A Delay-Locked Loop for Clock Recovery and Data Synchronization

by

John F. Bulzacchelli

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements of the Degrees of
Bachelor of Science in Electrical Engineering
and
Master of Science in Electrical Engineering
and
Electrical Engineer
at the
Massachusetts Institute of Technology
June, 1990

©John F. Bulzacchelli, 1990
The author hereby grants to MIT permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signature of Author

Department of Electrical Engineering and Computer Science
January 31, 1990

Certified by

James K. Roberge
Professor of Electrical Engineering
Thesis Supervisor

Certified by

Lawrence M. DeVito
Company Supervisor

Accepted by

Arthur C. Smith
Director of the Graduate School
A Delay-Locked Loop for Clock Recovery and Data Synchronization
by
John F. Bulzacchelli

Submitted to the Department of Electrical Engineering
and Computer Science on January 31, 1990 in partial fulfillment
of the requirements for the Degree of Master of Science in
Electrical Engineering.

ABSTRACT

Two new clock recovery circuits for NRZ data have been developed: the delay-locked
loop (DLL) and the delay-and-phase-locked loop (D/PLL). While application of the
DLL requires the availability of an exact frequency reference, no such restriction
applies to the D/PLL. By utilizing a voltage-controlled phase shifter, the DLL and
the D/PLL avoid many of the tradeoffs which limit the performance of conventional
clock recovery circuits. Rapid acquisition is achieved without compromising jitter
filtering. Neither the DLL nor the D/PLL exhibits jitter peaking. Static phase error
is ideally zero, only limited by the offset of the phase detector and the finite d.c.
gain of practical loop integrators. Both the DLL and the D/PLL are amenable to
monolithic construction. A differential charge pump phase detector which minimizes
static phase error and data-dependent jitter has been designed, and its performance
has been verified by computer simulation.

In order to evaluate the performance of an actual DLL and an actual D/PLL,
breadboard prototypes operating at approximately 40 Mbit/s were constructed. Mea-
surements taken on the prototypes confirmed the theoretical advantages of the DLL
and D/PLL architectures. For both the DLL and D/PLL prototypes, acquisition
occurs within 1 μs (40 clock periods). For the D/PLL prototype, the bandwidth of
the jitter transfer function is less than 1.5 KHz. The jitter transfer function of the
DLL prototype simply equals zero.

Thesis supervisors: Professor James K. Roberge
Professor of Electrical Engineering
Massachusetts Institute of Technology

Lawrence M. DeVito
Linear Operating Group
Analog Devices Semiconductor, Wilmington, MA
To Mom and Dad
Acknowledgements

I am grateful to the following people for all of their support in this project.

First I would like to thank Larry DeVito, my supervisor and mentor at Analog Devices Semiconductor, for his guidance, support, and inspiration throughout this study. Many of the best ideas presented in this thesis were the product of Larry’s imaginative suggestions. Above all, I appreciate Larry’s belief in my ability to carry out this research, even when its outcome looked doubtful.

Next I wish to thank Fred Benkley, who unselfishly worked with me on the design of the breadboard prototypes. Fred laid out the printed circuit boards for the prototypes and was instrumental in helping me debug them.

I also want to thank Bob Surette, the lab technician at ADS, who built the breadboard prototypes. The pride and care with which Bob carried out his work contributed to the successful operation of the prototypes.

Another person to whom I am indebted is John Newton. Our discussions often clarified issues with which I had been struggling. John also spent considerable time reviewing and improving this thesis.

I am also grateful to the innumerable people at ADS who devoted their valuable time to helping me complete this project.

I also thank Professor Roberge, my thesis supervisor at MIT, for reading the thesis and suggesting improvements.

Finally, I give thanks to my parents John and Frances, without whose love and support this research could not have been conducted.
## Contents

1 Introduction 12

2 Conventional Approaches to Clock Recovery 14
   2.1 Definition of Terms ................................. 15
   2.2 Clock Recovery Using Resonant Circuits ............ 17
      2.2.1 Basic Operating Principle ........................ 17
      2.2.2 Tradeoff Between Static Phase Error and Jitter ...... 20
      2.2.3 Acquisition Time .................................... 21
      2.2.4 Cost .................................................. 21
   2.3 Clock Recovery Using Phase-Locked Loops ............ 22
      2.3.1 Basic Operating Principle ........................ 22
      2.3.2 Static Phase Error .................................. 22
      2.3.3 Jitter Transfer Function and Jitter Peaking ........ 25
      2.3.4 Acquisition Time .................................... 27
   2.4 Prospectus .............................................. 28

3 The Delay-Locked Loop and the Delay-and-Phase-Locked Loop 29
   3.1 The Delay-Locked Loop ............................... 29
      3.1.1 Need for Delay-Locked Loop ....................... 29
      3.1.2 Basic Operating Principle ........................ 30
      3.1.3 Static Phase Error .................................. 32
      3.1.4 Jitter Transfer Function ........................... 33
      3.1.5 Acquisition Time .................................... 33
      3.1.6 Need to Reset the DLL .............................. 34
   3.2 The Delay-and-Phase-Locked Loop ..................... 35
      3.2.1 Basic Operating Principle ........................ 35
      3.2.2 Loop Dynamics of D/PLL ............................ 37
      3.2.3 Acquisition .......................................... 39
      3.2.4 Practical Limitations on Jitter Filtering ........ 45
      3.2.5 Need to Reset the D/PLL ............................ 48
4 Phase Detectors

4.1 The Multiplier-Based Phase Detector
4.1.1 Basic Operating Principle
4.1.2 The Offset of the Multiplier-Based Phase Detector
4.1.3 Phase Detector Gain
4.1.4 Metastability
4.1.5 The Ripple of the Multiplier-Based Phase Detector

4.2 The Self-Correcting Phase Detector
4.2.1 Basic Operating Principle
4.2.2 The Offset of the Self-Correcting Phase Detector
4.2.3 Phase Detector Gain
4.2.4 Metastability
4.2.5 The Ripple of the Self-Correcting Phase Detector

4.3 The Triwave Phase Detector
4.3.1 Basic Operating Principle
4.3.2 The Ripple of the Triwave Phase Detector

5 Design and Simulation of a Triwave Phase Detector

5.1 The Differential Charge Pump
5.1.1 Limitations of Process Technology
5.1.2 The Basic Differential Charge Pump
5.1.3 Negative Resistance Circuit
5.1.4 Effect of Current Source Mismatches in the Charge Pump
5.1.5 Reset Circuitry
5.1.6 The Complete Differential Charge Pump

5.2 The Digital Control Logic of the Phase Detector
5.2.1 Implementation of Flip-Flops with Latches
5.2.2 D-Type Latch
5.2.3 XOR Gate

5.3 Simulated Performance of Complete Phase Detector
5.3.1 Open Loop Results
5.3.2 Ideal Phase Shifter
5.3.3 Closed Loop Results

6 Breadboard Prototypes and Experimental Results

6.1 Description of Breadboard Prototypes
6.1.1 Phase Detector Prototype
6.1.2 Phase Shifter Prototype
6.1.3 VCXO Prototype ................................................. 119
6.1.4 Circuit Board Layout Techniques .......................... 119
6.2 Open Loop Measurements ........................................ 122
  6.2.1 Phase Shifter Measurements .............................. 122
  6.2.2 VCXO Measurements ......................................... 127
6.3 DLL Measurements ............................................... 128
  6.3.1 Acquisition ................................................ 128
  6.3.2 Static Phase Error ......................................... 128
  6.3.3 Jitter Performance ......................................... 132
6.4 D/PLL Measurements .............................................. 135
  6.4.1 Phase Acquisition ......................................... 135
  6.4.2 Frequency Acquisition .................................... 135
  6.4.3 Static Phase Error ......................................... 140
  6.4.4 Jitter Performance ......................................... 140

7 Conclusion .......................................................... 147

A Effect of Clock Asymmetry on Phase Detector Offset ....... 149
  A.1 The Multiplier-Based Phase Detector ...................... 149
  A.2 The Self-Correcting Phase Detector ....................... 151
  A.3 The Triwave Phase Detector ................................ 152
  A.4 The Modified Triwave Phase Detector ..................... 153
    A.4.1 Basic Operating Principle .............................. 153
    A.4.2 Effect of Clock Asymmetry .............................. 155

B Triwave Phase Detector Modification to Avoid Equivocation 157
  B.1 Basic Operating Principle .................................. 157
  B.2 Metastability ................................................. 161
List of Figures

1.1 NRZ Data ................................................. 13

2.1 Simplified Block Diagram of a Digital Receiver ...................... 14
2.2 Tuned Circuit Approach to Clock Recovery ............................ 17
2.3 Clock Recovery Circuit with Nonlinear Processor ..................... 18
2.4 Simple Nonlinear Processor ....................................... 18
2.5 Timing Diagram for Simple Nonlinear Processor ...................... 18
2.6 Power Spectra of Random NRZ Data before Nonlinear Processing .... 19
2.7 Power Spectra of Random NRZ Data after Nonlinear Processing .... 19
2.8 Phase-Locked Loop Approach to Clock Recovery ...................... 22
2.9 First-Order PLL ............................................ 23
2.10 Second-Order PLL with Integrator .................................. 24
2.11 Root Locus for Second-Order PLL ................................. 25
2.12 Jitter Transfer Function for a Critically Damped PLL ............... 26

3.1 Bus Architecture for Serial Data Communications .................... 30
3.2 Delay-Locked Loop for Clock Recovery .............................. 31
3.3 Improved DLL for Clock Recovery .................................. 32
3.4 First-Order DLL with Integrator ................................... 33
3.5 Delay-and-Phase-Locked Loop for Clock Recovery .................... 36
3.6 Improved D/PLL for Clock Recovery ................................ 36
3.7 D/PLL with Integrator ......................................... 37
3.8 Illustration of Zero in Loop Transmission of D/PLL ................. 37
3.9 Improved D/PLL with Integrator ................................. 38
3.10 Jitter Transfer Function, With and Without Zero .................... 40
3.11 First stage of phase acquisition .................................. 43
3.12 Second stage of phase acquisition ............................... 43
3.13 Frequency Acquisition of D/PLL .................................. 45
3.14 Transfer Characteristics of VCXO and Phase Shifter ................. 47
3.15 Transfer Characteristics Permitting the Use of the VCXO's Entire Range .................. 47
4.1 The Multiplier-Based Phase Detector ........................................... 51
4.2 Timing Diagram for the Multiplier-Based Phase Detector ................. 52
4.3 Timing Diagram: Delayed Data Advanced ..................................... 52
4.4 Transfer Characteristic of the Multiplier-Based Phase Detector ........ 53
4.5 DLL with Multiplier-Based Phase Detector .................................. 54
4.6 DLL with Higher-Order Poles .................................................... 56
4.7 Timing Diagram Illustrating Effect of Higher-Order Poles ................. 56
4.8 Timing Diagram: Change in Data Transition Density ....................... 57
4.9 The Self-Correcting Phase Detector .......................................... 58
4.10 Timing Diagram for the Self-Correcting Phase Detector ................. 60
4.11 Timing Diagram: Delayed Data Advanced ................................... 60
4.12 Transfer Characteristic of the Self-Correcting Phase Detector ......... 61
4.13 Effect of Flip-Flop Delays on Pulse Widths ................................ 61
4.14 Self-Correcting Phase Detector with Compensating Delay ............... 62
4.15 The Triwave Phase Detector ................................................... 65
4.16 Timing Diagram for the Triwave Phase Detector ......................... 67
4.17 Timing Diagram: Delayed Data Advanced ................................... 67
4.18 Root Contour for the DLL with 1 Higher-Order Pole ...................... 69
4.19 Output of Loop Integrator: Large Changes in Transition Density ....... 71
4.20 Filtered Control Voltage: One Higher-Order Pole with $\tau_1 = 63\,ns$ .... 73
4.21 Filtered Control Voltage: Two Higher-Order Poles with $\tau_2 = 35\,ns$ ... 73
4.22 Filtered Control Voltage: Three Higher-Order Poles with $\tau_3 = 25\,ns$ ... 74
4.23 Filtered Control Voltage: Four Higher-Order Poles with $\tau_4 = 20\,ns$ ... 74

5.1 Charge Pump Implementation of Triwave Phase Detector ................. 76
5.2 Timing Diagram: Charge Pump Implementation ................................ 76
5.3 The Differential Charge Pump .................................................. 78
5.4 Resistively Biased Charge Pump ............................................... 81
5.5 Simple Negative Resistance Circuit ......................................... 83
5.6 Improved Negative Resistance Circuit ....................................... 84
5.7 Negative Resistance Circuit with Base Current Compensation .......... 87
5.8 Fully Compensated Negative Resistance Circuit .......................... 89
5.9 Decay of Charge Pump Capacitor Voltage .................................. 89
5.10 Charge Pump with Reset Circuitry ............................................ 92
5.11 Digital Controller for Reset Circuitry ...................................... 94
5.12 Timing Diagram for Digital Controller ..................................... 94
5.13 Charge Pump Capacitor Voltage during Reset ................................ 95
5.14 Complete Schematic of Differential Charge Pump ........................ 96
5.15 Implementation of Flip-Flops with 6 Latches ............................... 98
5.16 Implementation of Phase Detector Logic with 4 Latches ................ 98
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.17 D-Type Latch</td>
<td>99</td>
</tr>
<tr>
<td>5.18 XOR Gate</td>
<td>100</td>
</tr>
<tr>
<td>5.19 Correct Way of Connecting XOR Gates with Asymmetrical Delays</td>
<td>101</td>
</tr>
<tr>
<td>5.20 Use of XOR Gate as Compensating Delay</td>
<td>102</td>
</tr>
<tr>
<td>5.21 Phase Detector Waveforms: Delayed Data 60 ps Behind Clock</td>
<td>104</td>
</tr>
<tr>
<td>5.22 The Basic Voltage-Controlled Delay Element</td>
<td>105</td>
</tr>
<tr>
<td>5.23 Timing Diagram for the Basic Delay Element</td>
<td>106</td>
</tr>
<tr>
<td>5.24 Ideal Phase Shifter</td>
<td>107</td>
</tr>
<tr>
<td>5.25 Timing Diagram for Ideal Phase Shifter</td>
<td>107</td>
</tr>
<tr>
<td>5.26 Voltage on Charge Pump Capacitor During Acquisition</td>
<td>110</td>
</tr>
<tr>
<td>5.27 Filtered Output of Phase Detector</td>
<td>110</td>
</tr>
<tr>
<td>5.28 Alignment between Delayed Data and Clock After 900 ns</td>
<td>111</td>
</tr>
<tr>
<td>5.29 Alignment between Delayed Data and Clock After 1.73 µs</td>
<td>112</td>
</tr>
<tr>
<td>5.30 Charge Pump Capacitor Voltage: Negative Resistance Circuit Disabled</td>
<td>113</td>
</tr>
<tr>
<td>5.31 Filtered Output of Phase Detector: Negative Resistance Circuit Disabled</td>
<td>113</td>
</tr>
<tr>
<td>6.1 Schematic of Phase Detector Prototype</td>
<td>115</td>
</tr>
<tr>
<td>6.2 Schematic of Phase Shifter Prototype</td>
<td>118</td>
</tr>
<tr>
<td>6.3 Schematic of VCXO Prototype</td>
<td>120</td>
</tr>
<tr>
<td>6.4 Photograph of Phase Detector Prototype</td>
<td>121</td>
</tr>
<tr>
<td>6.5 Photograph of Phase Shifter Prototype</td>
<td>121</td>
</tr>
<tr>
<td>6.6 Photograph of VCXO Prototype</td>
<td>122</td>
</tr>
<tr>
<td>6.7 Transfer Characteristic of Phase Shifter Prototype</td>
<td>123</td>
</tr>
<tr>
<td>6.8 Effect of Data Transition Density on Delay Through Phase Shifter</td>
<td>124</td>
</tr>
<tr>
<td>6.9 Data-Dependent Jitter Added to Delayed Data by Phase Shifter</td>
<td>126</td>
</tr>
<tr>
<td>6.10 Transfer Characteristic of VCXO Prototype</td>
<td>127</td>
</tr>
<tr>
<td>6.11 Control Voltage of DLL During Acquisition</td>
<td>129</td>
</tr>
<tr>
<td>6.12 Control Voltage of DLL During Metastability</td>
<td>130</td>
</tr>
<tr>
<td>6.13 Steady State DLL Waveforms</td>
<td>131</td>
</tr>
<tr>
<td>6.14 Random Jitter on Delayed Data of DLL</td>
<td>133</td>
</tr>
<tr>
<td>6.15 Random and Data-Dependent Jitter on Delayed Data of DLL</td>
<td>134</td>
</tr>
<tr>
<td>6.16 Control Voltage of D/PLL: First Stage of Phase Acquisition</td>
<td>136</td>
</tr>
<tr>
<td>6.17 Control Voltage of D/PLL: Beginning of Second Stage</td>
<td>137</td>
</tr>
<tr>
<td>6.18 Control Voltage of D/PLL: Second Stage of Phase Acquisition</td>
<td>138</td>
</tr>
<tr>
<td>6.19 Control Voltage of D/PLL: Frequency Acquisition</td>
<td>139</td>
</tr>
<tr>
<td>6.20 Steady State D/PLL Waveforms</td>
<td>141</td>
</tr>
<tr>
<td>6.21 Random Jitter on Delayed Data of D/PLL</td>
<td>143</td>
</tr>
<tr>
<td>6.22 Random Jitter on Recovered Clock of D/PLL</td>
<td>144</td>
</tr>
<tr>
<td>6.23 Random and Data-Dependent Jitter on Delayed Data of D/PLL</td>
<td>145</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

It has long been recognized that the ability to regenerate binary data is an inherent advantage of digital transmission of information. Provided that a bit error does not occur, the received bit stream is identical to the one originally transmitted, and no information is lost. In order to regenerate binary data with the fewest bit errors, the data must be sampled at the optimum instants of time. The timing information for sampling is contained in a signal known as the clock. In digital communications systems, the transmission of a separate clock over long distances is not practical. It is more economical to implement the clock recovery circuitry needed to derive the timing information from the incoming data itself [1].

When the incoming data signal has spectral energy at the clock frequency, a synchronous clock can be obtained by passing the incoming data through a bandpass filter tuned to the nominal clock frequency. Because of bandwidth restrictions, however, in most signalling formats the incoming data signal itself has no spectral energy at the clock frequency. One such signalling format is non-return-to-zero (NRZ), also called simple binary. In the NRZ format, which is illustrated in Figure 1.1, each “one” is encoded as a high voltage, and each “zero” is encoded as a low voltage. The lack of spectral energy at the clock frequency in NRZ data complicates clock recovery.

The main purpose of this project was to examine the limitations of existing clock recovery circuits for NRZ data, and then to design an architecture for clock recovery which overcomes some of these limitations. Chapter 2 reviews two conventional clock recovery techniques: the tuned circuit approach, and the phase-locked loop (PLL) approach. The cost and performance tradeoffs of each approach are examined in detail. Chapter 3 introduces two new architectures for clock recovery: the delay-locked loop (DLL) and the delay-and-phase-locked loop (D/PLL). By utilizing a voltage-controlled phase shifter, the DLL and D/PLL architectures offer a combination of jitter filtering and rapid acquisition not achievable with conventional clock recovery circuits.
Chapters 4 through 6 describe the detailed implementation of a DLL and a D/PLL. Particular attention is given to the design of the phase detector, as imperfections in the phase detector often limit the performance of the entire clock recovery circuit. Chapter 4 examines the performance of three phase detectors suitable for use in a DLL or a D/PLL. Guided by the analysis of the previous chapter, Chapter 5 presents the detailed transistor-level design of a differential charge pump phase detector; the design is suitable to monolithic integration. Simulation results are discussed at the end of the chapter. To evaluate the performance of an actual DLL and an actual D/PLL, breadboard prototypes were constructed. Chapter 6 describes the prototypes and examines their measured performance. Concluding remarks are made in Chapter 7.
Chapter 2

Conventional Approaches to Clock Recovery

Timing recovery is a critical receiver function in digital communications systems. In order to regenerate binary data with the smallest bit error rate (BER), the data must be sampled at the optimum instants of time. Throughout this thesis, the optimum instants for sampling are assumed to be in the middle of each bit cell. Typically, this timing information is derived from the incoming data. Since the recovered timing information is contained in a signal known as the clock, the process of symbol synchronization is called clock recovery. The role of clock recovery in digital receivers is illustrated in Figure 2.1.

![Figure 2.1: Simplified Block Diagram of a Digital Receiver](image-url)
In high speed communications systems (e.g. fiber optics), conventional clock recovery techniques fall under two general categories: the tuned circuit approach, and the phase-locked loop (PLL) approach [2]. This chapter reviews both of these approaches and examines their limitations.

2.1 Definition of Terms

Before reviewing past approaches to symbol synchronization, it is necessary to define the basic terminology that is used to describe the performance of clock recovery circuits. The following terms apply to all clock recovery circuits:

- **Phase Error – Static and Dynamic**
  
  As mentioned above, the optimum instants for sampling the data are often assumed to be in the middle of each bit cell. The most basic function of any clock recovery circuit is to maintain the sampling edge of the clock (usually the rising edge) near this optimum position. The phase error of the clock recovery circuit is the time difference between the sampling edge of the clock and the optimum sampling instant. The static phase error is the d.c. average of phase error and should always be minimized. The dynamic phase error is the a.c. component of phase error. Minimizing the dynamic phase error is not always desirable, for it often conflicts with the clock recovery circuit's ability to filter out jitter.

- **Jitter – Random and Data-Dependent**

  Jitter (or phase noise) is simply the displacement of digital edges from their long term average positions. The term jitter can apply to all the digital signals in a clock recovery system. Jitter on the incoming data can cause dynamic phase errors. Jitter on the recovered clock causes jitter in the retimed data.

  There are two distinct classes of jitter. Random jitter arises from physical noise processes, such as thermal noise. Like noise, random jitter can only be bounded in a probabilistic sense. The other class of jitter is data-dependent jitter (or “pattern noise”). It is caused by the aperiodic nature of the data signal. Data-dependent jitter can often be bounded in a deterministic sense. In many communications systems data-dependent jitter is the dominant source of jitter [3,4].
• Jitter Transfer Function and Jitter Peaking

The jitter transfer function describes the amount of jitter induced on the recovered clock by sinusoidal phase modulation of the incoming data. In general, the amount of jitter induced on the recovered clock is a strong function of the frequency of the modulation. At low frequencies the recovered clock tracks the phase of the incoming data, and the output jitter equals the input jitter. At high frequencies the recovered clock cannot track the input jitter, and the magnitude of the jitter transfer function is less than one. Hence, the jitter transfer function typically has a lowpass filter characteristic.

A clock recovery circuit is said to exhibit jitter peaking if the magnitude of the jitter transfer function exceeds unity at any modulation frequency. Jitter peaking is especially deleterious in repeater applications, in which cascaded regenerators ret ime the data over and over again. If no jitter peaking exists in the retiming circuits, data-dependent jitter accumulates as \( \sqrt{N} \) and random jitter as \( \sqrt{N} \), where \( N \) is the number of regenerators in the chain. If jitter peaking exists, both data-dependent jitter and random jitter accumulate exponentially with the number of regenerators [5]. For this reason, jitter peaking is not usually tolerated in cascaded regenerators.

