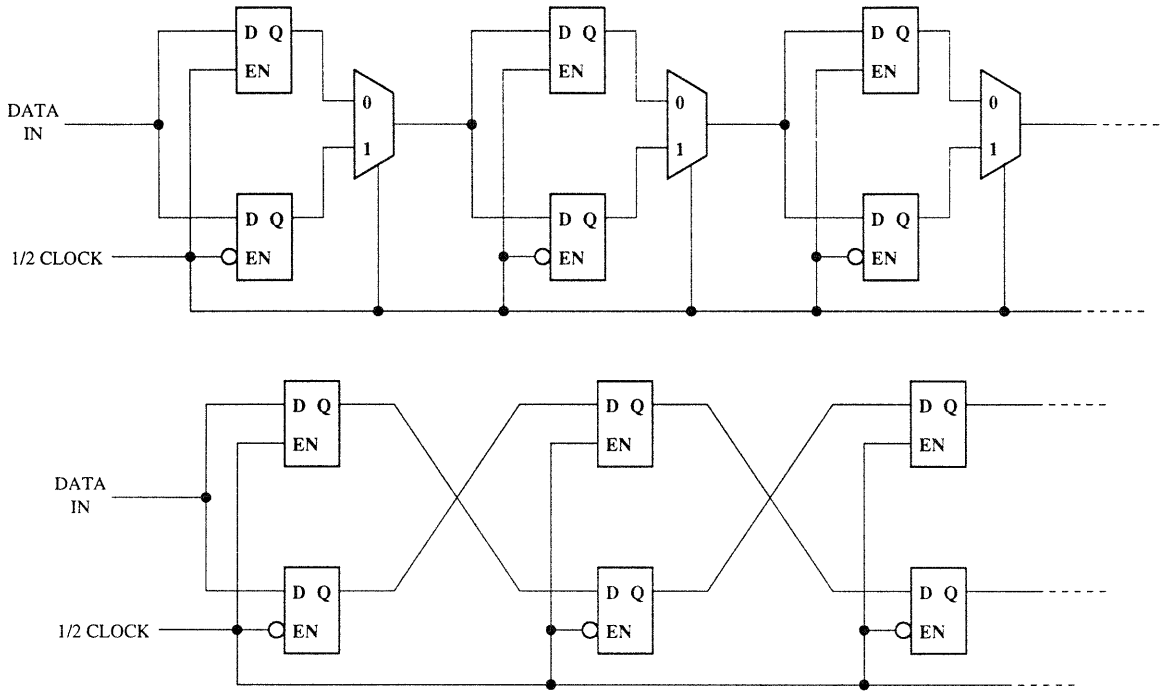


Figure D-3: Cascaded flip latches with and without the multiplexer

Figure D-3a which shows the first three flip-latches in a long cascade of flip-latches. This circuit simply samples an input of serial bits, and shifts these bits down the cascade string. Upon closer inspection of the cascaded flip-latch configuration, it can be shown that the multiplexer can be removed without affecting the normal operation. The function of the multiplexer is simply to feed the output of one latch to the next latch of opposite clock phase—the mux functions as a type of time division multiplexer. During one half-cycle, the "top" latch of the first stage is connected to the "bottom" latch of the next stage, and during the opposite half-cycle, the "bottom" latch of the first stage is connected to the "top" latch of the next stage. All of the "top" latches are active on one clock phase while the "bottom" latches are active on the opposite clock phase. Adjacent latches with similar phase clocks are never connected together, and so the multiplexers can be replaced simply with wires that connect opposite phase latches, as shown in Figure D-3b. In doing this, slower latches can be used—essentially half the speed and half the power—because the latch-

to-latch delay is approximately cut in half. The GST-1 digital standard cell library can be used, and is ideal for this application (see Appendix A).

Bibliography

[1] Toni Beninger. *SONET Basics.* Intertec Publishing Corp., Chicago, 1991.

[2] Aaron W. Buchwald, Kenneth W. Martin, Aaron K. Oki, and Kevin W. Kobayashi. A 6-GHz Integrated Phase-Locked Loop Using AlGaAs/GaAs Heterojunction Bipolar Transistors. *IEEE Journal of Solid-State Circuits*, 27(12):1752–1761, December 1992.

[3] Dennis T. Kong. 2.488 Gb/s SONET Multiplexer/Demultiplexer with Frame Detection Capability. *IEEE Journal on Selected Areas in Communications*, 9(5):726–731, June 1991.

[4] Arthur J. Metz. Adjustable delay element for digital systems. Technical report, United States Patent 4,801,827, 1987.

[5] Ansgar Pottbacker, Ulrich Langmann, and Hans-Ulrich Schreiber. A Si Bipolar Phase and Frequency Detector IC for Clock Extraction up to 8 Gb/s. *IEEE Journal of Solid-State Circuits*, 27(12):1747–1751, December 1992.

[6] Bell Communications Research. TR-NWT-000253 Synchronout Optical Network (SONET) Transport Systems: Common Generic Criteria. Technical Reference 2, Bell Communications Research, Inc., Morristown, NJ, December 1991.

[7] Bell Communications Research. TA-NWT-000436 Digital Network Synchronization Plan. Technical Advisory 2, Bell Communications Research, Inc., Morristown, NJ, June 1993.

[8] Mike Sexton and Andy Reid. *Transmission Networking: SONET and the Synchronous Digital Hierarchy.* Artech House, Boston, 1992.

[9] Tektronix, Inc. SDH/SONET Telecommunications Standards Primer. Technical report, Tektronix, Inc., Beaverton, OR, 1991.

[10] Tektronix, Inc. GST-1 Full Custom Integrated Circuit Design Guide. Technical reference, Tektronix, Inc., Beaverton, OR, 1992.

[11] Tektronix, Inc. GST-1 Standard Cell IC User Documentation. Technical reference, Tektronix, Inc., Beaverton, OR, 1993.

[12] Dan H. Wolaver. *Phase-Locked Loop Circuit Design.* Prentice-Hall, New Jersey, 1991.