• Data Transition Density

Data transition density measures the number of consecutive clock periods between edges of the incoming bit stream. A pattern like 01010101 is said to have a high data transition density, while a pattern like 00001111 is said to have a low data transition density. In an effort to minimize data-dependent jitter and aid acquisition, most data is encoded to limit the minimum data transition density.

• Acquisition Time

In some communication systems, the incoming data stream is rarely interrupted, and the clock is expected to remain in synchronization for long periods of time. In others, the data stream consists of short bursts; often, the phase relationship between adjacent bursts is random [6]. Consequently, at the start of each burst, the clock must acquire synchronization before the data can be successfully regenerated. The time required for this to happen is the acquisition time of the clock recovery circuit. To maximize channel capacity, the acquisition time should be minimized. However, there is often a tradeoff between acquisition time and jitter filtering.

Sometimes there is a preamble code at the beginning of each burst; one of its functions is to make the acquisition process faster or more reliable. Usually,
preamble codes possess high transition densities.

2.2 Clock Recovery Using Resonant Circuits

2.2.1 Basic Operating Principle

Perhaps the most straightforward method of extracting a synchronous clock from the incoming data is to process the incoming data through a bandpass filter tuned to the nominal clock frequency. The sinusoidal output of the bandpass filter can be converted into a square wave through the use of a hard limiter. Such an architecture is depicted in Figure 2.2. The bandpass filter can either be a passive lumped element filter or a surface acoustic wave (SAW) filter.

![Tuned Circuit Approach to Clock Recovery](image)

Figure 2.2: Tuned Circuit Approach to Clock Recovery

In most signalling formats (such as NRZ), the data signal itself has no spectral energy at the clock frequency. A discrete spectral component at the clock frequency must then be generated from the data signal through appropriate nonlinear processing [7]. For this reason, a nonlinear processor is usually placed between the incoming data and the bandpass filter, as shown in Figure 2.3. The nonlinear processing circuitry can be as simple as a delay line and one exclusive OR (XOR) logic gate (see Figure 2.4). This circuit produces a positive pulse upon each data edge, as demonstrated in Figure 2.5. Under the assumption of random NRZ data, the incoming data has the power spectra shown in Figure 2.6, and the output of the nonlinear processor has the power spectra shown in Figure 2.7 [7].
Figure 2.3: Clock Recovery Circuit with Nonlinear Processor

Figure 2.4: Simple Nonlinear Processor

Figure 2.5: Timing Diagram for Simple Nonlinear Processor
Figure 2.6: Power Spectra of Random NRZ Data before Nonlinear Processing

Figure 2.7: Power Spectra of Random NRZ Data after Nonlinear Processing

As one can see, the processed signal has a discrete spectral component at the bit rate of the data. The randomness of the data is responsible for the continuous spectral energy present in the signal. The degree to which this continuous spectral energy can be filtered out is determined by the $Q$ of the resonant circuit. It is important to realize,
however, that finite Q does not necessarily contribute to jitter in the recovered clock. If the resonant circuit is perfectly tuned, the residual continuous spectral energy will amplitude modulate the filter's sinusoidal output, but the zero crossings of the filter's output will not be affected [7]. In this way the output of the hard limiter will be a jitter free square wave at the bit rate.

Due to phase shifts and propagation delays through the circuit, the output of the hard limiter will not have the correct phase relationship with the incoming data. In practice, a phase shifter is employed to align the output of the hard limiter with the incoming data (see Figure 2.3).

2.2.2 Tradeoff Between Static Phase Error and Jitter

Tuning errors after manufacturing are inevitable in a resonant circuit because of aging and temperature drifts. Near resonance, the phase shift through a tuned circuit is a strong function of frequency. Any tuning error will translate directly into a static phase error. For moderate tuning errors, the static phase error \( \phi_e \) equals (in radians) [8]

\[
\phi_e = 2Q \frac{\Delta f}{f_o}
\] (2.1)

where \( \frac{\Delta f}{f_o} \) is the fractional tuning error. As one might expect, the phase shift produced by mistuning is magnified by the Q of the resonant circuit. This suggests that static phase errors can be minimized by using low Q resonant circuits.

However, low Q resonant circuits have poor noise filtering properties. Assuming the resonant circuit can be modelled as a first-order bandpass filter, the jitter transfer function of the clock recovery circuit equals [2]

\[
H(s) = \frac{1}{\tau s + 1}, \text{ where } \tau = \frac{Q}{\pi f_o}
\] (2.2)

The ability of the tuned circuit to filter out high frequency jitter improves as Q is increased. Hence, minimization of jitter on the recovered clock conflicts directly with minimization of static phase error. Typically, static phase error considerations limit the Q that can be used. As a result, clock recovery circuits based on bandpass filters usually have more jitter on the recovered clock than do narrowband PLL-based clock recovery circuits [9].

The first-order transfer function presented in Equation 2.2 suggests that it is not possible to get jitter peaking with tuned circuit clock recovery. More realistic models of bandpass filters show that jitter peaking is possible with tuned circuits but can be avoided through proper design of the bandpass filter [2].
2.2.3 Acquisition Time

Unless special techniques are employed, the acquisition time of a tuned circuit clock recovery system is directly related to the Q of the bandpass filter. Assume, for the moment, that energy from the preceding burst of data remains in the tuned circuit. The phase of the recovered clock will not settle to its final value until the energy from the preceding burst has decayed away. Of course, the decay rate of energy storage in a resonant circuit is controlled by the circuit's Q. Acquisition time requirements often necessitate a reduction in Q and compromise the jitter filtering properties of the clock recovery circuit.

On the other hand, it is interesting to consider the acquisition time if no stored energy exists prior to acquisition. For perfectly tuned lumped element filters, there is no long phase transient; the Q of the tuned circuit inhibits the amplitude of the filter’s sinusoidal output from building up quickly, but the locations of the output’s zero crossings are instantaneously correct. Since the hard limiter ideally responds only to the zero crossings, phase acquisition is much faster than one might expect from considerations of the filter’s bandwidth. This phenomenon can be exploited if a “quenching” circuit is used to discharge the filter before each acquisition [6]. Such techniques improve the tradeoff between rapid acquisition and jitter filtering when lumped element filters are used. Because of its distributed element nature, a SAW device cannot not be “discharged” quickly.

2.2.4 Cost

Perhaps the greatest disadvantage of tuned circuit clock recovery is its cost. Bandpass filters with excellent frequency stability over time and temperature are required, and the phase alignment shown in Figure 2.3 must usually be performed manually [10,11]. As a result, tuned circuit clock recovery is significantly more expensive than clock recovery based upon monolithic phase-locked loops.
2.3 Clock Recovery Using Phase-Locked Loops

2.3.1 Basic Operating Principle

Because of its amenability to monolithic construction, a phase-locked loop (PLL) is an attractive alternative to tuned circuit clock recovery. A simple block diagram for a PLL is presented in Figure 2.8. Negative feedback phase locks the voltage-controlled

![Diagram](image)

Figure 2.8: Phase-Locked Loop Approach to Clock Recovery

oscillator (VCO) to the incoming data in the following manner. The phase detector compares the phase of the clock to that of the data. If the clock is behind the optimum sampling instant, the phase detector drives the VCO to a higher frequency. The faster clock then catches up to the data, eliminating the initial phase error. If the clock is ahead of the optimum sampling instant, the negative feedback simply acts in the opposite direction.

2.3.2 Static Phase Error

In the steady state, the phase error equals exactly that required to drive the VCO to the bit rate of the incoming data. There are two significant contributions to this static phase error. The first is the offset of the phase detector itself. Imperfections in real phase detectors produce d.c. outputs when the phase error between the data and the clock is zero. This error can only be reduced through proper design of the phase detector. A phase detector which directly compares the phases of NRZ data and clock with ultralow offset is described in Chapters 4 and 5.

The second contribution to static phase error is finite d.c. gain between the phase detector and the VCO. The linearized block diagram for a first-order PLL is

22
displayed in Figure 2.9. The PLL is first-order because there is one integrator in the loop transmission. This integrator is intrinsic to all PLL’s and reflects the integration relating phase to frequency. Because the gain between the phase detector and the VCO is finite in a first-order PLL, a constant or static phase error is needed to drive the VCO away from its center frequency. If the frequency of the incoming data deviates $\Delta f$ (in Hz) from the center frequency of the VCO, the static phase error (in radians) equals:

$$\phi_e = \frac{2\pi \Delta f}{K_D K_O}$$

(2.3)

The static phase error is inversely proportional to the gain $K_D K_O$.

However, increasing the gain $K_D K_O$ in order to minimize the static phase error reduces the PLL’s ability to filter out jitter. It is readily shown that the jitter transfer function for the first-order PLL equals

$$H(s) = \frac{\phi_{clk}(s)}{\phi_{data}(s)} = \frac{1}{\tau s + 1}, \text{ where } \tau = \frac{1}{K_D K_O}$$

(2.4)

Hence, the first-order PLL is afflicted by the same tradeoff between static phase error and jitter filtering as clock recovery circuits using bandpass filters (cf. Equations 2.1 and 2.2).

A second-order PLL, as represented in Figure 2.10, avoids this tradeoff. An integrator is placed between the phase detector and the VCO; a zero is needed in the loop filter to stabilize the loop. If the frequency of the incoming data deviates from the center frequency of the VCO, no static phase error results, for the integrator has
infinite d.c. gain. The jitter transfer function for this PLL is

\[ H(s) = \frac{\phi_{\text{clk}}(s)}{\phi_{\text{data}}(s)} = \frac{K(1 + \tau s)}{s^2 + K\tau s + K}, \text{ where } K = K_D K_C \] (2.5)

As one can see, the jitter transfer function of the second-order PLL has two poles and one zero. The bandwidth of the transfer function can be adjusted by changing the time constant \( \tau \) of the zero. In this way very narrowband PLL’s are feasible. The independence of static phase error and jitter filtering is the most important advantage of PLL’s over tuned circuit clock recovery [8].
2.3.3 Jitter Transfer Function and Jitter Peaking

A root locus diagram provides insight into the dynamics of a second-order PLL (see Figure 2.11). The stability of the PLL is determined by the gain around the loop.

For low values of $K$, the closed loop poles are complex, while for high values of $K$, the poles are overdamped. For $K = 4/\tau^2$, the poles are critically damped. The closed loop poles, of course, determine the natural frequencies of the closed loop system.

One might think that as long as the poles are sufficiently damped, jitter peaking would not be present in a second-order loop. But the zero in the closed loop transfer function (Equation 2.5) cannot be neglected. The jitter transfer function, when $\tau = 10\mu s$ and the poles are critically damped, is plotted in Figure 2.12; the 1.25 dB peak in the transfer function is due to the zero.

Note that one closed loop pole approaches the zero as the loop is made more and more overdamped (see Figure 2.11). The closer the pole is to the zero, the smaller the jitter peaking will be; this is a result of pole-zero cancellation. Jitter peaking can be minimized by heavily overdamping the loop, but not eliminated entirely [8,12]. In practice, the degree to which jitter peaking can be reduced is often limited by higher order poles (not modelled here). Because of the second-order PLL's problems with jitter peaking, SAW devices are often preferred to PLL's in long repeater chains [2].
Figure 2.12: Jitter Transfer Function for a Critically Damped PLL
2.3.4 Acquisition Time

The presence of the integrator between the phase detector and the VCO complicates acquisition. While a first-order PLL only has memory of phase, a second-order PLL has memory of phase and frequency. For this reason, the acquisition process of a second-order PLL has two distinct stages: a frequency acquisition stage, during which the frequency of the clock is adjusted toward the bit rate of the data, and a phase acquisition stage, during which the phase error between the clock and the data is driven to zero.

Because the limited linear range of practical phase detectors, the mathematics of frequency acquisition is highly nonlinear and quite formidable [13]. While a detailed analysis of frequency acquisition is beyond the scope of this cursory review, it suffices to say that frequency acquisition is more time consuming and difficult than phase acquisition. Indeed, frequency acquisition is so unreliable that most PLL's require frequency acquisition aids [14,15,16]. While acquisition aids can make frequency acquisition reliable, frequency acquisition remains slower than phase acquisition.

For this reason, the frequency acquisition process must be avoided when very fast acquisition is required (as in burst mode communications). That is, one must ensure that at the start of acquisition, the frequencies of the VCO and the incoming data do not differ significantly. In almost all communications systems, the bit rate of the incoming data is crystal-controlled and relatively constant. The real challenge, then, is to control the VCO's frequency at the start of acquisition.

One obvious, but self-defeating, method of controlling the VCO's frequency is to employ a voltage-controlled crystal oscillator (VCXO). VCXO's have excellent center frequency stability (e.g., ±50 ppm) and very narrow tuning ranges (e.g., ±100 ppm); thus, the frequency of a VCXO cannot be pulled significantly from the bit rate of the data. While this property of VCXO's eliminates the problems associated with frequency acquisition, it hinders phase acquisition. Because a VCXO cannot be pulled very far from the bit rate, phase errors between the data and the clock cannot be reduced quickly. Thus, VCXO's cannot be utilized when rapid acquisition is desired.

Relaxation and ring oscillators, in contrast, have very wide tuning ranges (e.g., ±25%). Between bursts of data, the frequency of these oscillators can drift very far from the bit rate of the data. Special precautions must be taken in order to avoid the time-consuming frequency acquisition process at the start of each burst. One technique which can be used is to lock the VCO to an asynchronous crystal reference between data bursts. When a new data burst is detected, the PLL is reconnected to the incoming data, and only the phase acquisition process takes place. Because of the wide tuning range of the VCO, phase acquisition can be quite fast - as fast as the loop bandwidth of the PLL.

Once again, there is a tradeoff between jitter filtering and phase acquisition. Faster
acquisition necessitates a higher loop bandwidth, which diminishes the PLL’s ability to filter out jitter. This tradeoff is not as favorable as the one presented by “quenched” tuned circuits, for “quenched” tuned circuits acquire faster than their filter bandwidth might suggest.

It must be mentioned that there is considerable risk in utilizing an asynchronous reference on an integrated circuit. It is possible for injection locking to occur, in which parasitic signal injection causes the VCO to lock to the asynchronous reference instead of the incoming data [17]. Even if injection locking does not occur, jitter may be created by asynchronous interference [18]. At high frequencies, such coupling can occur through the power supplies, through off-chip mutual inductances, or even through the substrate itself [19]. Unfortunately, if rapid acquisition is desired, the standard PLL requires an asynchronous reference.

2.4 Prospectus

In this chapter, the advantages and disadvantages of existing clock recovery architectures have been reviewed. In the next chapter, two new architectures for clock recovery are presented, both of which combine the advantages of tuned circuit clock recovery with the advantages of PLL-based clock recovery. In particular, the new architectures promise simultaneously:

1. Rapid acquisition approaching that of “quenched” tuned circuits.
2. Absolutely no jitter peaking, like properly designed tuned circuits.
3. Very low static phase error, like second-order PLL’s.
4. Excellent jitter filtering, like very narrowband PLL’s.
5. Amenability to monolithic construction.
Chapter 3

The Delay-Locked Loop and the Delay-and-Phase-Locked Loop

This chapter introduces two new architectures for clock recovery: the delay-locked loop (DLL), and the delay-and-phase-locked loop (D/PLL). It is heuristically useful to examine the DLL first, for the D/PLL can be viewed as a generalization of the DLL architecture.

3.1 The Delay-Locked Loop

3.1.1 Need for Delay-Locked Loop

A DLL can be used for clock recovery when an exact frequency reference is available. This section describes a typical system architecture in which a DLL can be used effectively.

A bus architecture for serial data communications is depicted in Figure 3.1. The central processor continuously transmits data to the peripheral units over the read bus; each unit is accessed through appropriate addressing. The bit rate on the read bus is established by a crystal reference inside the central processor. Since the data stream on the read bus is never interrupted, the clock recovery circuits in the peripheral units remain permanently locked to the data on the read bus. Because acquisition time is not critical, conventional narrowband PLL's can be used for clock recovery in the peripheral units.

The peripheral units send data back to the central processor over the write bus; the units share one bus through time multiplexing. Since each peripheral unit is synchronized to the data on the read bus, the bit rate on the write bus exactly equals the bit rate on the read bus. Because of propagation delays through the system, the data arriving at the central processor exhibits a phase shift with respect to the
original crystal reference; consequently, clock recovery is also needed in the central processor.

In this case, though, the clock recovery circuit must acquire very quickly. Because the peripheral units are not equidistant from the central processor, the data arriving at the central processor exhibits a phase jump each time a different peripheral unit begins to send data. The clock recovery circuit must reacquire at the start of each data burst. Typically, each data burst includes a preamble code, during which acquisition is supposed to take place. Since a preamble code does not convey useful information, it is usually quite short (e.g. 36 bits).

As discussed in the foregoing chapter, PLL's are troublesome when rapid acquisition is a priority. "Quenched" tuned circuits can acquire quickly but are more expensive than monolithic clock recovery circuits. A DLL promises rapid acquisition and compatibility with integrated circuit technology.

3.1.2 Basic Operating Principle

In the bus architecture described above, if the phase shift between the incoming data and the crystal reference were eliminated, the crystal reference could be used to retiming the incoming data. A DLL employs a voltage-controlled phase shifter to achieve phase alignment between the incoming data and an exact frequency reference. The block diagram for a DLL is presented in Figure 3.2.

The loop operation is similar to that of a PLL. The phase detector compares the phase of the clock to that of the data. If the clock is ahead of the data, the phase detector increases the delay through the phase shifter until the clock is no longer ahead of the data. If the clock is behind the data, the phase detector decreases the
delay through the phase shifter until the clock is no longer behind the data.

The DLL architecture illustrated in Figure 3.2 is not original; phase shifting a clock for data synchronization is a well-known technique [20]. A more innovative DLL architecture is portrayed in Figure 3.3. In this architecture the data is phase shifted instead of the frequency reference.

The loop operation is similar to that of the previous DLL, except that in this DLL the phase detector compares the phase of the clock to that of the delayed data. If the delayed data is ahead of the clock, the phase detector increases the delay through the phase shifter until the delayed data is no longer ahead of the clock. If the delayed data is behind the clock, the phase detector decreases the delay through the phase shifter until the delayed data is no longer behind the clock.

Phase shifting the data instead of the frequency reference is highly preferable because it greatly reduces the jitter on the retimed data. In general, the jitter on the retimed data equals the jitter on the recovered clock (since the retimed data is gated by the recovered clock). When the incoming data is phase shifted, as depicted in Figure 3.3, the recovered clock is the frequency reference. In this case, the jitter on the retimed data will be as low as that of the crystal reference. This important benefit will not be obtained if the frequency reference is phase shifted, for the recovered clock will then tend to track the jitter of the incoming data.

It must be noted, however, that phase shifting the data instead of the frequency reference does not improve the ability of the decision making circuit to retime the data correctly. The bit error rate (BER) of the receiver is only a function of how well the data and the clock are aligned at the decision making circuit. It is easily
Figure 3.3: Improved DLL for Clock Recovery

demonstrated that the two DLL's presented in Figures 3.2 and 3.3 are completely equivalent in terms of their ability to align the data and the clock at the decision making circuit. On the other hand, phase shifting the data instead of the frequency reference does improve the BER of subsequent regenerators by reducing the jitter on the retimed data.

In the remainder of this thesis, the term “Delay-Locked Loop (DLL)” refers to the improved architecture of Figure 3.3.

3.1.3 Static Phase Error

As in PLL's, there are two significant contributions to static phase error. The first is the offset of the phase detector itself. As mentioned earlier, this error can only be reduced through careful design of the phase detector. A phase detector with ultralow offset is described in Chapters 4 and 5.

The second contribution to static phase error is finite d.c. gain between the phase detector and the voltage-controlled phase shifter. The steady state value of the phase shifter's control voltage depends upon the phase relationship between the incoming data and the frequency reference. Any deviation from the center of the control voltage's range requires a d.c. phase error if the d.c. gain is finite.

This second contribution to static phase error can be eliminated by placing an integrator between the phase detector and the voltage-controlled phase shifter, for an integrator has infinite d.c. gain. The linearized block diagram for such a loop is presented in Figure 3.4. The integrator makes the DLL a first-order loop.
3.1.4 Jitter Transfer Function

As discussed above, in the DLL architecture the jitter on the retimed data is only that of the crystal reference and does not depend upon the jitter of the incoming data. By definition, then, the jitter transfer function equals zero:

\[ H(s) = \frac{\phi_{\text{clk}}(s)}{\phi_{\text{data}}(s)} = 0 \]  

(3.1)

It follows that jitter peaking is not a problem with the DLL. Nevertheless, DLL's are not generally useful in repeater chains, for each DLL requires the exact frequency reference. When the distances between successive repeaters are large, it is not economical to provide each repeater with the exact frequency reference.

3.1.5 Acquisition Time

Since the bit rate of the incoming data exactly matches the frequency reference, frequency acquisition is not required in a DLL. Acquisition in a DLL requires only phase acquisition, which is as fast as the loop bandwidth of the DLL.

At the start of acquisition, an initial phase error exists between the delayed data and the frequency reference. This initial phase error can range from \(-\pi\) to \(\pi\) radians\(^1\).

\(^1\)Phase errors can almost always be considered modulo \(2\pi\), for most phase detectors are not sensitive to \(2\pi\) changes in phase error. The transfer characteristic of a practical "sawtooth" phase detector is presented in Figure 4.12.
for the propagation delays through a high speed communications system are normally much longer than a clock period. (Throughout this thesis, one clock period = 2\pi radians = 360 degrees.) During acquisition, this initial phase error decays exponentially:

$$\phi_{error}(t) = \phi_o \exp(-K_D K_\phi t)$$ (3.2)

where $\phi_o$ equals the initial phase error and $-\pi \leq \phi_o \leq \pi$.

Clearly, increasing the loop bandwidth $K_D K_\phi$ makes the DLL acquire more quickly. Unlike the case of a PLL, a higher loop bandwidth does not diminish the DLL's ability to filter jitter; the jitter transfer function of the DLL equals zero, regardless of the loop bandwidth.

Nevertheless, the loop bandwidth of the DLL should not be increased unnecessarily. Most practical phase detectors\(^2\) generate outputs which have significant a.c. ripple at the clock frequency (or a multiple of the clock frequency). If the ripple were allowed to reach the voltage-controlled phase shifter, significant jitter would be added to the delayed data. While jitter on the delayed data does not yield jitter on the retimed data, it does degrade the ability of the decision making circuit to retime the data correctly. For this reason, the DLL must include additional filtering poles between the output of the phase detector and the voltage-controlled phase shifter. However, in a high bandwidth loop, too much filtering will degrade the stability of the loop. Hence, in order to minimize the jitter added to the delayed data, the loop bandwidth should be no higher than that required by acquisition.

### 3.1.6 Need to Reset the DLL

In general, any real voltage-controlled phase shifter will have finite range. This presents a potential problem for the DLL, for successive acquisitions may drive the voltage-controlled phase shifter to the end of its control range.

This issue is best explained through an example. Consider the bus architecture depicted in Figure 3.1. Imagine that each time a different peripheral unit begins to send data, the phase jumps ahead $\pi/2$ radians; this is possible if the distances between successive peripheral units are chosen correctly. At the start of each acquisition, the phase detector observes that the delayed data is $\pi/2$ radians ahead of the clock. In order to reduce this phase error, the control voltage of the phase shifter is adjusted to delay the incoming data an additional $\pi/2$ radians. Since this same adjustment is made upon each acquisition, the voltage-controlled phase shifter will eventually be driven to the end of its range. Once the phase shifter has been driven to the end of its range, acquisition is prematurely halted, and the entire data burst is lost.

\(^2\)Phase detectors which use sample-and-holds can suppress ripple almost completely [21], but their performance at high bit rates is limited by the difficulty of building very high frequency sample-and-holds.
This predicament can be avoided by "resetting" the DLL. Before acquisition begins, the integrator can be initialized to the midpoint of the phase shifter's range. Since the initial phase error lies between $-\pi$ and $\pi$ radians, the phase shifter will not be driven more than $\pm \pi$ radians from the middle of its range. Hence, acquisition will always be successful as long as the total range of the voltage-controlled phase shifter exceeds $2\pi$ radians.

In practice, a range considerably larger than $2\pi$ radians is desirable, for jitter on the incoming data can temporarily drive the phase shifter more than $\pm \pi$ radians from the middle of its range. Additional range is also needed if the integrator is not initialized to the exact midpoint of the control range. For most applications, a total range of $3\pi$ radians is sufficient.

Once acquisition has occurred, the integrator must not be reset until the end of the data burst. Otherwise, part of the data burst will be lost. The reset circuitry must only be activated between data bursts. Deciding when to reset the integrator could be a challenging control problem. Fortunately, in burst mode communications systems, a binary signal named signal_detect is usually available. This signal is asserted when a data burst is present. Thus, this signal can be used to deactivate the reset circuitry of the DLL.

### 3.2 The Delay-and-Phase-Locked Loop

#### 3.2.1 Basic Operating Principle

A delay-and-phase-locked loop (D/PLL) is an interesting cross between a DLL and a PLL. Unlike a DLL, it can be used when no exact frequency reference is available (e.g., in a repeater chain).

The block diagram for a D/PLL is presented in Figure 3.5. A D/PLL contains two parallel control loops. The phase detector, loop amplifier, and voltage-controlled crystal oscillator (VCXO) form the core of a PLL, while the phase detector, loop amplifier, and voltage-controlled phase shifter form the core of a DLL.

Negative feedback phase-locks the clock (the output of the phase shifter) to the incoming data. The phase detector compares the phase of the clock to that of the data. If the clock is behind the data, the phase detector drives the VCXO to a higher frequency and, at the same time, decreases the delay through the voltage-controlled phase shifter. Both of these actions serve to reduce the initial phase error. If the clock is ahead of the delayed data, the negative feedback simply acts in the opposite direction.

As in the DLL, the phase shifter can also be applied to the incoming data. Such an architecture is illustrated in Figure 3.6. In this loop the phase detector compares the phase of the clock to that of the delayed data. If the clock is behind the delayed
data, the phase detector drives the VCXO to a higher frequency and, at the same time, increases the delay through the voltage-controlled phase shifter. The faster clock picks up phase, while the delayed data loses phase. Eventually, the initial phase error is reduced to zero.

Phase shifting the data instead of the VCXO’s output greatly improves the D/PLL’s ability to filter jitter, as the next section explains.
3.2.2 Loop Dynamics of D/PLL

Consider the D/PLL portrayed in Figure 3.5. If the loop amplifier is made an integrator (in order to minimize static phase error), the linearized block diagram of the system becomes:

![Diagram of D/PLL with integrator]

**Figure 3.7: D/PLL with Integrator**

Rearranging the blocks in the figure, one obtains:

![Revised block diagram with mathematical expression]

**Figure 3.8: Illustration of Zero in Loop Transmission of D/PLL**

It is enlightening to compare Figure 3.8 with Figure 2.10. If one associates \( \tau \)
with the ratio \( K_\phi/K_O \), the two block diagrams are completely equivalent. One can consider the phase shifter in Figure 3.7 as the implementation of the zero in Figure 2.10.

The jitter transfer function of this D/PLL is identical to that of the second-order PLL (cf. Equation 2.5):

\[
H(s) = \frac{\phi_{\text{clk}}(s)}{\phi_{\text{data}}(s)} = \frac{K(1 + \tau s)}{s^2 + K\tau s + K}
\]

(3.3)

where \( K = K_D K_O \) and \( \tau = K_\phi/K_O \). As discussed in the previous chapter, the presence of the zero in the jitter transfer function results in jitter peaking.

While the two loops are equivalent in their small signal dynamics, they are quite different in their large signal behavior. Because a VCXO cannot be pulled very far from its center frequency, the phase error between the clock and the data cannot be reduced very quickly in a conventional PLL. In this D/PLL, on the other hand, there is no such limit, for the phase shifter allows the phase of the recovered clock to be changed almost instantaneously. Hence, the phase acquisition of this D/PLL can be significantly faster than that of a PLL with VCXO.

The D/PLL portrayed in Figure 3.6 offers improved jitter filtering. If the loop amplifier is an integrator, the linearized block diagram of the improved D/PLL becomes:

![Figure 3.9: Improved D/PLL with Integrator](image)

From the block diagram, one observes:

\[
v_1 = \frac{K_D}{s} (\phi_{\text{data}} - K_\phi v_1 - \phi_{\text{clk}})
\]

(3.4)

38
\[ \phi_{\text{clk}} = v_1 \frac{K_O}{s} \]  

(3.5)

Eliminating \( v_1 \), one finds the jitter transfer function:

\[ H(s) = \frac{\phi_{\text{clk}}(s)}{\phi_{\text{data}}(s)} = \frac{K}{s^2 + K\tau s + K} \]  

(3.6)

where \( K = K_D K_O \) and \( \tau = K_\phi/K_O \). Comparing Equations 3.3 and 3.6, one sees that the improved system has the same closed loop poles as the standard second-order PLL (and hence the same stability) but does not have the closed loop zero. Thus, as long as one picks \( K \) and \( \tau \) so that the loop has a damping ratio \( \zeta \geq \sqrt{2}/2 \), the improved D/PLL exhibits no jitter peaking.

Elimination of the zero also lowers the bandwidth of the jitter transfer function — especially if the loop is highly overdamped. Equations 3.3 and 3.6 are plotted in Figure 3.10 for the case of \( \tau = 150\mu s \) and \( \zeta = 12 \). The bandwidth of the transfer function with the zero is predominantly controlled by the high frequency pole at 610KHz, for the zero effectively cancels the low frequency pole at 1.1KHz. In contrast, the bandwidth of the transfer function without the zero is predominantly controlled by the low frequency pole at 1.1KHz. (This distinction is most prominent at high damping ratios, where the two poles are widely separated in frequency.) The implication is that for the same loop bandwidth (and thus the same acquisition speed), the improved D/PLL provides much more jitter filtering.

It is not surprising that phase shifting the data instead of the VCXO’s output reduces the jitter on the retimed data. When the incoming data is phase shifted, as depicted in Figure 3.6, the jitter on the retimed data equals the jitter on the output of the VCXO. When the output of the VCXO is phase shifted, as depicted in Figure 3.5, the jitter on the retimed data equals the jitter on the output of the VCXO plus the jitter added by the phase shifter. (Note the parallelism between this discussion and the discussion given for the DLL in Section 3.1.2.) Equations 3.3 and 3.6 merely quantify this difference.

In the remainder of this thesis, the term “Delay-and-Phase-Locked Loop (D/PLL)” refers to the improved architecture of Figure 3.6.

### 3.2.3 Acquisition

The closed loop poles of the D/PLL can be found by solving the system’s characteristic equation:

\[ s^2 + K\tau s + K = 0 \]  

(3.7)

Applying the quadratic formula yields:

\[ s_1, s_2 = \frac{-K\tau \pm \sqrt{(K\tau)^2 - 4K}}{2} \]  

(3.8)
Figure 3.10: Jitter Transfer Function, With and Without Zero
Assume now that the D/PLL is highly overdamped. (The advantages of overdamping the loop will be revealed shortly.) Under this assumption, $K \tau \gg 1/\tau$, and one can make the approximation:

$$\sqrt{(K\tau)^2 - 4K} \approx K\tau - \frac{2}{\tau}$$  \hspace{1cm} (3.9)

Equation 3.8 then becomes:

$$s_1 \approx -\frac{1}{\tau} = -\frac{K_o}{K_\phi}$$  \hspace{1cm} (3.10)

$$s_2 \approx -(K\tau - \frac{1}{\tau}) \approx -K\tau = -K_D K_\phi$$  \hspace{1cm} (3.11)

During acquisition, the phase error decays to zero with the natural frequencies of the closed loop system:

$$\phi_{error}(t) = A \exp s_1 t + B \exp s_2 t$$  \hspace{1cm} (3.12)

where $A$ and $B$ are determined by initial conditions. Since the D/PLL is a second-order system, two initial conditions must be specified: the initial phase error and the initial frequency error. If the initial phase error equals $\phi_0$, and the initial frequency error equals $\Delta \omega$, it can shown that Equation 3.12 becomes:

$$\phi_{error}(t) = \left(\frac{K_D K_\phi + s_2}{s_2 - s_1} \phi_0 + \frac{\Delta \omega}{s_2 - s_1}\right) \exp s_1 t + \left(-\frac{K_D K_\phi + s_1}{s_2 - s_1} \phi_0 - \frac{\Delta \omega}{s_2 - s_1}\right) \exp s_2 t$$

Combining Equations 3.10, 3.11, and 3.13, and noting that $|s_2| \gg |s_1|$, one can write:

$$\phi_{error}(t) \approx \left(-\frac{1}{\tau K_D K_\phi} \phi_0 - \frac{\Delta \omega}{K_D K_\phi}\right) \exp\left(-\frac{t}{\tau}\right) + \left(\phi_0 + \frac{\Delta \omega}{K_D K_\phi}\right) \exp\left(-K_D K_\phi t\right)$$

(3.14)

The first of these exponential terms hinders rapid acquisition; because of its presence, the phase error settles to zero with a slow “tail”.

Fortunately, this tail can be neglected if the first term's coefficient is sufficiently small (e.g., \(\leq 2\) degrees). This coefficient has two components – one due to the initial phase error $\phi_0$ and one due to the initial frequency error $\Delta \omega$. The latter component is negligibly small if:

$$\Delta \omega \ll K_D K_\phi$$  \hspace{1cm} (3.15)

Because a VCXO has excellent center frequency stability and a very narrow tuning range, the initial frequency error $\Delta \omega$ is very small (e.g. 200 ppm of the bit rate). $K_D K_\phi$, which approximately equals the crossover frequency of the D/PLL, can be
much larger (e.g. 2% of the bit rate). If $\Delta \omega$ is 200 ppm of the bit rate, and $K_D K_\phi$ is 2% of the bit rate, the component of the tail due to the initial frequency error equals:

$$\frac{\Delta \omega}{K_D K_\phi} = 0.01 \text{ radians} = 0.57 \text{ degrees}$$  \hspace{1cm} (3.16)

In almost all applications, such an error is negligible.

The component of the tail due to the initial phase error is negligibly small if:

$$\frac{1}{\tau K_D K_\phi} \phi_o \ll 1$$  \hspace{1cm} (3.17)

Noting that $\tau K_D K_\phi$ equals $4\zeta^2$ ($\zeta$ is the damping ratio), one can rewrite the last inequality as:

$$\frac{1}{4\zeta^2} \phi_o \ll 1$$  \hspace{1cm} (3.18)

The desirability of overdamping the loop is now evident. Unlike the case of a second-order PLL, overdamping the D/PLL improves acquisition speed but does not affect the loop’s ability to filter jitter, as will be explained in the next section. If $\zeta$ equals 12 and $\phi_o$ equals $\pi$, the component of the tail due to the initial phase error is:

$$\frac{1}{4\zeta^2} \phi_o = \frac{\pi}{576} = 0.0055 \text{ radians} = 0.31 \text{ degrees}$$  \hspace{1cm} (3.19)

This error, too, is usually negligible.

Thus, if $K_D K_\phi$ is much larger than $\Delta \omega$, and the D/PLL is highly overdamped, Equation 3.14 can be simplified to:

$$\phi_{error}(t) \approx \phi_o \exp (-K_D K_\phi t)$$  \hspace{1cm} (3.20)

Under these conditions, the D/PLL acquires as quickly as a DLL (cf. Equation 3.2).

Further insight into the acquisition process can be gained by observing the transient on the control voltage of the D/PLL (node $v_1$ in Figure 3.9). For the sake of clarity, the response to an initial phase error and the response to an initial frequency error are now considered separately. (Assuming the system is linear, the response to an initial phase error and an initial frequency error can be obtained by superposition.)

Assume that the D/PLL has an initial phase error equal to $\phi_o$. The transient on the control voltage is plotted in Figures 3.11 and 3.12 for the case when $\tau = 150\mu s$ and $\zeta = 12$. In these figures the control voltage has been normalized so that its entire transient lies between 0 and 1 V.

The D/PLL’s response to an initial phase error has two distinct stages. During the first stage, the control voltage is quickly adjusted to a new value (1 V in Figure 3.11).
Figure 3.11: First stage of phase acquisition

Figure 3.12: Second stage of phase acquisition
Simultaneously, the phase shifter's delay and the VCXO's frequency are adjusted. Since the VCXO cannot be pulled very far from the bit rate of the incoming data, the phase of the VCXO does not change very rapidly. Consequently, the VCXO's phase can be considered stationary during the first stage of acquisition. The first stage of acquisition is, for all practical purposes, the acquisition process of a DLL\(^3\). The phase of the clock remains unaltered while the phase shifter reduces the initial phase error. Hence, it is not surprising that the phase error is reduced to zero as fast as the DLL bandwidth \(K_D K_o\) (Equation 3.20).

During the second stage of acquisition, the control voltage slowly decays from its new value to its original value (Figure 3.12). At the start of the second stage, the VCXO's frequency does not match the bit rate of the incoming data. (The VCXO's frequency only matches the bit rate of the data when the control voltage is at its original value.) Because of this frequency error, the phase of the VCXO with respect to the incoming data changes (albeit slowly). As the phase of the VCXO changes, the delay of the phase shifter is adjusted so that the phase of the delayed data tracks that of the VCXO. This is accomplished by shifting the control voltage back toward its original value. This process continues until the control voltage reaches its original value, at which point the VCXO's frequency equals the bit rate of the data, and acquisition is complete.

It must be stressed that the phase error between the delayed data and the clock may be insignificant during the second stage of acquisition. Examination of Figure 3.9 readily shows that the phase error of the D/PLL is proportional to the time derivative of the control voltage. Since the control voltage is changing during the second stage of acquisition, a nonzero phase error persists for a long time (note the first exponential term in Equation 3.14). But if the control voltage is changing very slowly, the magnitude of this slow tail can be negligible (see Equations 3.17 and 3.18). Under such conditions, the second stage of acquisition can be ignored, for the data can be successfully regenerated as soon as the first stage of acquisition is completed (Equation 3.20).

Now assume that this same D/PLL has an initial frequency error \(\Delta \omega\). The control voltage's response is plotted in Figure 3.13. Once again, the control voltage has been normalized so that its entire transient lies between 0 and 1 V.

Note the similarity between frequency acquisition (Figure 3.13) and the second stage of phase acquisition (Figure 3.12). Essentially, the second stage of phase acquisition is a kind of frequency acquisition; at the beginning of the second stage, the phase error is very small, but a frequency error \((\phi_o/\tau)\) exists between the VCXO's output and the incoming data. The D/PLL's response to a general frequency error

\(^3\)In fact, the DLL can be considered a D/PLL in which the gain \(K_o\) equals 0 (compare Figures 3.4 and 3.9). If \(K_o\) equals 0, the clock's phase is stationary.
\[ \Delta \omega \] can be derived from the phase acquisition results simply by replacing \( \phi_o / \tau \) with \( \Delta \omega \). Thus, an initial frequency error adds a slow tail to the phase error’s transient response (see Equation 3.14). As in the case of phase acquisition, this tail can be ignored if the initial frequency error is small enough (compare Equations 3.15 and 3.17).

### 3.2.4 Practical Limitations on Jitter Filtering

Unlike the second-order PLL, the D/PLL realizes rapid acquisition without compromising jitter filtering. While phase errors are nulled out as quickly as the DLL bandwidth \( K_D K_\phi \), the jitter transfer function's bandwidth is predominantly controlled by the low frequency pole at \( K_O / K_\phi \). Increasing \( K_D \) makes the D/PLL acquire more quickly, but does not diminish the D/PLL's ability to filter jitter.\(^4\)

In practice, the D/PLL's ability to filter jitter is limited by the finite range of the phase shifter, the variability of the incoming data's bit rate, and the instability of the VCXO's center frequency. Together, the variability of the bit rate and the instability of the VCXO's center frequency determine the VCXO's required tuning range, as the tuning range must be large enough to ensure that the VCXO's frequency can always

\(^4\)As in the DLL, in order to minimize the jitter added to the delayed data, one should not make the loop bandwidth \( K_D K_\phi \) higher than that required by acquisition.
be pulled to the bit rate of the incoming data. But the entire tuning range of the VCXO is only usable if the VCXO can be tuned over its entire range without driving the phase shifter out of range. (Recall that the VCXO and the phase shifter share the same control voltage.) As explained earlier, the phase shifter stabilizes the D/PLL by adding a zero to the D/PLL's loop transmission. If the phase shifter is driven to the end of its range, the loop transmission no longer has a zero, and the D/PLL becomes unstable.

Figure 3.14 provides an example of a D/PLL in which the entire range of the VCXO is not usable. While the VCXO can be pulled ±100 ppm from its center frequency, only ±50 ppm of the tuning range is actually useful. Tuning the VCXO more than ±50 ppm from its center frequency drives the phase shifter out of its linear range and thereby destabilizes the D/PLL.

In order to take advantage of the VCXO's entire tuning range, the VCXO's control range must be a subset of the phase shifter's control range, as shown in Figure 3.15. Note how the VCXO can be tuned over its entire range without driving the phase shifter out of range. In this case, the D/PLL can lock to any data stream whose bit rate is within the tuning range of the VCXO.

If the VCXO's control range is a subset of the phase shifter's control range, it follows that:

$$\frac{\Delta \omega}{K_O} \leq \frac{\Delta \phi}{K_\phi} \quad (3.21)$$

where $\Delta \omega$ equals the VCXO's required tuning range (in radians per second), and $\Delta \phi$ equals the phase shifter's total range (in radians). Rearranging terms, one obtains:

$$\frac{K_O}{K_\phi} \geq \frac{\Delta \omega}{\Delta \phi} \quad (3.22)$$

Equation 3.22 places a lower limit on the bandwidth of the jitter transfer function, as the bandwidth of the jitter transfer function is approximately $K_O/K_\phi$. If $\Delta \omega$ is 200 ppm of the bit rate, and $\Delta \phi$ is $3\pi$ radians, this lower limit equals:

$$\frac{\Delta \omega}{\Delta \phi} = 21 \text{ ppm of bit rate} = 0.0021\% \text{ of bit rate} \quad (3.23)$$

Such a low bandwidth accommodates exceptionally good jitter filtering.

On the other hand, if the voltage-controlled oscillator were not a VCXO, but a relaxation oscillator, for example, the D/PLL's ability to filter jitter would be greatly diminished. Because of its poor center frequency stability, a relaxation oscillator's required tuning range is much larger than that of a VCXO (e.g. ±25% versus ±100 ppm). But, according to Equation 3.22, a large required tuning range precludes good jitter filtering. For instance, if $\Delta \omega$ is 50% of the bit rate, and $\Delta \phi$ is $3\pi$ radians, the
Figure 3.14: Transfer Characteristics of VCXO and Phase Shifter

Figure 3.15: Transfer Characteristics Permitting the Use of the VCXO's Entire Range
lower limit of the bandwidth of the jitter transfer function is 2500 times larger than the limit given in Equation 3.23. Because its narrow tuning range improves jitter filtering, a VCXO is highly desirable in a D/PLL.

3.2.5 Need to Reset the D/PLL

The finite range of the voltage-controlled phase shifter necessitates resetting the D/PLL before each acquisition – at least if rapid acquisition is desired. Normally, initial phase errors are rapidly nulled out during the first stage of phase acquisition. But the first stage of phase acquisition, which is basically the acquisition process of a DLL, can be prematurely halted if the voltage-controlled phase shifter is driven to the end of its range.

Successive acquisitions can drive the phase shifter to the end of its range in exactly the same manner as in the DLL. Imagine that the time between consecutive acquisitions is much shorter than \( \tau \), the ratio of \( K_\phi \) to \( K_O \). In this case, the decay of the control voltage between consecutive acquisitions (see Figures 3.12 and 3.13) can be ignored, for the time constant of the decay is approximately equal to \( \tau \). If the decay between consecutive acquisitions is negligible, successive acquisitions can repeatedly increment (or decrement) the control voltage until the phase shifter is at the end of its range.

As in the DLL, initializing the D/PLL's integrator to the midpoint of the phase shifter's range eliminates this problem, provided that the total range of the phase shifter is large enough. For most applications, a total range of \( 3\pi \) radians is sufficient.

On the other hand, it is not necessary to reset the D/PLL if rapid acquisition is not required. As long as the bit rate of the incoming data lies within the tuning range of the VCXO, the D/PLL eventually acquires. If the phase shifter has been driven to the end of its range, the first stage of phase acquisition is prematurely halted, and a significant phase error may remain. But if the phase shifter is at the end of its range, the VCXO must also be at the end of its range. (Remember, the VCXO's control range is a subset of the phase shifter's control range.) The resulting frequency error between the VCXO and the incoming data slew the phase error toward zero in an open loop manner. Once the phase error is reduced to zero, closed loop behavior resumes, and acquisition continues in the manner of Figure 3.13. In the end, the control voltage settles to the exact value needed to drive the VCXO's frequency to the bit rate of the data.

The D/PLL's ability to acquire without being reset is valuable when the incoming data stream is rarely interrupted, and acquisition time is not critical. Under such circumstances, it is often difficult to decide when to reset the integrator, as a signal detect bit is rarely available.

48
3.3 Specifications for DLL and D/PLL Prototypes

This chapter has presented a basic overview of the DLL and D/PLL architectures. The remainder of this thesis is concerned with the detailed implementation of a DLL and a D/PLL which meet the following specifications:

1. The DLL and D/PLL accept incoming NRZ data clocked at 40 Mbit/s; this implies that the clock period equals 25 ns.

2. The transition density of the data may be as low as 1 transition every 4 clock periods.

3. Each data burst includes a preamble code, which consists of alternating ones and zeros. Because the preamble code is only 40 bits long, the time allotted for acquisition is 1 μs (40 × 25 ns). A signal detect bit is available.

4. In both the DLL and the D/PLL, the static phase error is less than 1 ns.

5. For the D/PLL, the bandwidth of the jitter transfer function is less than 1.5 KHz. The jitter transfer function of the DLL simply equals zero. Of course, neither the DLL nor the D/PLL exhibits jitter peaking.
Chapter 4

Phase Detectors

The phase detector is a critical component in a DLL or a D/PLL. Imperfections in
the phase detector often limit the performance of the entire clock recovery circuit.
If the loop amplifier is an ideal integrator, the phase detector's offset is the only
cause of static phase error. Ripple on the phase detector's output can add significant
jitter to the delayed data, thereby degrading the ability of the decision making circuit
to retime the data correctly. In the case of a D/PLL, jitter can also be added to the retimed data, for ripple on the phase detector's output can result in unwanted
frequency modulation of the VCXO.

This chapter examines the performance limitations of three phase detectors which
are suitable for use in a DLL or a D/PLL\(^1\). For the reasons discussed above, particular
attention is given to reducing the phase detector's offset and to minimizing the ripple
on the phase detector's output.

4.1 The Multiplier-Based Phase Detector

4.1.1 Basic Operating Principle

Because of its simple implementation, the multiplier-based phase detector has been
used extensively in clock recovery PLL's [9,19,22]. The basic structure of the multiplier-
based phase detector is presented in Figure 4.1. The nonlinear processor is needed
because, in most signalling formats, the data signal itself has no spectral energy at
the clock frequency. As in the case of tuned circuit clock recovery, the nonlinear pro-
cessing circuitry can be as simple as a delay line and one exclusive OR (XOR) logic
gate (see Figures 2.4 – 2.7). The multiplier itself can be implemented as a single-
balanced or double-balanced modulator. A single-balanced modulator is preferable

\(^1\)Many phase detectors do not function properly in clock recovery applications, as they are conf-
fused by missing transitions in the data pattern.
if the duty cycle of the clock is not exactly 50%; asymmetry in the clock waveform causes data-dependent jitter if a double-balanced modulator is used (see Appendix A). In the main body of this thesis, only symmetrical clock waveforms are considered. Appendix A examines the effect of clock asymmetry on phase detector offset for several different phase detectors. The ensuing discussion assumes that the multiplier is a single-balanced modulator; nonetheless, if the multiplier is a double-balanced modulator, the same arguments apply.

Figure 4.2 is the timing diagram for the multiplier-based phase detector, under the assumption that the nonlinear processor is implemented as a delay line and an XOR logic gate. Because the multiplier is a single-balanced modulator, the output of the nonlinear processor is treated as a unipolar signal (i.e., 1 or 0), while the clock is treated as a bipolar signal (i.e., 1 or -1)². (A single-balanced modulator is essentially an overdriven two-quadrant analog multiplier.) As one can see, the output of the phase detector has zero average value; consequently, there is no net change in the loop integrator's output. (Recall that, in both the DLL and the D/PLL, the loop amplifier is an integrator). It should be noted that the delayed data and the clock are not optimally aligned in Figure 4.2, since the rising edge of the clock is not centered within the delayed data's bit cell. Actually, the delayed data is $\pi/2$ radians ahead of the clock in Figure 4.2. That is to say, the multiplier-based phase detector has an offset of $\pi/2$ radians; the consequences of such an offset are discussed in the next section.

²There is no loss of generality in assuming the signal amplitudes displayed in Figure 4.2. While the signal amplitudes do affect the gain of the phase detector, a change in phase detector gain can always be compensated by adjusting the gains of the phase shifter and, in the case of a D/PLL, the VCO.
Figure 4.2: Timing Diagram for the Multiplier-Based Phase Detector

Figure 4.3: Timing Diagram: Delayed Data Advanced
If the delayed data is slightly more than $\pi/2$ radians ahead of the clock, the output of the phase detector has a positive average value, as shown in Figure 4.3. In this case, the loop integrator's output exhibits a net increase. Conversely, if the delayed data is slightly less than $\pi/2$ radians ahead of the clock, the phase detector's output has a negative average value, and the loop integrator's output exhibits a net decrease.

By plotting the phase detector's average output as a function of phase error, one finds that the multiplier-based phase detector has a "triangular" transfer characteristic:

![Diagram showing the transfer characteristic of the phase detector.](attachment:phase-detector-transfer-characteristic.png)

*Figure Assumes Maximum Data Transition Density

Figure 4.4: Transfer Characteristic of the Multiplier-Based Phase Detector

Note that, consistent with Figure 4.2, the phase detector's average output equals zero when the phase error between the delayed data and the clock is $\pi/2$ radians.

### 4.1.2 The Offset of the Multiplier-Based Phase Detector

Since its average output equals zero when the data is $\pi/2$ radians ahead of the clock (rather than when the data and the clock are optimally aligned), the multiplier-based phase detector has an offset of $\pi/2$ radians. In most applications, a static phase error of $\pi/2$ radians is intolerable, and a compensating phase shifter must be employed to align the delayed data and the clock (see Figure 4.5).

In practice, it is difficult to compensate for the offset of the multiplier-based phase detector. In addition to the problems associated with implementing monolithically an accurate, drift-free phase shifter, one must cope with the uncertainty in the phase

53
detector's offset. The phase detector's offset may deviate from $\pi/2$ radians if the nonlinear processor's operation differs from that depicted in Figure 2.5. For instance, if the delay line in the nonlinear processor is only $T/4$ seconds long, the phase detector's offset equals $\pi/4$, not $\pi/2$ radians. Furthermore, propagation delays through the nonlinear processor directly add to the phase detector's offset. For these reasons, the compensation for the offset of the multiplier-based phase detector is only approximate, and significant static phase errors often remain when such a phase detector is used [10].

4.1.3 Phase Detector Gain

Figure 4.4, which assumes a maximum data transition density (i.e., a 01010101 pattern), suggests that the gain of the multiplier-based phase detector equals $\frac{1}{\pi}$ V/radian. However, the phase detector's gain is less than $\frac{1}{\pi}$ V/radian if the data transition density is lower than 1 transition per clock period. When the data has few transitions, the phase detector's output is zero most of the time, and the average output of the phase detector is diminished (see Figure 4.3). In fact, the gain of the multiplier-based phase detector is directly proportional to the data transition density. Therefore, if the data transition density is 1 transition every 4 clock periods, the phase detector's gain is $\frac{1}{4\pi}$ V/radian.

Because the gain $K_D$ of the phase detector varies with data transition density, the acquisition speed of the DLL and the D/PLL depends on the data pattern (examine Equations 3.2 and 3.20). Fortunately, in the application at hand, a high data tran-
sition density is present whenever acquisition is supposed to occur, as the preamble code consists of 40 alternating ones and zeros. After acquisition has occurred within 40 clock periods, meaningful data is transmitted, and the average data transition density drops. The resulting drop in loop bandwidth \( K_D K_f \) is not a problem since acquisition has already been completed. Actually, the drop in loop bandwidth is beneficial, for it improves the DLL's and the D/PLL's noise immunity. Hence, in this application, the variation of the phase detector's gain with data transition density is an advantage, not a disadvantage.

### 4.1.4 Metastability

As indicated in Figure 4.4, the average output of the multiplier-based phase detector equals zero not only when the phase error is \( \pi/2 \) radians, but also when the phase error is \(-\pi/2\) radians. Note, though, that the phase detector's transfer characteristic has a negative slope at \(-\pi/2\) radians. This negative slope provides positive feedback around the loop\(^3\); as a result, the null at \(-\pi/2\) radians is a point of unstable equilibrium – that is, a metastable point.

While the loop cannot remain at the unstable null forever (due to noise), it can dwell in the vicinity of the null for a long time – much longer than one would expect from the loop bandwidth (Equation 3.2)\(^4\). If the initial phase error places the loop near its unstable null, acquisition will not occur within 40 clock periods, and part of the data burst will be lost. For this reason, unless complex "anti-hangup" methods are employed, a multiplier-based phase detector is not suitable when fast, reliable acquisition is needed [23].

### 4.1.5 The Ripple of the Multiplier-Based Phase Detector

This section examines how ripple on the output of the multiplier-based phase detector adds jitter to the delayed data. As mentioned earlier, the phase detector's output is integrated and filtered by higher-order poles before it is applied to the phase shifter (see Figures 4.6 and 4.7). Note how the higher-order poles filter the high frequency components of the loop integrator's output. By attenuating high frequency fluctuations in the phase shifter's control voltage, the higher-order poles reduce the jitter added to the delayed data.

Nonetheless, the higher-order poles cannot filter out fluctuations in the loop integrator's average output. The loop integrator's average output can shift if the data transition density changes, as depicted in Figure 4.8. When the data transition density falls from 1 transition every clock period to 1 transition every 4 clock periods, the

---

\(^3\)In this section the loop can be either a DLL or a D/PLL.

\(^4\)Equation 3.2 is only valid if the phase detector is linear over a 2\(\pi\) radian range.
Figure 4.6: DLL with Higher-Order Poles

Figure 4.7: Timing Diagram Illustrating Effect of Higher-Order Poles
average output of the loop integrator drops, and the control voltage decays to a lower value (with the time constants of the higher order poles). Eventually, the negative feedback of the DLL or the D/PLL restores the control voltage to its original value, but during the transient, jitter is added to the delayed data. Because this jitter is caused by changes in the data transition density, it is a form of \textit{data-dependent} jitter.

Clearly, the largest drop in the loop integrator's average output occurs when the data transition density changes from the maximum transition density possible (i.e., 1 transition every clock period) to the minimum transition density permitted by the data code (in this case, 1 transition every 4 clock periods). Examining Figure 4.8, one finds that the drop in the loop integrator's average output due to such a change in data transition density equals $3T/64$, where $T$ is the clock period (25 ns). The jitter added to the delayed data is obtained by multiplying $3T/64$ by $K_\phi$, the gain of the phase shifter.

$K_\phi$ can be determined from the loop bandwidth $K_D K_\phi$, which is set by acquisition time requirements. Since acquisition must occur within 1\,\mu s (40 \times 25 \text{ ns}), $K_D K_\phi$ must equal at least 4\,Mrad/s. (A loop bandwidth of 4\,Mrad/s allows the phase error to decay to 1.83\% of its initial value within 1\,\mu s.) Noting from Figure 4.4 that $K_D$ equals $\frac{1}{\pi}$ during acquisition, one finds:

$$K_\phi = 4\pi \times 10^8$$ (4.1)
Consequently, the jitter (zero-to-peak) added to the delayed data equals:

\[
\frac{3T}{64} \times 4\pi \times 10^8 = 0.0147 \text{ radians} = 0.84 \text{ degrees}
\] (4.2)

In almost all applications, such worst case jitter performance is acceptable. (Unlike random jitter, data-dependent jitter can often be bounded deterministically by worst case limits.)

### 4.2 The Self-Correcting Phase Detector

#### 4.2.1 Basic Operating Principle

Compared with the multiplier-based phase detector, the self-correcting phase detector has lower offset and better performance near its metastable point. Figure 4.9 shows the self-correcting phase detector in its simplest form.

![Diagram of the Self-Correcting Phase Detector](image)

**Figure 4.9: The Self-Correcting Phase Detector**

The circuit directly compares the phases of the delayed data and the clock in the following manner. After the delayed data changes state, the D input and Q output of D-type flip-flop U3 are no longer equal. XOR gate U1 senses this inequality and raises its output high. U1’s output remains high until the next rising edge of the
clock, at which time the delayed data's new state is clocked through U3, thereby eliminating the inequality between the D input and Q output of U3. At the same time, XOR gate U2 raises its output high because now the D input and Q output of D-type flip-flop U4 are unequal. U2's output remains high until the next falling edge of the clock, at which time the delayed data's new state is clocked through U4. Assuming that the clock has a 50% duty cycle, U2's output is a positive pulse half a clock period wide for each data transition. U1's output is also a positive pulse for each data transition, but its width depends on the phase error between the delayed data and the clock; its width equals half a clock period when the delayed data and the clock are optimally aligned. Hence, the phase error can be obtained by comparing the widths of the pulses out of U1 and U2.

Figure 4.10 is the timing diagram for the self-correcting phase detector with the delayed data and the clock optimally aligned. (Note that the data transitions are aligned with the falling edge of the clock.) In this case, the output of the phase detector has zero average value, and there is no net change in the loop integrator's output.

If the delayed data is ahead of the clock, the output of the phase detector has a positive average value, as shown in Figure 4.11. As a result, the loop integrator's output exhibits a net increase. Conversely, if the delayed data is behind the clock, the phase detector's output has a negative average value, and the loop integrator's output exhibits a net decrease.

Plotting the phase detector's average output as a function of phase error yields the "sawtooth" transfer characteristic presented in Figure 4.12. Consistent with Figure 4.10, the phase detector's average output equals zero when the phase error between the delayed data and the clock is zero.

One noteworthy feature of this phase detector is that the decision making circuit is an integral component of the phase detector, for the output of flip-flop U3 is the retimed data. Because alignment of the delayed data and the clock at the decision flip-flop is an intrinsic part of the phase detection process, static phase error is minimized [10]. In this sense, the phase detector is "self-correcting".

4.2.2 The Offset of the Self-Correcting Phase Detector

The transfer characteristic shown in Figure 4.12 suggests that the self-correcting phase detector has zero offset. In practice, the propagation delays through the digital logic of the phase detector must be considered. The propagation delays of XOR gates U1 and U2 can be safely neglected because such delays do not alter the widths of the pulses out of U1 and U2. On the other hand, the clock-to-Q delay of flip-flop U3 directly adds to the offset of the phase detector, for the delay through U3 widens the pulse on U1's output (see Figure 4.13). Since the clock-to-Q delay of an ECL
Figure 4.10: Timing Diagram for the Self-Correcting Phase Detector

Figure 4.11: Timing Diagram: Delayed Data Advanced
Figure 4.12: Transfer Characteristic of the Self-Correcting Phase Detector

Figure 4.13: Effect of Flip-Flop Delays on Pulse Widths
flip-flop is on the order of 1 ns [24,25], 1 ns static phase errors can result if the delay through U3 is left uncompensated.

In order to achieve static phase errors smaller than 1 ns, a compensating delay is added between the delayed data and XOR gate U1, as shown in Figure 4.14. The offset of the phase detector is reduced to the extent that the compensating delay matches the clock-to-Q delay of U3. In the next chapter it is demonstrated that the delay of a properly designed XOR gate can match the clock-to-Q delay of a flip-flop within 100 ps (0.1 ns). Hence, static phase errors significantly smaller than 1 ns can be achieved when the self-correcting phase detector is compensated in this manner.

4.2.3 Phase Detector Gain

Figure 4.12, which assumes a maximum data transition density, indicates a gain of $\frac{1}{2\pi}$ V/radian for the self-correcting phase detector. However, the gain of the self-correcting phase detector, like that of the multiplier-based phase detector, is directly proportional to the data transition density. Therefore, if the data transition density is 1 transition every 4 clock periods, the gain of the phase detector is $\frac{1}{2\pi}$ V/radian.

In both the multiplier-based and self-correcting phase detectors, the output consists of positive and negative pulses, the relative widths of which determine the average output of the phase detector (see Figures 4.3 and 4.11). While both the positive
and negative pulse widths are affected by the phase error in the multiplier-based phase detector, only the positive pulse width is affected by the phase error in the self-correcting phase detector. As a result, the gain of the self-correcting phase detector is one half that of the multiplier-based phase detector.

4.2.4 Metastability

As labelled in Figure 4.12, the transfer characteristic of the self-correcting phase detector has an unstable null, or metastable point, at $\pm \pi$ radians. When the phase error equals $\pm \pi$ radians, the data transitions are aligned with the rising edge of the clock. Because flip-flop U3 is triggered by the rising edge of the clock, U3 samples the delayed data as the delayed data changes state. Whether or not the delayed data’s new state is clocked through U3 is indeterminate, for the setup and hold time requirements of U3 have not been met. If the new state of the delayed data is clocked through U3, a $-\pi$ radian phase error is registered, and the average output of the phase detector equals its maximum negative value. If the new state of the delayed data is not clocked through U3, a $\pi$ radian phase error is registered, and the average output of the phase detector equals its maximum positive value. In either case, the phase detector produces a large error voltage, which drives the loop away from its metastable point. Thus, the self-correcting phase detector has better performance near metastability than does the multiplier-based phase detector.

Unfortunately, this performance near metastability can be degraded if the delayed data has significant jitter. Imagine that the initial phase error is slightly less than $\pi$ radians, so that the average output of the phase detector is highly positive. This positive error voltage directs the loop to increase the delay through the phase shifter, but, before the delay can be changed appreciably, jitter on the delayed data pushes the instantaneous phase error slightly above $\pi$ radians, thereby making the average output of the phase detector highly negative. The phase detector now directs the loop to decrease the delay through the phase shifter. By “equivocating”, the loop can dwell in the vicinity of the unstable null for a long time [23]. Higher-order poles worsen the effect, for they reduce the speed with which the delay through the phase shifter can be changed. While this effect is difficult to reproduce in the noise-free environment of a circuit simulator, it is readily observed in the laboratory. For this reason, further discussion of metastability is deferred until Chapter 6, in which the experimental results for the DLL and D/PLL prototypes are presented.

---

5It is safe to ignore the metastability of the flip-flop itself, for the probability of an ECL flip-flop remaining in an invalid logic state for a significant fraction of the acquisition time is negligibly small.
4.2.5 The Ripple of the Self-Correcting Phase Detector

As in the multiplier-based phase detector, the average output of the loop integrator shifts whenever the data transition density changes. Since each triangular pulse on the output of the loop integrator has positive net area (see Figure 4.10), the presence or absence of such a pulse affects the average output of the loop integrator.

The data-dependent jitter which results from such fluctuations in the loop integrator's average output can be determined by following an analysis essentially identical to that presented earlier for the multiplier-based phase detector. In this case, however, the net area of each triangular pulse is 4 times larger than in the case of the multiplier-based phase detector. (The pulses are now twice as wide and twice as high.) For this reason, the fluctuations in the average output of the loop integrator are now 4 times larger. In addition, the gain of the self-correcting phase detector (\(\frac{1}{2\pi} V/\text{radian}\)) is one half that of the multiplier-based phase detector (\(\frac{1}{\pi} V/\text{radian}\)). In order to maintain the same loop dynamics (Equations 3.2 and 3.20), the gain of the phase shifter must now be twice as high. Consequently, the data-dependent jitter added to the delayed data is now 8 times higher. For the 1 \(\mu\)s acquisition time specified for the DLL and D/PLL prototypes, the data-dependent jitter (zero-to-peak) added to the delayed data equals:

\[
8 \times \frac{3T}{64} \times 4\pi \times 10^6 = 0.1178 \text{ radians} = 6.75 \text{ degrees}
\]  

(4.3)

In critical applications, such jitter is unacceptable.

4.3 The Triwave Phase Detector

4.3.1 Basic Operating Principle

The triwave phase detector is an improvement upon the self-correcting phase detector [26]. While retaining all of the advantages of the self-correcting phase detector\(^0\), the triwave phase detector greatly reduces the jitter added to the delayed data. Figure 4.15 shows the triwave phase detector in its simplest form.

The operation of the triwave phase detector is similar to that of the self-correcting phase detector. After the delayed data changes state, the D input and Q output of D-type flip-flop U4 are no longer equal. XOR gate U1 senses this inequality and raises its output high. The output of U1 remains high until the next rising edge of the clock, at which time the delayed data's new state is clocked through U4, thereby

---

\(^0\)If the duty cycle of the clock is not exactly 50%, the triwave phase detector has higher offset than the self-correcting phase detector. But a simple modification of the triwave phase detector, described in Appendix A, solves this problem.
eliminating the inequality between the D input and Q output of U4. At the same time, XOR gate U2 raises its output high because now the D input and Q output of D-type flip-flop U5 are unequal. The output of U2 remains high until the next falling edge of the clock, at which time the delayed data's new state is clocked through U5. At the same time, XOR gate U3 raises its output high because now the D input and Q output of D-type flip-flop U6 are unequal. The output of U3 remains high until the next rising edge of the clock, at which time the delayed data's new state is clocked through U6. Assuming that the clock has a 50% duty cycle, the output of U2 is a positive pulse half a clock period wide for each data transition. The same is true regarding the output of U3. The output of U1 is also a positive pulse for each data transition, but its width depends on the phase error between the delayed data and the clock; its width equals half a clock period when the delayed data and the clock are optimally aligned. Hence, the phase error can be obtained by comparing the variable width of the pulse out of U1 with the fixed widths of the pulses out of U2 and U3. Note that the pulses out of U1 and U3 are weighted by 1, while the pulse out of U2 is weighted by −2.
Figure 4.16 is the timing diagram for the triwave phase detector with the delayed data and the clock optimally aligned. In this case, the output of the phase detector has zero average value, and there is no net change in the loop integrator’s output.

If the delayed data is ahead of the clock, the output of the phase detector has a positive average value, as shown in Figure 4.17. As a result, the loop integrator’s output exhibits a net increase. Conversely, if the delayed data is behind the clock, the output of the phase detector has a negative average value, and the loop integrator’s output exhibits a net decrease.

As shown in Figure 4.16, each data transition initiates a three-sectioned transient on the output of the loop integrator; this three-sectioned transient is called the triwave (pronounced “tri-wave”). The most important feature of the triwave is that it has zero net area; therefore, its presence or absence does not change the average output of the loop integrator. Note that when the data transition density equals 1 transition every clock period, each triwave begins before the previous triwave has ended. Because of the pipelined nature of the phase detector’s logic, overlapping triwaves are added without error.

The triwave phase detector possesses all of the advantages of the self-correcting phase detector. The decision making circuit is an integral component of the phase detector, for the output of flip-flop U4 is the retimed data. The triwave phase detector has the “sawtooth” transfer characteristic plotted in Figure 4.12. Indeed, it can be shown that the triwave phase detector and the self-correcting phase detector are completely equivalent in terms of offset, gain, and performance near metastability.

In addition, the triwave phase detector greatly reduces the jitter added to the delayed data, as the next section explains.

### 4.3.2 The Ripple of the Triwave Phase Detector

For the triwave phase detector, the average output of the loop integrator does not shift when the data transition density changes. Because the triwave has zero net area, its presence or absence does not change the average output of the loop integrator. The ripple on the output of the loop integrator has only high frequency components, which can be filtered by higher-order poles.

The degree to which the ripple on the output of the loop integrator is filtered is determined by the placement of the higher-order poles. However, considerations of loop stability place constraints on the placement of the higher-order poles. While several low frequency poles would greatly attenuate high frequency fluctuations in the control voltage, they would also degrade the stability of the DLL or the D/PLL. In order to preserve acceptable loop stability, the higher-order poles must be placed at frequencies significantly higher than the loop bandwidth of the DLL or the D/PLL. The actual implementation of the higher-order poles in an integrated circuit places
Figure 4.16: Timing Diagram for the Triwave Phase Detector

Figure 4.17: Timing Diagram: Delayed Data Advanced
an additional constraint on their placement. Because complex poles are relatively difficult to implement monolithically, it is desirable that all the higher-order poles be real. The following discussion examines how the ripple on the output of the loop integrator can be best filtered under the given constraints.

Consider the case of the DLL. (The following arguments also apply to the D/PLL since the high frequency loop dynamics of the DLL and the D/PLL are essentially identical.) Assuming that the higher-order poles are real, the negative of the loop transmission for the DLL equals:

\[ -L(s) = \frac{K_D K_\phi}{s} \frac{1}{(\tau_1 s + 1)} \frac{1}{(\tau_2 s + 1)} \cdots \frac{1}{(\tau_n s + 1)} \]  

\[(4.4)\]

where \(\tau_1, \tau_2, \ldots, \tau_n\) are the time constants of the \(n\) higher order poles.

Assume that the frequencies of the poles are much higher than the loop bandwidth \(K_D K_\phi\):

\[ \frac{1}{\tau_i} \gg K_D K_\phi \quad \forall \quad 1 \leq i \leq n \]  

\[(4.5)\]

Under this assumption, \(\arctan(\tau_i K_D K_\phi) \approx \tau_i K_D K_\phi\), and the degradation in phase margin caused by the higher order poles can be approximated as:

\[-\Delta \phi_m \approx K_D K_\phi (\sum_{i=1}^{n} \tau_i) \]  

\[(4.6)\]

For a given loop bandwidth and a given phase margin, the sum of the time constants of the higher-order poles is fixed.

Assume further that the ripple on the output of the loop integrator only has spectral energy far above the frequencies of the higher-order poles. In this case, the ripple is attenuated by a factor proportional to the product of the time constants of the higher-order poles since

\[ \frac{1}{(\tau_1 s + 1)} \frac{1}{(\tau_2 s + 1)} \cdots \frac{1}{(\tau_n s + 1)} \approx \frac{1}{(\prod_{i=1}^{n} \tau_i) s^n} \quad \text{for } s \text{ large} \]  

\[(4.7)\]

Hence, the product of the time constants must be maximized, under the constraint that the sum of the time constants is fixed (Equation 4.6). Through the use of Lagrange multipliers [27], the optimum solution is found to be:

\[ \tau_1 = \tau_2 = \ldots = \tau_n = \frac{-\Delta \phi_m}{n K_D K_\phi} \]  

\[(4.8)\]

This result, which states that the time constants of the higher-order poles should be equal, has been verified by computer simulation – even for cases in which the simplifying assumptions made above are not valid.
If the time constants of the higher-order poles are equal, the negative of the loop transmission of the DLL can be rewritten as:

\[-L(s) = \frac{K_D K_\phi}{s} \frac{1}{(\tau_n s + 1)^n}\]  \hspace{1cm} (4.9)

Note that the subscript of \(\tau_n\) indicates the number of higher-order poles.

While Equation 4.8 can be used to estimate the optimum value of \(\tau_n\), a more accurate and informative technique for setting \(\tau_n\) is available. A root contour diagram is constructed by plotting the closed loop poles of the DLL as \(\tau_n\) is varied from zero to infinity [28]. Once such a diagram is constructed, it is a simple matter to find the largest value of \(\tau_n\) for which the loop is acceptably stable.

The root contour technique is illustrated in Figure 4.18 for the case when \(n = 1\). For very low values of \(\tau_1\), the DLL is highly overdamped, with one dominant pole at \(K_D K_\phi\). As \(\tau_1\) is increased, the loop becomes less and less damped. For \(\tau_1 > 1/(4K_D K_\phi)\), the loop is underdamped, and the closed loop poles are complex. For \(\tau_1 = 1/(4K_D K_\phi)\), the loop is critically damped, and the two poles are real and equal. The critically damped loop, with two poles at \(2K_D K_\phi\), is significantly faster than the highly overdamped loop; thus, adding a higher-order pole at \(4K_D K_\phi\) improves the acquisition time of the DLL. (Because of component tolerances, however, it is not prudent to rely on this effect to meet the acquisition time requirement.) Making the loop underdamped by setting \(\tau_1 > 1/(4K_D K_\phi)\) is undesirable for two reasons. First of all, acquisition is slower for the underdamped loop than for the critically damped loop since the real parts of the poles, which determine settling time, are less negative for the underdamped loop. Furthermore, one must consider the sampled-data nature of the system. Because the phase error between the data and the clock
can only be detected at data transitions, the phase detector does not examine the phase error continuously, but at discrete instants of time. This sampling nature of the phase detector tends to degrade loop stability [29]. In view of this possible stability degradation, conservative design dictates that the loop be critically damped. For the 1\mu s acquisition time specified for the DLL and D/PLL prototypes, the time constant of the higher-order pole should equal:

$$\tau_1 = \frac{1}{4K_D K_\phi} = \frac{1}{4(4 \times 10^9)} = 63 \text{ ns}$$  \hspace{1cm} (4.10)

Similarly, by constructing root contours for the cases \( n = 2, n = 3, \text{ and } n = 4 \), the values of \( \tau_2, \tau_3, \text{ and } \tau_4 \) consistent with critical damping can be determined. For a 1\mu s acquisition time, the values of \( \tau_2, \tau_3, \text{ and } \tau_4 \) consistent with critical damping are:

$$\tau_2 = 35 \text{ ns}$$ \hspace{1cm} (4.11)

$$\tau_3 = 25 \text{ ns}$$ \hspace{1cm} (4.12)

$$\tau_4 = 20 \text{ ns}$$ \hspace{1cm} (4.13)

One can now find the extent to which the ripple on the output of the loop integrator can be filtered when one to four higher-order poles are used. (There is no need to consider more than four higher-order poles, as will be soon shown.) As with the two previous phase detectors, the largest transient on the control voltage occurs when the data transition density changes from the maximum transition density possible to the minimum transition density permitted by the data code (or vice versa). Figure 4.19 shows the output of the loop integrator when the data transition density alternates between a very low transition density and the maximum transition density possible. In the figure the amplitude of the triwave has been normalized to 1 V; the bit rate of the data equals 40 Mbit/s.

Figures 4.20–4.23 demonstrate how the ripple of Figure 4.19 is attenuated by one pole with a time constant of 63 ns, two poles with a time constant of 35 ns, three poles with a time constant of 25 ns, and four poles with a time constant of 20 ns. For one pole with a time constant of 63 ns, the peak ripple is attenuated by a factor of \( \frac{1V}{151mV} = 7.6 \). For two poles with a time constant of 35 ns, the peak ripple is attenuated by a factor of 13.4, which is significantly larger than 7.6. On the other hand, using more than two higher-order poles offers little improvement. For three poles with a time constant of 25 ns, and for four poles with a time constant of 20 ns, the peak ripple is attenuated by a factor of 14.1. Thus, two poles with a time constant of 35 ns, while being simple to implement, provide near optimal filtering.

Assuming that two higher-order poles with a time constant of 35 ns are used, the jitter added to the delayed data can be obtained as follows. Examining Figure 4.16,
one finds that the unnormalized amplitude of the triwave equals $T/2$, where $T$ is the clock period (25 ns). Since the two higher-order poles attenuate the peak ripple by a factor of 13.4, the jitter added to the delayed data equals:

$$\frac{T}{2} \times \frac{1}{13.4} \times K_\phi$$  \hspace{1cm} (4.14)

$K_\phi$, the gain of the phase shifter, can be determined from the loop bandwidth $K_D K_\phi$. For a 1μs acquisition time, $K_D K_\phi$ must equal at least 4Mrad/s. Noting that $K_D$ equals $\frac{1}{2\pi}$ during acquisition, one finds:

$$K_\phi = 8\pi \times 10^6$$  \hspace{1cm} (4.15)

Consequently, the jitter (zero-to-peak) added to the delayed data equals:

$$\frac{T}{2} \times \frac{1}{13.4} \times 8\pi \times 10^6 = 0.0234 \text{ radians} = 1.34 \text{ degrees}$$  \hspace{1cm} (4.16)
Such jitter is five times lower than that of the self-correcting phase detector (6.75 degrees) and comparable to that of multiplier-based phase detector (0.84 degrees). In almost all applications, such worst case jitter performance is acceptable.

In conclusion, the triwave phase detector is preferred to the multiplier-based phase detector because it has lower offset, a larger linear range, and better performance near metastability. The triwave phase detector is preferred to the self-correcting phase detector because it greatly reduces the jitter added to the delayed data. The next chapter presents a detailed transistor-level design of the triwave phase detector.
Figure 4.20: Filtered Control Voltage: One Higher-Order Pole with $\tau_1 = 63 \text{ ns}$

Figure 4.21: Filtered Control Voltage: Two Higher-Order Poles with $\tau_2 = 35 \text{ ns}$
Figure 4.22: Filtered Control Voltage: Three Higher-Order Poles with $\tau_3 = 25 \text{ ns}$

Figure 4.23: Filtered Control Voltage: Four Higher-Order Poles with $\tau_4 = 20 \text{ ns}$
Chapter 5

Design and Simulation of a Triwave Phase Detector

A direct implementation of the triwave phase detector shown in Figure 4.15 is not practical for the following reason. Because digital signals have very fast rise and fall times (e.g., 1 ns for an ECL XOR gate), it is difficult to add and subtract the outputs of the XOR gates without error. For instance, if an operational amplifier is used to perform the addition and subtraction, errors can arise due to slew-induced distortion.

The implementation of the triwave phase detector shown in Figure 5.1 avoids this problem, for in this circuit currents are added and subtracted instead of voltages. The circuit uses three digitally-controlled current switches; the outputs of XOR gates U1, U2, and U3 control switches S1, S2, and S3, respectively. When the output of U1 is high, the current from source I1 is diverted into the capacitor C1, and the capacitor voltage ramps upward at a rate of I/C1. When the output of U2 is high, the current from source I2 is diverted into the capacitor C1. Because the source I2 has twice the magnitude and the opposite polarity of source I1, the capacitor voltage ramps downward at a rate of −2I/C1. When the output of U3 is high, the current from source I3 is diverted into the capacitor C1, and the capacitor voltage ramps upward at a rate of I/C1. The timing diagram for this circuit is given in Figure 5.2. As shown in the timing diagram, this circuit synthesizes the triwave waveform directly; a separate loop integrator is not needed. Because of the pipelined nature of the phase detector’s logic, overlapping triwaves are added without error.

This chapter presents a transistor-level design of the circuit shown in Figure 5.1. The chapter is divided into three major sections. The first describes the digitally-controlled charge pump, which consists of the current switches and the capacitor C1. The second describes the XOR gates and D-type flip-flops which constitute the digital control logic of the phase detector. The third presents the simulated performance of the entire phase detector.
Figure 5.1: Charge Pump Implementation of Triwave Phase Detector

Figure 5.2: Timing Diagram: Charge Pump Implementation
5.1 The Differential Charge Pump

5.1.1 Limitations of Process Technology

Limitations of process technology place important constraints on the design of any integrated circuit. In this application, high frequency transistors are required for satisfactory operation at high bit rates. At Analog Devices Semiconductor, the NPN transistors on the bipolar “FLASH” process are the fastest available. For these transistors, the cutoff frequency ($f_T$) is between 2 and 3 GHz, the current gain ($\beta$) exceeds 100, and the current gain–Early voltage product ($\beta V_A$) is 6000 V. But for the lateral PNP transistors, the cutoff frequency is less than 20 MHz, the current gain is only 25, and the current gain–Early voltage product is 700 V. Because of their low speed, these PNP transistors are unsuitable for use in high speed switching circuits. In particular, the current switches shown in Figure 5.1 must be realized with NPN transistors exclusively.

In addition to high frequency NPN transistors, the FLASH process also features precision thin film resistors and high quality MOS capacitors. Even without laser trimming, matching between thin film resistors is typically better than 0.1%. On the other hand, these resistors have much larger absolute tolerances – about 20%. The MOS capacitors also match well but have large absolute tolerances. The excellent matching of resistors and capacitors available on the FLASH process is exploited in the design of the differential charge pump, described next.

5.1.2 The Basic Differential Charge Pump

Because they require complementary devices, most charge pumps do not perform well at high bit rates [30]. In contrast, the differential charge pump employs only NPN transistors and provides excellent performance at high bit rates. Figure 5.3 shows the differential charge pump in its simplest form\(^1\). Three differential pair switches, Q1-Q2, Q3-Q4, and Q5-Q6, divert current into and out of the capacitor, the voltage across which is sensed differentially. Because they handle twice the current, Q3 and Q4 have twice the areas of the other transistors. These three differential pairs are connected to the three XOR gates of Figure 5.1 as follows. The (differential) output of XOR gate U1 controls differential pair Q1-Q2. When the output of U1 is low, Q1 is on, and Q2 is off; when the output of U1 is high, Q2 is on, and Q1 is off. Similarly, the outputs of XOR gates U2 and U3 control differential pairs Q3-Q4 and Q5-Q6, respectively. When the output of U2 is low, Q3 is on, and Q4 is off. When the output of U3 is low, Q5 is on, and Q6 is off.

\(^1\)For the purposes of this discussion, one may assume that the common-mode voltage on the collectors of the differential pairs is established by a common-mode feedback loop. Such feedback could be applied by adjusting the total current through sources I1 and I2.
When there are no data transitions, the outputs of XOR gates U1, U2, and U3 are all low; consequently, Q1, Q3, and Q5 are on. Because the bias currents for the three differential pairs are precisely weighted 1:2:1, the charge pump is balanced; that is, there is no tendency to charge the capacitor C1 one way or the other.

When a data transition occurs, the charge pump is activated. Immediately after the data changes state, XOR gate U1 raises its output high, thereby switching differential pair Q1-Q2. The charge pump is no longer balanced, and a current equal to I flows through the capacitor C1 (to the right in Figure 5.3). In this manner the first section of the triwave is produced. Upon the next rising edge of the clock, differential pair Q1-Q2 switches back to its original state (since the output of U1 is now low). Simultaneously, XOR gate U2 raises its output high, thereby switching differential pair Q3-Q4. Because the bias current for differential pair Q3-Q4 equals 2I, a current equal to 2I now flows through the capacitor in the opposite direction (to the left in Figure 5.3). In this manner the second section of the triwave is produced. Upon the next falling edge of the clock, differential pair Q3-Q4 switches back to its original state. Simultaneously, XOR gate U3 raises its output high, thereby switching differential pair Q5-Q6. A current equal to I now flows through the capacitor (to the right in Figure 5.3), and the third section of the triwave is produced. Upon the next rising edge of the clock, differential pair Q5-Q6 switches back to its original state,
and the charge pump is again balanced.

Under the assumption that the data has maximum transition density\(^2\), it is readily shown that the gain of this phase detector equals:

\[
\frac{K_D}{s} = \frac{I}{2\pi C1 s}
\]

(5.1)

Because this phase detector includes an intrinsic integration, the gain of the phase detector has a pole at the origin. Note that \(K_D\) is directly proportional to \(I\) and inversely proportional to \(C1\). In order to minimize the variation in phase detector gain with temperature, the charging current \(I\) should have a low temperature coefficient. By changing \(I\) or \(C1\), the loop bandwidth \(K_DK_\phi\) is easily modified. The ease with which \(K_D\) can be adjusted is particularly important because \(K_\phi\) is constrained by available signal levels, as the following discussion demonstrates.

As explained in Chapter 3, rapid acquisition in a DLL or a D/PLL requires that the output of the loop integrator (in this case, the voltage on capacitor \(C1\)) be initialized to the middle of the phase shifter's range before each acquisition, and that the total range of the phase shifter be sufficiently large. For most applications, a total range of \(3\pi\) radians is adequate. Assuming that the total range of the phase shifter is \(3\pi\) radians, and 0 V on the capacitor corresponds to the middle of the phase shifter's range, the capacitor voltage needed to drive the phase shifter to the end of its range equals:

\[
\frac{3\pi/2}{K_\phi}
\]

(5.2)

For reasons to be discussed later, it is necessary to keep the capacitor voltage between 150 mV and -150 mV at all times. This in turn requires that:

\[
\frac{3\pi/2}{K_\phi} \leq 150 \text{ mV}
\]

(5.3)

Rearranging terms, one obtains:

\[
K_\phi \geq \frac{3\pi/2}{150 \text{ mV}} = 10\pi \text{ radian/V}
\]

(5.4)

Equation 5.4 places a lower limit on the gain of the phase shifter. In general, \(K_\phi\) should be no larger than necessary, for higher values of \(K_\phi\) makes the phase shifter more sensitive to noise on its control voltage. Hence, the gain of the phase shifter is effectively fixed by Equation 5.4.

\(^2\)As discussed in the last chapter, the gain of the triwave phase detector is proportional to the data transition density.
In the application at hand, $K_D K_\phi$ must exceed 4 Mrad/s during acquisition. If $I = 25 \, \mu A$ and $C_1 = 30 \, \text{pF}$ (both of which are reasonable values on the FLASH process), the loop bandwidth $K_D K_\phi$ during acquisition equals:

$$K_D K_\phi = \frac{I}{2\pi C_1} \times 10\pi = \frac{10\pi (25 \times 10^{-6})}{2\pi (30 \times 10^{-12})} = 4.17 \text{Mrad/s} \quad (5.5)$$

which is quite close to 4 Mrad/s.

5.1.3 Negative Resistance Circuit

If sufficient supply voltage were available, current sources I1 and I2 could be implemented with lateral PNP transistors, the total current through which could be controlled by a common-mode feedback loop. Unfortunately, because the DLL and the D.PLL must operate on a single supply as low as 4.5 V, the upper current sources must have a compliance of 1 V. As a result, the PNP transistor current sources cannot be degenerated heavily enough to ensure either high output resistance or good matching between I1 and I2.

An alternative approach to implementing I1 and I2, in which resistors are used instead of current sources, is illustrated in Figure 5.4. The only problem with such a charge pump is that it allows the capacitor to discharge through the bias resistors. That is, the integrator is leaky, and the gain of the phase detector equals:

$$\frac{K_D}{(s + \frac{1}{\tau_d})} \quad (5.6)$$

where $\tau_d$ is the time constant of the discharge. Note that the d.c. gain of the phase detector is now finite and equals:

$$K_D \tau_d \quad (5.7)$$

The time constant of the discharge $\tau_d$ is simply the product of $C_1$ and the Thévenin resistance facing $C_1$:

$$\tau_d = C_1 R_{th} = C_1 \frac{2 R_1 R_2}{(2 R_1 + R_2)} \quad (5.8)$$

Because $\tau_d$ is directly proportional to $C_1$, and $K_D$ is inversely proportional to $C_1$ (Equation 5.1), the d.c. gain of the phase detector $K_D \tau_d$ is independent of $C_1$, as one would expect.

The negative resistance technique is based on the following observation. If $R_2$ equals $-2R_1$ exactly, $\tau_d$ is infinite, and therefore so is the d.c. gain of the phase detector (Equation 5.7). While a discharging current will continue to flow through the upper bias resistors, it will be cancelled by current injected by the negative resistance,
which can be synthesized by active circuitry. If the current cancellation were perfect, a perfect integrator would be realized, and the capacitor voltage could be held forever.

Perfect current cancellation is of course impossible, but also unnecessary. The d.c. gain of the phase detector must only be large enough to keep the static phase error negligibly small. Consider the case of the DLL. (The analysis for the D/PLL is similar and leads to essentially identical results.) Assuming that the voltage on capacitor C1 is initialized to 0 V before each acquisition, the voltage on the capacitor after acquisition equals:

$$\frac{\phi_o}{K_\phi}$$

(5.9)

where $\phi_o$ equals the initial phase error. Since the gain of the phase detector equals $K_D \tau_d$, the static phase error due to finite d.c. gain equals:

$$\phi_e = \frac{\phi_o}{K_D K_\phi \tau_d}$$

(5.10)

As $-\pi \leq \phi_o \leq \pi$, the worst case static phase error due to finite d.c. gain equals:

$$\phi_e^* = \frac{\pi}{K_D K_\phi \tau_d}$$

(5.11)

Because $K_D$ is proportional to the data transition density, the static phase error due to finite phase detector gain changes whenever the data transition density changes. But
if the static phase error is changing with the data transition density, data-dependent jitter is added to the delayed data. In the case of a D/PLL, data-dependent jitter is also added to the retimed data. For this reason, it is important to keep this component of static phase error very small.

Assume now that the worst case static phase error must be less than 1.5 degrees when the data transition density equals 1 transition every 4 clock periods, or equivalently, that the worst case static phase error must be less than 0.375 degrees when the data has maximum transition density. (Such a requirement ensures that the peak-to-peak jitter added to the delayed data does not exceed 1.125 degrees.) Since $K_D$ equals $I/(2\pi C_1)$ when the data has maximum transition density, and $K_\phi$ is constrained to be $10\pi$ (Equation 5.4), the latter condition can be written as:

$$\phi^*_c = \frac{\pi}{(2\pi C_1)(10\pi)(C_1|R_{th}|)} \leq 0.0065 \text{ radians} = 0.375 \text{ degrees}$$ (5.12)

The absolute value sign around $R_{th}$ reflects the fact that $R_{th}$ may be positive or negative\(^3\), depending on the relative magnitudes of $2R_1$ and $R_2$. Rearranging and cancelling terms, one obtains:

$$I|R_{th}| \geq 97 \text{ V}$$ (5.13)

Considerations of saturation of the transistors limit the voltage drop across each of the upper bias resistors to 0.5 V:

$$(2I)R_1 = 0.5V$$ (5.14)

Dividing Equation 5.13 by Equation 5.14 yields:

$$\frac{|R_{th}|}{2 R_1} \geq 194$$ (5.15)

Thus the magnitude of $R_{th}$ must be almost 200 times larger than $2R_1$. From Equation 5.8 it follows that this condition is satisfied if the matching between the positive and negative resistances is better than 0.5%. It should be noted that no assumptions were made about $I$ or $C_1$ in deriving this result. Consequently, given the headroom constraints cited above, this result holds for any loop bandwidth (Equation 5.5).

In the present implementation, $I=25 \mu A$, $C_1=30 \text{ pF}$, and $R_1=10 \text{ k}\Omega$. (Note that $(2I)R_1$ does indeed equal 0.5 V.) In this case, Equation 5.15 is satisfied if $|R_{th}| \geq 4 \text{ M}\Omega$, or equivalently, if:

$$|\tau_d| \geq (30 \times 10^{-12}) \times (4 \times 10^6) = 120 \mu s$$ (5.16)

\(^3\)If $R_{th}$ is negative, $\tau_d$ is negative, and the phase detector is unstable open loop (see Equation 5.6). However, when the phase detector is placed inside a DLL or a D/PLL, it is stabilized by the overall negative feedback of the loop. For this reason, the polarity of $R_{th}$ is inconsequential; only its magnitude is important.
This last equation is particularly convenient, for $\tau_d$ is easily determined on a circuit simulator by observing how the capacitor voltage decays (or grows) over time.

The problem of synthesizing a negative resistance satisfying Equation 5.16 is considered next.

Figure 5.5 shows the negative resistance circuit in its simplest form.

![Negative Resistance Circuit Diagram]

**Figure 5.5: Simple Negative Resistance Circuit**

The differential-mode input resistance of this circuit can be calculated as follows. (The common-mode input resistance is ideally infinite.) Assume, for the moment, that $\beta$ is infinite, and $v_{IN}$ is sufficiently small so that the transistors can be modelled as linear about their operating points. It then follows that $v_E$ equals $v_{IN}$ multiplied by the gain of the emitter followers Q1 and Q2:

$$v_E = \frac{v_{IN}}{1 + \frac{1}{g_m R1}}$$  \hspace{1cm} (5.17)

where $g_m$ is the transconductance of the transistors. Since $\beta$ is infinite, the collector current and emitter current of each transistor are equal, and $i_{in^4}$ equals:

$$i_{in} = \frac{1}{2 R1} \times \frac{v_{IN}}{1 + \frac{1}{g_m R1}}$$  \hspace{1cm} (5.18)

\(^4\text{The lower case subscript indicates that } i_{in} \text{ is an incremental quantity}\)

83
The differential-mode input resistance is then found to be:

\[ R_{in} = \frac{v_{IN}}{i_{in}} = -2R1 \left( 1 + \frac{1}{g_m R1} \right) \] (5.9)

Because the gain of the emitter followers Q1 and Q2 is less than unity, the magnitude of \( i_{in} \) is smaller than the desired value, and the magnitude of the negative resistance is larger than 2R1 by the factor:

\[ 1 + \frac{1}{g_m R1} = 1 + \frac{2kT/q}{IBIAS R1} \] (5.20)

While increasing \( I_{BIAS} R1 \) improves the matching between the positive and negative resistances, \( I_{BIAS} R1 \) is limited to a maximum of 600 mV by considerations of transistor saturation, for replacing R2 in Figure 5.4 with this negative resistance circuit increases the voltage drop across the upper bias resistors by \( (I_{BIAS} R1)/2 \). If \( I_{BIAS} R1 = 600 \) mV, and \( T = 300 \) K, the magnitude of the negative resistance is 9% larger than 2R1. Such a large mismatch between the positive and negative resistances is intolerable.

The accuracy of the negative resistance circuit can be improved by the addition of two diode-connected transistors, as shown in Figure 5.6.

![Figure 5.6: Improved Negative Resistance Circuit](image-url)
The differential-mode input resistance of the improved circuit can be determined in the following manner. Again assume that $\beta$ is infinite. Under this assumption, transistors Q1 and Q3 have identical collector currents, as do transistors Q2 and Q4. Therefore, assuming that the transistors are identical, and the Early effect is negligible, one can write:

$$v_{BE1} = v_{BE3}$$  \hspace{1cm} (5.21)
$$v_{BE4} = v_{BE2}$$  \hspace{1cm} (5.22)

Adding these last two equations, one obtains:

$$v_{BE1} + v_{BE4} = v_{BE2} + v_{BE3}$$  \hspace{1cm} (5.23)

That is, the voltage drop between the base of Q1 and the emitter of Q4 equals the voltage drop between the base of Q2 and the emitter of Q3. Therefore $v_E$ equals $v_{IN}$ exactly. Since $\beta$ is infinite, the collector current and emitter current of each transistor are equal, and $i_{in}$ equals:

$$i_{in} = -\frac{v_{IN}}{2R1}$$  \hspace{1cm} (5.24)

The differential-mode input resistance is now:

$$R_{in} = \frac{v_{IN}}{i_{in}} = -2R1$$  \hspace{1cm} (5.25)

Hence, under the assumption of infinite $\beta$, the positive and negative resistances match as well as the thin film resistors themselves. Since the matching between thin film resistors on the FLASH process is typically better than 0.1%, the requirement that the positive and negative resistances match better than 0.5% is satisfied.

It should be noted that Equation 5.25 was derived without assuming that the transistors could be replaced by linear circuit models. Consequently, Equation 5.25 is valid for relatively large values of $v_{in}$.

On the other hand, Equation 5.25 is not valid when $v_{IN}$ is larger than ±200 mV. Even when $v_{IN} = 0$ V, the base-to-collector voltages of transistors Q3 and Q4 equal 0 V. For nonzero values of $v_{IN}$, the base-to-collector junction of either Q3 or Q4 is forward-biased. While the effects of transistor saturation can be neglected when $v_{IN}$ is less than ±200 mV, they degrade the performance of the negative resistance circuit when $v_{IN}$ is larger than ±200 mV — especially at high temperatures. In order to be conservative, it is desirable to keep the voltage on the capacitor of the charge pump between 150 mV and -150 mV at all times.

Finite $\beta$ degrades the matching between positive and negative resistances through two different effects. It is valid to consider each effect separately, for the two effects are small and do not interact significantly.
The first (and dominant) effect is that finite base currents directly subtract from the magnitude of $i_n$ determined above. Refer again to Figure 5.6. For the sake of simplicity, assume that $v_E$ equals $v_{IN}$ exactly. It follows then that the incremental emitter current of Q3 equals:

$$i_{e3} = \frac{v_{IN}}{2R1} \quad (5.26)$$

Neglecting the variation of $\beta$ with collector current, one can write:

$$i_{c3} = -i_{e3} \left(\frac{1}{1 + \frac{1}{\beta}}\right) \approx -i_{e3} \left(1 - \frac{1}{\beta}\right) = -\frac{v_{IN}}{2R1} \left(1 - \frac{1}{\beta}\right) \quad (5.27)$$

$$i_{b3} = -i_{e3} \left(\frac{1}{\beta + 1}\right) \approx -i_{e3} \left(\frac{1}{\beta}\right) = -\frac{v_{IN}}{2\beta R1} \quad (5.28)$$

$$i_{b4} \approx \frac{v_{IN}}{2\beta R1} \quad (5.29)$$

Applying KCL at the emitter of Q1 yields:

$$i_n = -i_{c1} = i_{c3} + i_{b4} \approx -\frac{v_{IN}}{2R1} \left(1 - \frac{2}{\beta}\right) \quad (5.30)$$

Hence, the magnitude of $i_n$ is smaller than the desired value (Equation 5.24) by two base currents. If $\beta = 100$, the magnitude of the negative resistance is 2% larger than the positive resistance (2R1).

In order to improve the matching between the positive and negative resistances, a base current compensation circuit is added to the negative resistance circuit, as shown in Figure 5.7. The topology of the compensation circuit is very similar to that of the negative resistance circuit itself, except that the collectors of Q5 and Q6 are connected to the positive power supply. Note that the bias current for the compensation circuit equals $2I_{BIAS}$. Accordingly, transistors Q5-8 have twice the areas of transistors Q1-4, and the resistors in the compensation circuit are half as large as those in the negative resistance circuit itself. Since only the bases of Q5 and Q6 are connected to the negative resistance circuit, the compensating current $i_a$ equals to a good approximation:

$$i_a \approx -\frac{v_{in}}{\beta R1} \quad (5.31)$$

The magnitude of $i_n$ is now correct, since

$$i_n \approx -\frac{v_{IN}}{2R1} \left(1 - \frac{2}{\beta}\right) = \frac{v_{IN}}{2R1} - \frac{v_{IN}}{\beta R1} \quad (5.32)$$
Figure 5.7: Negative Resistance Circuit with Base Current Compensation

The second effect caused by finite $\beta$ is that $v_E$ no longer equals $v_{IN}$ exactly. From Equation 5.30 it follows that:

$$i_{c1} = -i_{c3} \left( \frac{1}{1 + \frac{1}{\beta}} \right) \approx -\frac{v_{IN}}{2R1} \left( 1 - \frac{2}{\beta} \right) \left( 1 - \frac{1}{\beta} \right) \approx -\frac{v_{IN}}{2R1} \left( 1 - \frac{3}{\beta} \right) \quad (5.33)$$

Comparing Equations 5.33 and 5.27, one sees that if $\beta$ is finite, the collector currents of Q1 and Q3 are not equal but differ by:

$$i_{c1} - i_{c3} \approx \frac{v_{IN}}{\beta R1} \quad (5.34)$$

Assuming now that $v_{IN}$ is sufficiently small that the transistors can be modelled as linear about their operating points\(^5\), it follows that:

$$v_{BE1} \approx v_{BE3} + \frac{v_{IN}}{\beta g_m R1} \quad (5.35)$$

Similarly, one can show that

$$v_{BE4} \approx v_{BE2} + \frac{v_{IN}}{\beta g_m R1} \quad (5.36)$$

\(^5\text{Because the effects discussed here are small, the errors introduced by using a linearized circuit model are insignificant.}\)
Adding these last two equations, one obtains:

\[ v_{BE1} + v_{BE4} + v_{BE5} + \frac{2v_{IN}}{\beta g_m R1} \]  

(5.37)

That is, the voltage drop between the base of Q1 and the emitter of Q4 exceeds the voltage drop between the base of Q2 and the emitter of Q3 by a voltage equal to \( 2v_{IN}/(\beta g_m R1) \). Therefore \( v_E \) is smaller than \( v_{IN} \):

\[ v_E \approx v_{IN} \left( 1 - \frac{2}{\beta g_m R1} \right) \]  

(5.38)

Ignoring base current errors (since they were accounted for earlier), the magnitude of \( i_{in} \) is smaller than the desired value by an error current equal to:

\[ i_{error} \approx \frac{1}{2} \frac{v_{IN}}{R1} = \frac{v_{IN}}{\beta g_m R1^2} \]  

(5.39)

If \( R1 = 10 \, \text{k}\Omega, \beta = 100, I_{BIAS} = 60 \, \mu\text{A}, \) and \( T = 300 \, \text{K} \), the magnitude of \( i_{in} \) is 0.17% smaller than the desired value. Since the total mismatch between the positive and negative resistances due to all sources of error must be less than 0.5%, a 0.17% contribution to the error budget is considerable.

Because the error current (Equation 5.39) is inversely proportional to \( \beta \), it can be compensated by adding a resistor RCC to the base current compensation circuit\(^6\), as shown in Figure 5.8. From Equation 5.39 it follows that the value of RCC needed to compensate the error current equals:

\[ RCC = g_m R1^2 \]  

(5.40)

In order for such compensation to be effective over a wide temperature range, \( g_m \) must be independent of temperature, which requires that \( I_{BIAS} \) (i.e., both current sources of Figure 5.8) be proportional to absolute temperature (PTAT). In the current implementation, \( R1 = 10 \, \text{k}\Omega, \) and \( I_{BIAS}, \) which is PTAT, equals 60 \( \mu\text{A} \) at a temperature of 300K. In this case, the necessary value of RCC is 120 \( \text{k}\Omega. \)

The resistively biased charge pump with fully compensated negative resistance was simulated on ADICE, Analog Devices' proprietary version of the popular circuit simulator SPICE. Figure 5.9 shows the decay of the voltage on the charge pump capacitor when no data transitions are present, so that the charge pump is inactive.

---

\(^6\)As a matter of practical interest, the same compensation can be effected without adding RCC by decreasing the values of the resistors in series with the emitters of Q7 and Q8 by an appropriate amount. Representing the compensation as shown in Figure 5.8 merely emphasizes the fact that two different effects are being compensated.
Figure 5.8: Fully Compensated Negative Resistance Circuit

Figure 5.9: Decay of Charge Pump Capacitor Voltage
The finite time constant evident in the figure is a result of effects neglected in the preceding analysis. (The finite Early voltage of transistors Q3 and Q4 is responsible for one such effect.) Nonetheless, the matching between the positive and negative resistances is better than 0.5%, for the time constant (285 μs) satisfies Equation 5.16. In fact, a 285 μs time constant corresponds to 0.2% matching between the positive and negative resistances. The performance of the negative resistance circuit over temperature was also simulated and verified.

While resistor and β mismatches can degrade this performance, a worst case analysis based on 0.1% resistor matching and 10% β matching shows that the positive and negative resistances should match better than 0.5%. As explained earlier, such a mismatch between the positive and negative resistances results in a worst case static phase error of 1.5 degrees when the data transition density equals 1 transition every 4 clock periods. Such performance is quite satisfactory, for current source mismatches in the charge pump result in larger static phase errors, as the next section explains.

5.1.4 Effect of Current Source Mismatches in the Charge Pump

This section examines how current source mismatches (and mismatches which can be modelled as current source mismatches) in the charge pump result in phase detector offset and thereby increase static phase error. Refer again to Figure 5.4. If the bias current for the middle differential pair does not equal the sum of the bias currents for the left and right differential pairs, the capacitor voltage will droop when the charge pump is inactive, even if the positive and negative resistances match perfectly. For instance, if the bias current for the middle differential pair equals \(2I - \Delta I/2\), and the sum of the bias currents for the left and right differential pairs equals \(2I + \Delta I/2\), the capacitor voltage will decrease at a rate of \(\frac{\Delta I}{2C_1}\). In the steady state, a static phase error is necessary to cancel this droop; hence, the phase detector has an offset. Since the gain of the phase detector equals \(K_D/\delta\), the required static phase error equals:

\[
\phi_e = \frac{1}{K_D} \times \frac{\Delta I}{2C_1}
\]  

(5.41)

Because \(K_D\) is proportional to the data transition density, the static phase error due to current source mismatches changes whenever the data transition density changes, just like the static phase error due to finite phase detector gain (cf. Equation 5.10). As explained earlier, data-dependent jitter is added to the delayed data when the static phase error depends on the data transition density.

When the data transition density equals 1 transition every 4 clock periods, \(K_D\) equals \(I/(8\pi C_1)\), and the static phase error equals its maximum value:

\[
\phi_e = \frac{8\pi C_1 \Delta I}{2 I C_1} = \frac{8\pi \Delta I}{2I}
\]  

(5.42)
Assuming now that $\Delta I$ is 0.1% of $2I$, the last equation becomes:

$$\phi_e = 0.025 \text{ radians} = 1.44 \text{ degrees}$$  \hspace{1cm} (5.43)

Unfortunately, there are other sources of phase detector offset besides the one just discussed. For instance, assume that the upper bias resistors (those labelled R1 in Figure 5.4) do not match. (A mismatch between the upper biasing resistors does not imply that the gain of the phase detector is finite, for the magnitude of the negative resistance can still equal the series resistance of the two mismatched resistors.) In this case unequal currents flow through the upper biasing resistors when the voltage across the capacitor equals 0 V. This imbalance in currents causes the capacitor voltage to droop and can be modeled as an equivalent current source mismatch. Thus, a mismatch in the upper biasing resistors results in the same kind of phase detector offset discussed above. The same is true regarding resistor and base-to-emitter voltage ($V_{BE}$) mismatches in the negative resistance circuit (Figure 5.6).

Assuming that the base-to-emitter voltages of the transistors of the negative resistance circuit match so that the offset from $v_{IN}$ to $v_E$ in Figure 5.6 is less than 0.5 mV, and assuming 0.1% matching for the current sources and resistors, a worst case analysis shows that in the present implementation the offset of the phase detector should be less than 6 degrees when the data transition density equals 1 transition every 4 clock periods. As noted earlier, with this same data transition density, the expected mismatch between the positive and negative resistances results in a worst case static phase error of only 1.5 degrees. Excluding timing errors in the digital control logic, which are discussed later, the total static phase error of the DLL or the D/PLL is expected to be less than 7.5 degrees under all conditions.

5.1.5 Reset Circuitry

As explained in Chapter 3, rapid acquisition in a DLL or a D/PLL requires that the output of the loop integrator (in this case, the voltage on the charge pump capacitor) be initialized to the middle of the phase shifter's range before each acquisition. Figure 5.10 presents the reset circuitry used to discharge the capacitor to 0 V before each acquisition. In order not to clutter up the schematic, the negative resistance circuit is not shown in the figure.

When a data burst is present, and the signal_detect bit is high, the reset circuitry is disabled. Both binary signals VR and VR2 are low. Consequently, transistors Q9,

---

7Such matching between currents is achievable on the FLASH process if the resistors and transistors making up the current sources are laid out with care.

8Such base-to-emitter voltage matching is achievable on the FLASH process if the transistors making up the negative resistance circuit are laid out properly.
Q11, and Q14 are off, and Q10, Q12, and Q13 are on. The collector of Q13 pulls down on the bases of Q7 and Q8, thereby cutting Q7 and Q8 off.

When a data burst is not present, and the signal_detect bit is low, both VR and VR2 are high. Now transistors Q9, Q11, and Q14 are on, and Q10, Q12, and Q13 are off. The bases of Q7 and Q8 are pulled up passively by resistor RP, while the emitters of Q7 and Q8 are pulled down actively by the collectors of Q9 and Q11. In response, Q7 and Q8 now conduct current and begin to equalize their base-to-emitter voltages. The process of equalization can be explained in the following manner. If the capacitor voltage is positive, Q8 turns on harder than Q7 and pushes up the right side of the charge pump relative to the left side. Similarly, if the capacitor voltage is negative, Q7 turns on harder than Q8 and pushes up the left side of the charge pump relative to the right side. At equilibrium, both ends of the capacitor will be at the same potential. Ignoring slewing effects, the capacitor voltage is reset to zero with a time constant:

\[
\tau_r = \frac{2C1}{g_m} \tag{5.44}
\]

where \(g_m\) is the transconductance of Q7 and Q8. Because the bias current IA can be relatively large, \(g_m\) may be quite large, and \(\tau_r\) quite small. In the current implementation, \(IA=200\ \mu A\), and \(C1=30\ pF\). For these values of IA and C1, \(\tau_r\) is approximately 7.5 ns at a temperature of 300 K. (Because \(\tau_r\) is negligibly small, its temperature dependence can be ignored.)
The timing between control signals VR and VR2 must be arranged carefully in order not to saturate some of the transistors in the charge pump and the negative resistance circuit. The issue is best explained through an example. Imagine that VR and VR2 become high at the same time, so that differential pairs Q9-Q10, Q11-Q12, and Q13-Q14 all switch simultaneously. Because of capacitance, the bases of Q7 and Q8 are pulled up to the positive supply with a nonzero time constant. Consequently, Q7 and Q8 remain in cutoff for a while. Meanwhile, Q9 and Q11 are pulling large currents out of nodes VOUTPLUS and VOUTMIN, which causes the common-mode voltage on the capacitor to drop. (Because of the lack of common-mode capacitance, the common-mode voltage can drop quickly.) Eventually, Q7 and Q8 become active and restore the common-mode voltage to one base-to-emitter voltage below the positive supply, but during the transient the differential pairs of the charge pump and the current sources of the negative resistance circuit can be saturated.

This problem is avoided if VR2 becomes high one half of a clock period before VR becomes high. In this case the bases of Q7 and Q8 are already near the positive supply by the time Q9 and Q11 draw large currents out of nodes VOUTPLUS and VOUTMIN. Therefore the common-mode voltage on the capacitor can no longer drop more than one base-to-emitter voltage below the positive supply. Similarly, when the reset circuitry is disabled, VR2 should become low one half of a clock period after VR becomes low.

A simple digital circuit which derives the signals VR and VR2 from the user-supplied signal detect bit is shown in Figure 5.11. As shown in the timing diagram of Figure 5.12, VR2 becomes high one half of a clock period before VR becomes high. Conversely, VR2 becomes low one half of a clock period after VR becomes low.

The reset circuitry was simulated on ADICE. Figure 5.13 shows how the voltage on the charge pump capacitor is reset to 0 V when VR2 becomes high at t=112.5 ns, and VR becomes high at t=125 ns. As evident in the figure, the reset circuitry discharges the capacitor to less than 10 mV within 2 clock periods (recall that the clock period is 25 ns).
Figure 5.11: Digital Controller for Reset Circuitry

Figure 5.12: Timing Diagram for Digital Controller
Figure 5.13: Charge Pump Capacitor Voltage during Reset

5.1.6 The Complete Differential Charge Pump

The complete schematic of the differential charge pump is shown in Figure 5.14. Transistors Q15 through Q20 constitute an output buffer for the charge pump. The buffer’s most important feature is its high input resistance (> 100 MΩ), for resistive loading by the output buffer degrades the matching between positive and negative resistances and therefore reduces the d.c. gain of the phase detector. The low frequency gain of the output buffer is approximately 2.7.

As discussed in the last chapter, two higher-order poles with a time constant of 35 ns provide near optimum ripple filtering. The two resistors labelled RS1 and the two capacitors labelled C2 implement the first such higher-order pole. The capacitors labelled C3 and C3a implement the second. By adding common-mode capacitance to nodes VOUTP and VOUTN, the two capacitors labelled C3a also improve the high frequency common-mode rejection of the output buffer.

In order to match the parasitic capacitances to the substrate on each side of the charge pump, capacitor C1 is actually composed of two parallel 15 pF capacitors which are connected oppositely. While not explicitly shown in the schematic, capacitor C3 is also composed of two parallel capacitors connected oppositely.

For the final design, the bias currents should be supply-independent, and their temperature dependence should be well-defined. (For instance, the bias currents for the differential pairs of the charge pump should be independent of temperature,
Component Values:

- R1 = 10 Kohms
- R2 = 1.8 Kohms
- R3 = 330 Kohms
- R4 = 750 Kohms
- R5 = 48 Kohms
- R6 = 16 Kohms
- R7 = 10 Kohms
- R8 = 2 Kohms
- R9 = 1 Kohm
- R11 = 330 ohms
- R12 = 6 Kohms
- R13 = 4.3 Kohms
- R14 = 5 Kohms
- R15 = 7.5 Kohms
- RE = 600 ohms
- C1 = 30 pF
- C2 = 5 pF
- C3 = 20 pF
- C3a = 5 pF
- RCC = 120 Kohms

Figure 5.14: Complete Schematic of Differential Charge Pump
while those for the negative resistance circuit should be PTAT.) However, because known techniques exist for realizing both temperature-independent and PTAT current sources, the details of biasing were not considered in this thesis. While the simple current sources shown in Figure 5.14 are only accurate at one temperature and one supply voltage, they are useful for simulating the finite output impedance of real current sources. For the component values listed in the figure, the current sources are accurate at a temperature of 300 K and a supply voltage of 4.5 V.

In order to minimize common-mode glitches on the filtered output of the phase detector (i.e., nodes VOUTP and VOUTN), the current sources for the output buffer should be biased independently of the current sources for the differential pairs of the charge pump. Consider the simple current sources shown in Figure 5.14. Because of capacitive coupling, the voltage on the bias line driving Q30 through Q34 exhibits a negative spike each time one of the differential pairs of the charge pump switches. As a result, all of the current sources attached to the bias line are modulated. If the current sources for the output buffer were also attached to this bias line, the filtered output of the phase detector would have common-mode glitches. By biasing the current sources for the output buffer independently, as suggested in Figure 5.14, this problem is avoided. (The differential pairs of the reset circuitry can be ignored, for they are inactive when incoming data is present.)

5.2 The Digital Control Logic of the Phase Detector

5.2.1 Implementation of Flip-Flops with Latches

As is well-known, an edge-triggered D-type flip-flop can be formed by connecting two D-type latches in a master-slave configuration. Figure 5.15 shows how the 3 flip-flops of the phase detector logic can be implemented with 6 latches.

However, in terms of minimizing power dissipation and chip area, the circuit shown in Figure 5.15 is not optimal. Because latches U4B and U5A are both gated by the noninverted clock, they perform identical functions. U5A is transparent when U4B is transparent, and U5A is latched when U4B is latched. Because it is redundant, latch U5A can be eliminated without altering the function of the phase detector. Similarly, latch U6A can be eliminated. Therefore, the phase detector logic can be implemented with only 4 latches, as shown in Figure 5.16.
Figure 5.15: Implementation of Flip-Flops with 6 Latches

Figure 5.16: Implementation of Phase Detector Logic with 4 Latches
5.2.2 D-Type Latch

The transistor-level design of the D-type latch is shown in Figure 5.17.

![D-Type Latch Circuit Diagram]

Figure 5.17: D-Type Latch

Completely differential operation permits the use of reduced logic swings, which enhances switching speed. In the present design, the logic swing is just over 300 mV.

As mentioned in the section on the differential charge pump, the details of biasing have not been considered in this thesis. The simple current source shown in Figure 5.17 is useful for simulating the latch at one temperature and one supply voltage (300 K and 4.5 V for the value of R5 shown in the figure).

Resistor R6 ensures that a small bias current (about 100 µA) flows through differential pair Q1-Q2 when the latch is transparent. By keeping either Q1 or Q2 in forward bias at all times, this technique minimizes the glitch on the output when the latch switches from the transparent to latched condition. It should be noted that this extra bias current adds constructively to the output signal when the latch is transparent. For instance, if the D input is high, the Q output of the latch is high (and the Q output is low) when the latch is in the transparent condition. Therefore Q1 is on and Q2 is off. The collector current of Q1 lowers the Q output further, thereby
increasing the differential output signal. Hence, the addition of resistor R6 does not interfere with the basic operation of the latch.

In order to interface with two-level gate structures such as the standard ECL XOR gate, level shifted versions of the output signal are required. The level shifted signals QX and Q\overline{X} are useful for driving the lower-level inputs of the XOR gates of the phase detector logic.

5.2.3 XOR Gate

The transistor-level design of the XOR gate is shown in Figure 5.18.

Figure 5.18: XOR Gate

Again the signal path is completely differential so that reduced logic swings can be used. In the current design, the logic swing is just over 300 mV.

Recall that the differential outputs of the three XOR gates drive the differential pairs of the charge pump. Resistors R3 and R4 level shift the outputs of the XOR gate to a level appropriate for the charge pump. A bandgap reference can be used to obtain the bias voltage VGO, which is approximately 1.2 V. Again the details for realizing such biasing have not been carefully studied.
One important characteristic of the standard XOR gate is that its two inputs are not matched in terms of propagation delay. When the lower-level input DS changes state, the finite speed of both the lower differential pair and the upper differential pairs contributes to the total propagation delay. But when the higher-level input DF changes state, the finite speed of the lower differential pair does not contribute to the total propagation delay, for the lower differential pair does not even switch. For this reason, the propagation delay from the lower-level input to the output is larger than the propagation delay from the higher-level input to the output. According to ADICE simulations, the difference in delay is approximately 300 ps for the XOR gate shown in Figure 5.18. In the notation of Figure 5.18, the S in DS stands for “Slow”, and the F in DF stands for “Fast”.

This asymmetry in propagation delay affects the operation of the phase detector. Assume that the left input of XOR gate U1 is a slow input, and that the right input of U1 is a fast input, as shown in Figure 5.19. In this case the positive pulse on U1’s output is narrowed, for its rising edge is delayed more than its falling edge. It is important to realize, however, that the narrowing of the pulse on U1’s output does not necessarily contribute to phase detector offset. If all of the left inputs of the XOR gates are slow inputs, as shown in Figure 5.19, the pulses on the outputs of U1, U2, and U3 will be narrowed equally, and the offset of the phase detector will not be affected. Phase detector offset is only affected when the XOR gates are connected inconsistently; e.g., when the left inputs of U1 and U3 are slow inputs, but the left input of U2 is a fast input.

![Diagram](image)

Figure 5.19: Correct Way of Connecting XOR Gates with Asymmetrical Delays
It is interesting to compare the topology of the latch (Figure 5.17) with that of the XOR gate (Figure 5.18). In the case of the latch, a lower differential pair, driven by the clock signal, is used to switch currents between one of two upper differential pairs. In the case of the XOR gate, a lower differential pair, driven by the slow input DS, is used to switch currents between one of two upper differential pairs. Because of the topological similarities, one might expect the clock-to-Q delay of the latch to match closely the propagation delay from DS to output of the XOR gate - at least if the bias currents for the two circuits are similar. If this expectation were correct, the delay compensation discussed in Chapter 4 could be implemented with an XOR gate\(^9\), as shown in Figure 5.20.

![Diagram of XOR gate as compensating delay](image)

Figure 5.20: Use of XOR Gate as Compensating Delay

In order to check the matching between the clock-to-Q delay of the latch and the delay from DS to output of the XOR gate, the entire circuit shown in Figure 5.20 was simulated on ADICE. By simulating the entire circuit (as opposed to simulating the latch and the XOR gate separately), the effects of capacitive loading were accounted for. According to the simulation, the clock-to-Q delay of the latch equals 1.06 ns, while the delay from DS to output of the XOR gate equals 1.00 ns. Thus the compensating delay matches the clock-to-Q delay within 60 ps.

\(^9\)Because the XOR gate used for delay compensation drives another logic gate instead of the charge pump, its output is level shifted not with resistors, as shown in Figure 5.18, but with diode-connected transistors, as shown in the schematic of the latch, Figure 5.17.
While mismatches in device speed can degrade this performance, a worst case analysis shows that the offset of the phase detector due to timing errors in the digital control logic should be less than 300 ps (4.3 degrees). It should be mentioned that the offset of the phase detector due to timing errors in the digital control logic does not depend on the data transition density, and therefore such timing errors do not cause data-dependent jitter.

As noted earlier, the total static phase error due to errors in the charge pump is expected to be less than 520 ps (7.5 degrees) under all conditions. Including timing errors in the digital control logic, the total static phase error of the DLL or the D/PLL is expected to be less than 820 ps (11.8 degrees) under all conditions.

5.3 Simulated Performance of Complete Phase Detector

5.3.1 Open Loop Results

The complete phase detector, including the differential charge pump (Figure 5.14) and the digital control logic (Figures 5.20, 5.17, and 5.18), was simulated on ADICE. Figure 5.21 shows the delayed data, the clock, and the charge pump capacitor voltage when one transition of the delayed data occurs 60 ps after the falling edge of the clock. As the figure suggests, there is no net change in the voltage on the charge pump capacitor when the delayed data is 60 ps behind the clock. Hence, in the absence of resistor and device mismatches, the phase detector has an offset of 60 ps. This offset is a simple consequence of the compensating delay being 60 ps smaller than the clock-to-Q delay of the latches, as discussed in the last section.

As explained in Chapter 4, the most important feature of the triwave is that it has zero net area. Because it is highly symmetric, the triwave waveform shown in Figure 5.21 satisfies this requirement.

While open loop tests such as the one represented in Figure 5.21 are useful for determining phase detector offset (and perhaps \( K_D \)), it is generally more expedient and more natural to evaluate the phase detector in a closed loop connection. For instance, by creating an ideal voltage-controlled phase shifter in ADICE, the phase detector's operation inside a DLL can be simulated. The basic operation of such an ideal phase shifter is described next.
Figure 5.21: Phase Detector Waveforms: Delayed Data 60 ps Behind Clock
5.3.2 Ideal Phase Shifter

One technique for delaying digital signals, which is used in Analog Devices' AD9500 Delay Generator, is illustrated by the circuit shown in Figure 5.22.

![Circuit Diagram](image)

**Figure 5.22: The Basic Voltage-Controlled Delay Element**

The circuit delays the rising edge of the incoming data in the following manner. Upon each rising edge of the incoming data, flip-flop U1 raises its output high, thereby opening switch S1. Current then flows out of the capacitor, and the capacitor voltage ramps downward at a rate of $-I_0/C$. When the capacitor voltage drops below $-V_1$, U2 raises its output high. After a short time determined by $\Delta V$, U3 raises its output high and asynchronously resets flip-flop U1, thereby closing switch S1. Because the capacitor voltage is reset to 0 V, both comparators lower their outputs\(^{10}\). The timing diagram for this circuit is given in Figure 5.23.

Ignoring logic and comparator delays, the delay $t_d$ between the rising edge of the incoming data and the rising edge of the pulse on U2’s output equals:

$$ t_d = \frac{V_1}{(I_0/C)} = \frac{C V_1}{I_0} \quad (5.45) $$

By making $V_1$ a linear function of the control voltage, an adjustable delay is realized.

\(^{10}\)In practical implementations of this circuit, such as the AD9500, the output of comparator U2 could be used to reset the flip-flop, and comparator U3 could be eliminated. Logic and comparator delays would ensure that the pulse on U2's output have nonzero width.
Figure 5.23: Timing Diagram for the Basic Delay Element

The range of this delay element is limited, as \( t_d \) cannot be larger than two clock periods. Because this delay element can only process one rising edge of the incoming data at a time, the flip-flop must be reset before the next rising edge occurs. (Otherwise, the next rising edge would be missed.) Since the time between consecutive rising edges may be as small as two clock periods (e.g., if the data has maximum transition density), \( t_d \) is restricted to the range \( 0 < t_d < 2T \), where \( T \) is the clock period (25 ns). Ignoring logic and comparator delays, as well as the width of the pulse on U2's output, the total range of this delay element is \( 4\pi \) radians.

In order to produce a delayed version of the incoming data, the phase shifter must delay both rising and falling edges. For this reason, the phase shifter includes two delay elements, as shown in Figure 5.24. The upper delay element is triggered by the rising edges of the incoming data, the lower delay element by the falling edges. By connecting the outputs of U2 and U5 to set-reset flip-flop U7, a delayed version of the incoming data is reconstructed (see the timing diagram of Figure 5.25).

The entire voltage-controlled phase shifter of Figure 5.24 was created in ADICE from ideal current sources, ideal capacitors, and ideal voltage-controlled switches. (The ideal switch model is a special feature of the ADICE program.) Because the ideal switches can change state instantaneously, logic and comparator delays are nonexistent, and the total range of the phase shifter is \( 4\pi \) radians. As explained in Chapter 3, a range of \( 3\pi \) radians is usually sufficient.
Figure 5.24: Ideal Phase Shifter

Figure 5.25: Timing Diagram for Ideal Phase Shifter
5.3.3 Closed Loop Results

Using the ideal phase shifter described above, the phase detector's operation inside a DLL was simulated on ADICE. The clock frequency was 40 MHz. Figure 5.26 shows the voltage on the charge pump capacitor during acquisition; the initial phase error between the delayed data and the clock is slightly less than $\pi$ radians. The data burst includes a preamble code, as the incoming data has maximum transition density for approximately 1 $\mu$s. After 1 $\mu$s the transition density drops to 1 transition every 4 clock periods, and after 1.5 $\mu$s the transition density returns to 1 transition every clock period.

The filtered output of the phase detector during the same acquisition is shown in Figure 5.27. (The difference in signal levels between Figures 5.26 and 5.27 is attributable to the gain of the output buffer of the charge pump.) The figure clearly demonstrates how the two higher-order poles attenuate high frequency fluctuations in the phase shifter's control voltage.

The lack of overshoot evident in Figures 5.26 and 5.27 indicates that the DLL is critically damped. Hence, for the given loop bandwidth, the sampling nature of the phase detector does not significantly degrade loop stability.

Figure 5.28 illustrates the alignment between the delayed data and the clock after 900 ns. Since 0 V is the threshold voltage for differential ECL logic, the phase error after 900 ns is less than 200 ps (2.9 degrees). Therefore the requirement that acquisition occur within 1 $\mu$s is satisfied.

For all practical purposes, the DLL has settled to its final state by 1.73 $\mu$s. For this reason, the phase error between the delayed data and the clock after 1.73 $\mu$s, shown in Figure 5.29, represents the static phase error of the loop. Since the offset of the phase detector was shown to be only 60 ps in the open loop test, the low value (< 100 ps) of static phase error evident in the figure indicates that the d.c. gain of the phase detector is large (i.e., the negative resistance circuit is working well).

In order to demonstrate the necessity of the negative resistance circuit, the acquisition of the DLL was also simulated with the negative resistance circuit disabled. The effect of the negative resistance circuit was cancelled by shunting the charge pump capacitor with a 20 k$\Omega$ resistor. Figure 5.30 shows the voltage on the charge pump capacitor in this case. The discharging of the capacitor can be seen when the data transition density drops at $t=1$ $\mu$s. The discharging of the capacitor also affects the filtered output of the phase detector, shown in Figure 5.31. The decay in control voltage when the data transition density drops adds significant jitter to the delayed data. (In the case of a D/PLL, jitter would be added to the retimed data as well.) Without the negative resistance circuit, the phase detector does not have enough d.c. gain to ensure a small static phase error, even when the data has maximum transition density. Note that while the control voltage after 900 ns approached 240 mV in the
previous simulation (Figure 5.27), the control voltage after 900 ns now approaches only 175 mV, which is too small to eliminate the initial phase error.
Figure 5.26: Voltage on Charge Pump Capacitor During Acquisition

Figure 5.27: Filtered Output of Phase Detector
Figure 5.28: Alignment between Delayed Data and Clock After 900 ns
Figure 5.29: Alignment between Delayed Data and Clock After 1.73 μs
Figure 5.30: Charge Pump Capacitor Voltage: Negative Resistance Circuit Disabled

Figure 5.31: Filtered Output of Phase Detector: Negative Resistance Circuit Disabled
Chapter 6

Breadboard Prototypes and Experimental Results

In order to evaluate the performance of an actual DLL and an actual D/PLL, breadboard prototypes were constructed. This chapter describes the prototypes and examines their measured performance.

6.1 Description of Breadboard Prototypes

6.1.1 Phase Detector Prototype

The phase detector prototype is a simplified version of the triwave phase detector described in the last chapter. Its schematic is presented in Figure 6.1. In the figure devices with the same unit number (such as U1A and U1B) belong to a common package. In order not to clutter up the schematic, the termination resistors for the ECL logic are not shown in the figure.

The principal difference between the prototype and the phase detector described in the last chapter is in the negative resistance circuit. The fully compensated negative resistance circuit described in the last chapter (Figure 5.8) uses 8 matched transistors. The use of so many matched transistors is practical in an integrated circuit but not in a prototype built from discrete components. For this reason a simpler negative resistance circuit, consisting of transistors U13A-E, is employed in the prototype. Because the prototype is only designed for operation at room temperature, any mismatch between positive and negative resistances in the charge pump can be eliminated by increasing or decreasing the upper bias resistors via trimpots R8 and R10. In addition, by trimming R8 and R10 oppositely, the effects of current source mismatches in the charge pump can also be cancelled.

Other differences include the following. To minimize the effect of board capaci-
Figure 6.1: Schematic of Phase Detector Prototype
tance, the currents in the charge pump of the prototype are 10 times larger than the values given in the last chapter, and the resistors and capacitors are scaled accordingly. (Note, for instance, that the charge pump capacitor is now 300 pF instead of 30 pF.) Because the prototype uses commercially available logic circuits and operational amplifiers, the prototype does not operate off a single supply. Like the other breadboard prototypes, the phase detector prototype uses five power supplies: 20, 7, 2, -3.2, and -13 V. Capacitors C1 and C3 simulate the effect of parasitic capacitances to the substrate (again scaled appropriately). Finally, the use of JFET's (U430 from Siliconix) simplifies the output buffer considerably. The gain of the output buffer is adjusted by changing the bias current for the JFET differential pair via trimpot R23.

The charge pump and reset circuitry are implemented with arrays of 1 GHz transistors (CA3127 and CA3102). In the CA3102 arrays, the transistors are wired internally as current source biased differential pairs. Because of the lack of devices with scaled areas, two differential pairs in parallel (U12A and U1B) are used to realize the middle differential pair of the charge pump. Because high frequency transistors are not required in the negative resistance circuit, the negative resistance circuit is implemented with an array of 200 MHz transistors produced on one of Analog Devices' low speed processes. (Since high frequency transistors have a tendency to oscillate in the presence of stray capacitance or inductance, it is generally unwise to employ faster transistors than the application requires – particularly on a breadboard.)

The differential output of the JFET buffer is converted into a single-ended signal by an AD842 operational amplifier. This amplifier, which possesses a gain bandwidth of 80 MHz, is fast enough to have no significant effect on the loop dynamics of the DLL or D/PLL. The diode bridge on the amplifier's output restricts the control voltage to a range between 2.2 and 3.7 V so that the phase shifter is not overdriven. By placing the bridge inside the feedback loop of the amplifier, a clamp with very sharp corners is realized. The 2.2 and 3.7 V supplies are generated by two LM317 adjustable voltage regulators (not shown). The only disadvantage with placing the bridge inside the feedback loop is that the amplifier is open loop when the control voltage is clamped. If not for diodes D1-5, the amplifier would saturate. By forcing the amplifier into current limiting, however, diodes D1-5 prevent the amplifier from saturating. Such a technique is only useful because the AD842's recovery from current limiting is better behaved than its recovery from saturation. While admittedly crude, the technique worked well enough for the purposes of prototyping.

The design of the reset circuitry implicitly assumes that 0 V on the charge pump capacitor corresponds to the middle of the phase shifter's range. JFET current source Q1 permits the offset adjustment needed to ensure this.

In the case of the D/PLL, the control voltage drives both the phase shifter and the VCXO. An AD711 operational amplifier buffers the control voltage for the VCXO.

All of the logic components used belong to Motorola's 10KH ECL family, which
features typical gate delays of 1 ns. Operating the ECL logic off 2 and -3.2 V supplies allows simple termination of the transmission lines over which the digital signals travel; only a resistor to the ground plane is required for proper termination. Normally, when ECL logic is operated off a single -5.2 V supply, the transmission lines must be terminated with a resistor to -2 V, not ground.) As shown in the schematic, differential line receivers (MC10116) are used extensively to buffer incoming and outgoing digital signals. As in the phase detector described in the previous chapter, an XOR gate (U15A) is used for delay compensation.

6.1.2 Phase Shifter Prototype

The phase shifter prototype employs four AD9500 Delay Generators, as shown in the schematic of Figure 6.2. Each AD9500 is essentially a realization of the ideal voltage-controlled delay element shown in Figure 5.22. As in the ideal phase shifter (Figure 5.24), set-reset flip-flops (U3A and U15A) are used to reconstruct delayed versions of the incoming data.

In terms of performance, the most important difference between the AD9500 and the ideal delay element of Figure 5.22 is that the AD9500 does not reset the capacitor to 0 V instantaneously when switch S1 is closed. Instead, the capacitor is discharged with a nonzero time constant. If too little time is allowed for the capacitor voltage to settle to 0 V, the next rising edge will be delayed less than the amount predicted by Equation 5.45. The magnitude of this error depends on the data transition density, for at high transition densities less time is available for discharging the capacitor. Because the delay through the AD9500 depends upon the data transition density, data-dependent jitter is added to the delayed data. In order not to add significant data-dependent jitter to the delayed data, at least 15 ns must be allotted for resetting the capacitor when the data has maximum transition density. This requirement limits the maximum delay through the AD9500 to 35 ns. Because of logic and comparator delays, the minimum delay through the AD9500 is 10 ns. Therefore the useful range of each AD9500 is only 25 ns, which is only one clock period. To achieve a larger range of delay, the phase shifters must be cascaded as shown in the schematic. By cascading phase shifters, the total range of the prototype is doubled to 50 ns (4π radians), which is more than adequate.

The delay through each AD9500 is adjusted by changing the voltage on its offset adjust pin (pin 6), as this voltage sets the threshold of the comparator inside the AD9500. Simple common-emitter amplifiers with emitter degeneration (U10A) drive the offset adjust pins of the four AD9500's.

Because the rising and falling edges of the incoming data are delayed by different
Figure 6.2: Schematic of Phase Shifter Prototype
AD9500’s, the gains and offsets of the AD9500’s must be adjusted to ensure that the rising and falling edges are delayed equally\(^1\). The external resistor on pin 18 of the AD9500 sets the current used to generate the downward ramp in capacitor voltage. By changing this current, the gain of the AD9500 is adjusted (see Equation 5.45). Trimpots R1, R2, R21, and R22 are used to make this adjustment. The offset of each AD9500 is adjusted digitally, for each AD9500 contains an 8-bit digital-to-analog converter which can be used to offset the comparator threshold. Dipswitches SW1-4 are used to program in the desired offset adjustment.

### 6.1.3 VCXO Prototype

The VCXO prototype, whose schematic is shown in Figure 6.3, employs a Damon 7718WXA-X VCXO having a center frequency of 77.42 MHz, a tuning range of \(\pm 100\) ppm, and a modulation bandwidth exceeding 20 KHz. (Modulation bandwidth describes the speed with which the frequency of an oscillator can be changed.) Flip-flop U3A divides the frequency of the VCXO by 2, thereby producing a tunable 38.71 MHz clock with a 50% duty cycle.

As explained in Chapter 3, the control range of the VCXO should be a subset of the phase shifter’s control range. Operational amplifier U4A provides the gain of 10 needed to ensure this. Zener diodes D1 and D2 limit the control voltage of the VCXO to desirable levels. Current source Q1 is used to adjust the offset of the VCXO prototype so that 0 V on the charge pump capacitor corresponds to the middle of the VCXO’s range.

### 6.1.4 Circuit Board Layout Techniques

All of the breadboard prototypes were constructed on double-sided printed circuit boards. The top sides of the boards were used almost entirely for ground planes, as large ground planes are necessary for good high frequency bypassing and proper microstrip operation. Large value (10-20 \(\mu\)F) tantalum electrolytic capacitors bypass the power supplies to the ground plane on each circuit board. The power supply leads of individual devices are bypassed to the local ground plane by ceramic disk (0.1 \(\mu\)F) and ceramic chip (1000 pF) capacitors. Microstrip lines with a characteristic impedance of 100 \(\Omega\) carry the ECL signals on each circuit board. Coaxial cables with a characteristic impedance of 50 \(\Omega\) and with SMA-type fittings are used for board-to-board interconnections. Microstrip lines with a characteristic impedance of 50 \(\Omega\) were used to match the impedance of the coaxial cables. Photographs of the prototypes are presented in Figures 6.4 – 6.6.

\(^1\)Provided that the delay through the AD9500 is a linear function of the control voltage, the gain and offset of the AD9500 characterize its transfer characteristic completely.
Figure 6.3: Schematic of VCXO Prototype
Figure 6.1: Photograph of Phase Detector Prototype

Figure 6.2: Photograph of Phase Shifter Prototype
6.2 Open Loop Measurements

6.2.1 Phase Shifter Measurements

The delay through the phase shifter was measured with a Tektronix 11302A Counter Timer Oscilloscope. The digital outputs of a LeCroy 9109 Arbitrary Function Generator were used as the source of incoming data. To avoid errors due to incomplete discharging of the capacitors inside the AD9500's, the data transition density was first set at 1 transition every 20 clock periods. The delay through the phase shifter as a function of control voltage is plotted in Figure 6.7. For control voltages greater than 2.2 V, the phase shifter exhibits excellent linearity, and the rising and falling edges are delayed almost identically. The degraded linearity at low values of control voltage is a consequence of the AD9500's internal voltage ramp not being linear in its start up region [311]. Ignoring control voltages less than 2.2 V, the gain of the phase shifter is 31.3 ns V (7.87 rad V).

To evaluate the errors due to incomplete discharging of the capacitors inside the AD9500's, the data transition density was then varied between 1 transition every clock period and 1 transition every 4 clock periods. Figure 6.8 shows how the delay of a rising edge through the phase shifter is affected by changes in data transition density. The delay of a falling edge through the phase shifter is affected similarly. For data transition densities less than or equal to 1 transition every 2 clock periods,
Figure 6.7: Transfer Characteristic of Phase Shifter Prototype
Figure 6.8: Effect of Data Transition Density on Delay Through Phase Shifter
sufficient time is allowed for discharging the capacitors, and the measured delays do not differ significantly from those shown in Figure 6.7. But when the data transition density equals 1 transition every clock period, the delay through the phase shifter is reduced because less time is available for discharging the capacitors. The effect is most noticeable at high values of control voltage, for large delays through the AD9500's leave little time for resetting the capacitors.

The resultant data-dependent jitter added to the delayed data is shown in Figure 6.9. In this experiment the control voltage equalled 3.6 V, and the data transition density was alternated between 1 transition every clock period and 1 transition every 4 clock periods. Because the oscilloscope (Tektronix 11402) was triggered on the rising edges of the incoming data, the jitter depicted in the figure represents the jitter added to the delayed data by the phase shifter. The two distinct traces evident in the figure reflect the fact that the delay through the phase shifter is different for the two transition densities. The separation between the traces indicates that the data-dependent jitter added to the delayed data by the phase shifter is approximately 600 ps (peak-to-peak) when the control voltage equals 3.6 V. As will be shown later, the non-ideality of the phase shifter limits the jitter performance of the DLL and the D/PLL prototypes.
Figure 6.9: Data-Dependent Jitter Added to Delayed Data by Phase Shifter
6.2.2 VCXO Measurements

The frequency of the VCXO was measured with the Tektronix 11302A Counter/Timer Oscilloscope. The deviation of the VCXO's frequency from its nominal center frequency (38.71 MHz) as a function of control voltage is plotted in Figure 6.10. Comparing this figure with Figure 6.7, one sees that the control range of the VCXO is a subset of the control range of the phase shifter; therefore the D/PLL prototype can lock to any data stream whose bit rate is within the tuning range of the VCXO, as explained in Section 3.2.4.

As explained in Chapter 3, the bandwidth of the jitter transfer function of the D/PLL is approximately $K_O/K_\phi$. Because of the nonlinearity evident in Figure 6.10, the incremental gain $K_O$ of the VCXO depends upon operating point. For this reason the bandwidth of the jitter transfer function of the D/PLL is affected by the bit rate of the incoming data. The bandwidth of the jitter transfer function is highest when the bit rate of the incoming data is very close to 38.71 Mbit/s so that (after acquisition) the control voltage is in the middle of the VCXO's range. Under this condition $K_O$ equals 11.2 KHz/V, and $K_O/K_\phi$ equals 1.43 KHz. (The gain of the phase shifter $K_\phi$ was given in the last section.) Thus, the bandwidth of the jitter transfer function of the D/PLL prototype is less than or equal to 1.43 KHz under all conditions.

![Frequency of VCXO vs. Control Voltage](image)

Figure 6.10: Transfer Characteristic of VCXO Prototype
6.3 DLL Measurements

The operating frequency of the DLL prototype is 40 Mbit/s. The 40 MHz frequency reference for the DLL is derived from an 80 MHz Vectron CO-633A crystal oscillator. By dividing the frequency of the crystal oscillator by 2, a 40 MHz clock with a 50% duty cycle is obtained. Incoming data whose bit rate exactly equals the clock frequency is produced by triggering the 9109 LeCroy Arbitrary Function Generator on the clock signal.

6.3.1 Acquisition

Figure 6.11 shows the control voltage of the DLL during acquisition. The requirement that acquisition occur within 1 μs is clearly satisfied. The lack of overshoot evident in the figure indicates a high degree of loop stability. Furthermore, the measured transient on the control voltage is similar to the transient simulated in ADICE (compare with Figure 5.27).

A simple test circuit employing two AD9500’s was used to change the initial phase error prior to acquisition. To evaluate the DLL’s performance near metastability, the initial phase error was set close to π radians. Even when the incoming data had minimal jitter, metastability was readily observed. After the signal detect bit was asserted, and the reset circuitry was disabled, the phase detector did not always decisively drive the loop away from its metastable point. Figure 6.12 shows the control voltage during metastability. By equivocating, the loop sometimes dwelled in the vicinity of the unstable null over 1 μs, thereby violating the acquisition time requirement. (As explained in Chapter 4, equivocation occurs when the phase detector produces alternating positive and negative error voltages, whose effects tend to cancel.) The trajectories of the control voltage during metastability are distinct because the phase detector measures the phase error only at discrete instants of time.

The severity of this problem was unexpected; apparently the two higher-order poles at 35 ns significantly degrade the DLL’s performance near metastability by reducing the speed with which the delay through the phase shifter can be changed. While removing the higher-order poles might improve the DLL’s performance near metastability, the jitter on the delayed data would be increased greatly by such a change. Clearly a better solution to the metastability problem would be to modify the triwave phase detector so that equivocation is avoided during acquisition. Appendix B describes such a modification.

6.3.2 Static Phase Error

Figure 6.13 shows the alignment between the delayed data and the clock after acquisition is completed. As shown in the figure, the data transitions are just over 1
Figure 6.11: Control Voltage of DLL During Acquisition
Figure 6.12: Control Voltage of DLL During Metastability
Figure 6.13: Steady State DLL Waveforms
ns ahead of the falling edges of the clock. This 1 ns (14.4 degree) phase error was determined to be the result of several small delay mismatches in the phase detector logic. Consistent with this determination, no dependence of static phase error on data transition density was observed. Presumably, a triwave phase detector implemented on a single integrated circuit would have lower offset because of the high degree of matching afforded by monolithic construction. For the phase detector described in the last chapter, the offset due to timing errors in the digital control logic is expected to be less than 300 ps. (As explained in the last chapter, errors in the charge pump also cause static phase error.)

6.3.3 Jitter Performance

The jitter on the delayed data when the data transition density equals 1 transition every clock period is shown in Figure 6.14. The Tektronix 11402 oscilloscope was triggered on the 40 MHz frequency reference for this measurement. Since the incoming data was a periodic waveform in this experiment, the jitter shown in the figure is random jitter and arises from physical noise processes. Because of the probabilistic nature of random jitter, the peak-to-peak jitter observed on a storage oscilloscope is a function of the observation period. For the sake of consistency, 8 hour observation periods were used for all of the jitter measurements taken on the DLL and the D/PLL. The peak-to-peak jitter shown in the figure is approximately 450 ps.

Figure 6.15 shows the jitter on the delayed data when the data transition density is alternated between 1 transition every clock period and 1 transition every 4 clock periods at a rate slow relative to the loop bandwidth of the DLL. (As explained in Chapter 4, such a data pattern produces the greatest data-dependent jitter.) The peak-to-peak jitter (1.2 ns) is now greater because there is a significant amount of data-dependent jitter on the delayed data. The dominant source of this data-dependent jitter is the non-ideality of the phase shifter. As shown in Figure 6.9, the data-dependent jitter added to the delayed data by the phase shifter may be as great as 600 ps (peak-to-peak). In this manner the non-ideality of the phase shifter limits the jitter performance of the DLL prototype. If the phase shifter were improved so that it does not add data-dependent jitter to the delayed data, the total data-dependent jitter on the delayed data would be less than 400 ps (peak-to-peak), most of which would be caused by errors in the charge pump.
Figure 6.14: Random Jitter on Delayed Data of DLL
Figure 6.15: Random and Data-Dependent Jitter on Delayed Data of DLL
6.4 D/PLL Measurements

The D/PLL prototype operates at any bit rate within the tuning range of the 38.71 MHz VCXO prototype. Incoming data whose bit rate can be modulated about 38.71 Mbit/s is produced by triggering the 9109 LeCroy Arbitrary Function Generator on the output of a Hewlett-Packard 8656B Signal Generator.

6.4.1 Phase Acquisition

The transient on the control voltage of the D/PLL when the initial phase error is close to $\pi$ radians is shown in Figures 6.16 – 6.18. As discussed in Chapter 3, the first stage of phase acquisition of the D/PLL is essentially the acquisition process of a DLL. Hence, it is not surprising that the transient on the control voltage during the first stage of phase acquisition is similar to the acquisition transient of the DLL (compare Figures 6.11 and 6.16). As in the case of the DLL, the initial phase error is reduced to a negligible value within 1 $\mu$s. It should also be noted that the entire phase acquisition transient is similar to the transient determined theoretically in Chapter 3 (refer to Figures 3.11 and 3.12). Thus, there is good correlation between measurement and theory.

The effect of the finite modulation bandwidth of the VCXO can be seen in Figure 6.17. During the second stage of phase acquisition, the control voltage changes because the frequency of the VCXO no longer equals the bit rate of the incoming data. Due to finite modulation bandwidth, however, the frequency of the VCXO does not change as quickly as its control voltage. For a short time after the first stage of phase acquisition, the frequency of the VCXO equals the bit rate of the incoming data, and the control voltage does not change (i.e., its slope is zero). The response time shown in the figure is consistent with a 20 KHz modulation bandwidth. As Figures 6.16 – 6.18 demonstrate, in this application a 20 KHz modulation bandwidth is high enough that its effect on loop dynamics and acquisition is negligible.

No attempt was made to observe metastability in the D/PLL. Since the first stage of phase acquisition of the D/PLL is essentially the acquisition process of a DLL, there is no reason to expect the performance near metastability to be different from that of the DLL.

6.4.2 Frequency Acquisition

To observe the D/PLL's response to an initial frequency error, the bit rate of the incoming data was modulated with a square wave. Figure 6.19 shows the response on the control voltage of the D/PLL. The response is somewhat nonlinear because of the nonlinear transfer characteristic of the VCXO. Except for this nonlinearity, the frequency acquisition transient is similar to the one determined theoretically in
Figure 6.16: Control Voltage of D/PLL: First Stage of Phase Acquisition
Figure 6.17: Control Voltage of D/PLL: Beginning of Second Stage
Figure 6.18: Control Voltage of D/PLL: Second Stage of Phase Acquisition

<table>
<thead>
<tr>
<th>C2</th>
<th>Main</th>
<th>Stopped</th>
<th>Linear</th>
<th>100μs/div</th>
<th>-190μs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast</td>
<td>910240 pts</td>
<td>Stopped</td>
<td>Linear</td>
<td>100μs/div</td>
<td>-190μs</td>
</tr>
</tbody>
</table>

| 50Ω | DC | 600MHz | All Wfms Status | C2 Main | off |
Figure 6.19: Control Voltage of D.PLL: Frequency Acquisition
Chapter 3 (refer to Figure 3.13). As noted in Chapter 3, the dynamic phase error during frequency acquisition is often negligible. During the entire transient of Figure 6.19, the dynamic phase error is less than 0.6 degrees.

6.4.3 Static Phase Error

Figure 6.20 shows the alignment between the delayed data and the clock after acquisition is completed. As in the DLL, the static phase error is about 1 ns. As mentioned earlier, this phase error is a result of several small delay mismatches in the phase detector logic.

6.4.4 Jitter Performance

The jitter on the delayed data when the data transition density equals 1 transition every clock period is shown in Figure 6.21. Since the incoming data was a periodic waveform in this experiment, the jitter shown in the figure is random jitter. As in the case of the DLL, 8 hour observation periods were used for all of the jitter measurements taken on the D/PLL. The peak-to-peak jitter shown in the figure is approximately 600 ps.

The jitter on the recovered clock (i.e., the output of the VCXO) when the data transition density equals 1 transition every clock period is shown in Figure 6.22. The peak-to-peak jitter on the recovered clock equals only 200 ps. As discussed in Chapter 3, phase shifting the data instead of the VCXO’s output reduces the jitter on the recovered clock and therefore the jitter on the retimed data.

Figure 6.23 shows the jitter on the delayed data when the data transition density is slowly alternated between 1 transition every clock period and 1 transition every 4 clock periods. The peak-to-peak jitter (750 ps) is now greater because there is both random and data-dependent jitter on the delayed data. During this experiment the steady state value of control voltage was lower than when the jitter of the DLL was being measured (Figure 6.15). Because the non-ideality of the phase shifter is more significant at higher values of control voltage, in this experiment the D/PLL exhibited less data-dependent jitter than the DLL. The dependence of the data-dependent jitter on the steady state value of the control voltage is further evidence that the non-ideality of the phase shifter is the dominant source of such jitter.

Figure 6.24 shows the jitter on the recovered clock when the data transition density is alternated in the manner described above. The peak-to-peak jitter on the recovered clock is now approximately 400 ps. Again the non-ideality of the phase shifter is the dominant source of data-dependent jitter.

The non-ideality of the phase shifter increases the jitter on the recovered clock as follows. In the steady state, the control voltage of the D/PLL equals the value at
Figure 6.20: Steady State D.PLL Waveforms
which the frequency of the VCXO equals the bit rate of the incoming data. Hence, the control voltage for the phase shifter is fixed by the bit rate of the incoming data. Assuming that the delay through the phase shifter is affected by the data transition density, the phase of the delayed data relative to that of the incoming data changes with the data transition density. In the steady state, the phase error between the delayed data and the recovered clock is zero. (Recall that the loop amplifier is an integrator.) When the data transition density shifts, the phase of the recovered clock changes in order to accommodate the change in delay through the phase shifter. For this reason, even if the incoming data has no jitter, the recovered clock has data-dependent jitter.
Figure 6.21: Random Jitter on Delayed Data of D/PLL
Figure 6.22: Random Jitter on Recovered Clock of DPLL
Figure 6.23: Random and Data-Dependent Jitter on Delayed Data of D_PLL
Figure 6.24: Random and Data-Dependent Jitter on Recovered Clock of D/PLL
Chapter 7

Conclusion

Two new architectures for clock recovery have been developed: the DLL and the D/PLL. While application of the DLL requires the availability of an exact frequency reference, no such restriction applies to the D/PLL. Both the DLL and the D/PLL avoid many of the tradeoffs which limit the performance of conventional clock recovery circuits. Rapid acquisition is achieved without compromising jitter filtering. Neither the DLL nor the D/PLL exhibits jitter peaking. Static phase error is ideally zero, only limited by the offset of the phase detector and the finite d.c. gain of practical loop integrators. Excluding the crystal required for the VCO of the D/PLL, both architectures are amenable to monolithic construction.

The limitations of three phase detectors suitable for use in a DLL or a D/PLL were studied. The triwave phase detector was shown to have lower offset, a larger linear range, and better performance near metastability than the multiplier-based phase detector. Compared with the self-correcting phase detector, the triwave phase detector adds less data-dependent jitter to the delayed data. A transistor-level design of a triwave phase detector with very low offset has been presented. Its performance has been verified by computer simulation.

In order to evaluate the performance of an actual DLL and an actual D/PLL, breadboard prototypes operating at approximately 40 Mbit/s were constructed. Measurements taken on the prototypes confirmed the theoretical advantages of the DLL and D/PLL architectures. With a preamble code, both the DLL and D/PLL acquire within 1 μs (40 clock periods). For the D/PLL, the bandwidth of the jitter transfer function is less than 1.5 KHz. The jitter transfer function of the DLL simply equals zero. For both the DLL and D/PLL, the static phase error is approximately 1 ns (14 degrees). It is concluded that the DLL and D/PLL architectures offer significant improvements in performance over conventional clock recovery circuits.

The only experimental results which fell short of expectations concerned metastability and jitter performance. Even when the incoming data had minimal jitter,
metastability was readily observed in the DLL. The severity of the metastability problem was attributed (but not demonstrated) to the effect of higher-order poles in the DLL. One solution to this problem is proposed in Appendix B.

The jitter performance of the DLL and the D/PLL was limited by the non-ideality of the phase shifter. Because of the finite speed with which the capacitors inside the AD9500's are discharged, the delay through the phase shifter is affected by the data transition density. As a result, more data-dependent jitter than expected was observed on the delayed data and, in the case of the D/PLL, on the recovered clock. In order to achieve the ultimate jitter performance of which the DLL and D/PLL are capable, future efforts should be directed toward improving phase shifter operation.
Appendix A

Effect of Clock Asymmetry on Phase Detector Offset

In the main body of this thesis, only symmetrical clock waveforms were considered. The following sections examine the effect of clock asymmetry on phase detector offset for each of the phase detectors discussed in Chapter 4. It is shown that the triwave phase detector is more sensitive to clock asymmetry than either the multiplier-based phase detector or the self-correcting phase detector. For this reason, a simple modification of the triwave phase detector that solves this problem is also described.

A.1 The Multiplier-Based Phase Detector

When the duty cycle of the clock is not exactly 50%, a distinction must be made between multiplier-based phase detectors with double-balanced modulators and those with single-balanced modulators, for clock asymmetry causes data-dependent jitter when a double-balanced modulator is used. Figure A.1, which is the timing diagram for the multiplier-based phase detector with double-balanced modulator, illustrates this point. Because the multiplier is a double-balanced modulator, both the output of the nonlinear processor and the clock are treated as bipolar signals (1 or -1). (A double-balanced modulator is, essentially, an overdriven four-quadrant multiplier.) Note the direct clock feedthrough on the output of the phase detector when there are no data transitions. If the clock is not symmetric, the clock feedthrough has a d.c. component. Since the loop amplifier is an integrator, the average output of the phase detector must equal zero in the steady state. Because the phase detector produces a d.c. component when there are no data transitions, a static phase error is necessary to make the average output of the phase detector zero. Unfortunately, because the gain of the phase detector is proportional to the data transition density, the static phase error required to make the average output of the phase detector zero changes.
Figure A.1: Timing Diagram: Double-Balanced Modulator

whenever the data transition density changes. As a result, data-dependent jitter is added to the delayed data and, in the case of a D/PLL, to the retimed data.

This behavior should be contrasted with that of the multiplier-based phase detector with single-balanced modulator. When there are no data transitions, the output of the single-balanced modulator is identically zero, independent of the duty cycle of the clock (see Figure 4.2). Hence, clock asymmetry does not cause data-dependent jitter when a single-balanced modulator is used. Because of its superior performance in the presence of clock asymmetry, particular attention is now given to the multiplier-based phase detector with single-balanced modulator.

Assume that the multiplier is a single-balanced modulator, so that Figure 4.2 describes the operation of the phase detector. Since the positive pulses generated by the nonlinear processor are $T/2$ seconds wide, the output of the phase detector has zero average output when the data transitions are $T/4$ seconds ($\pi/2$ radians) ahead of the falling edge of the clock — whether or not the duty cycle of the clock equals 50%. Assume also that a $\pi/2$ radian phase shifter is employed to align the delayed data and the clock at the decision making circuit, as shown in Figure 4.5. It then follows that, under steady state conditions, the data transitions are aligned with the falling edge of the clock at the decision making circuit, even in the presence of clock asymmetry.

However, aligning the data transitions with the falling edge of the clock is only
optimal for digital regeneration when the clock is symmetrical. Fundamentally, the data transitions should be aligned halfway between the sampling (i.e., rising) edges of the clock. If the duty cycle of the clock is not 50%, the falling edge of the clock is not halfway between the rising edges of the clock. If the duty cycle of the clock equals D, the time between the falling edge and the next rising edge of the clock equals:

\[(1 - D)T\]  \hspace{1cm} \text{(A.1)}

The difference between this quantity and half a clock period is the offset of the multiplier-based phase detector with single-balanced modulator in seconds:

\[(1 - D)T - 0.5T = -(D - 0.5)T\]  \hspace{1cm} \text{(A.2)}

Since there are 2\(\pi\) radians in a clock period, the offset of the multiplier-based phase detector with single-balanced modulator equals in radians:

\[\phi_{offset} = -(D - 0.5)2\pi\]  \hspace{1cm} \text{(A.3)}

**A.2 The Self-Correcting Phase Detector**

The offset of the self-correcting phase detector in the presence of clock asymmetry is easily calculated. Refer to Figures 4.9 and 4.10. Since the pulse on the output of XOR gate U2 is initiated upon a rising edge of the clock and terminated upon the following falling edge, the width of the pulse on the output of U2 equals DT, where D is the duty cycle of the clock. Similarly, the width of the pulse on the output of XOR gate U1 equals the time between a data transition and the following rising edge of the clock. Since the pulses on the outputs of U1 and U2 must have equal widths in the steady state, the time between a data transition and the following rising edge of the clock equals DT in the steady state. The difference between DT and half a clock period is the offset of the self-correcting phase detector in seconds:

\[DT - 0.5T = (D - 0.5)T\]  \hspace{1cm} \text{(A.4)}

Accordingly, the offset of the self-correcting phase detector equals in radians:

\[\phi_{offset} = (D - 0.5)2\pi\]  \hspace{1cm} \text{(A.5)}

The offset of the self-correcting phase detector has the same magnitude but the opposite polarity as that of the multiplier-based phase detector with single-balanced modulator (cf. Equation A.3).
A.3 The Triwave Phase Detector

The offset of the triwave phase detector in the presence of clock asymmetry is also easily calculated. Refer to Figures 4.15 and 4.16. Just as in the case of the self-correcting phase detector, the width of the pulse on the output of XOR gate U2 equals DT, and the width of the pulse on the output of XOR gate U1 equals the time between a data transition and the following rising edge of the clock. Since the pulse on the output of XOR gate U3 is initiated upon a falling edge of the clock and terminated upon the following rising edge, the width of the pulse on the output of U3 equals (1−D)T. In the steady state, the average output of the phase detector must equal zero. (Recall that the loop amplifier is an integrator.) Denoting the time between a data transition and the following rising edge of the clock as δt, this requirement can be written as:

\[ 1 \times \delta t - 2 \times DT + 1 \times (1 - D)T = 0 \]  

(A.6)

Note that DT, the width of the pulse on U2’s output, is weighted twice as much as the other pulse widths. Solving for δt, one finds:

\[ \delta t = (3D - 1)T \]  

(A.7)

The difference between this quantity and half a clock period is the offset of the triwave phase detector in seconds:

\[ (3D - 1)T - 0.5T = 3(D - 0.5)T \]  

(A.8)

Accordingly, the offset of the triwave phase detector equals in radians:

\[ \phi_{offset} = 3(D - 0.5)2\pi \]  

(A.9)

The offset of the triwave phase detector is three times larger than that of the self-correcting phase detector (cf. Equation A.5).

Because the triwave phase detector is highly sensitive to clock asymmetry, even modest clock asymmetry results in large phase detector offset. For instance, if the duty cycle of the clock equals 53%, the offset of the triwave phase detector equals 32.4 degrees. Under the same conditions, the offset of the self-correcting phase detector and the offset of the multiplier-based phase detector (with single-balanced modulator) equal only 10.8 and −10.8 degrees, respectively.

In order to minimize static phase error in the presence of clock asymmetry, it is desirable to modify the triwave phase detector so that it is no more sensitive to clock asymmetry than either the multiplier-based phase detector or the self-correcting phase detector. Such a modification is described in the next section.
A.4 The Modified Triwave Phase Detector

A.4.1 Basic Operating Principle

The modified triwave phase detector is no more sensitive to clock asymmetry than either the multiplier-based phase detector (with single-balanced modulator) or the self-correcting phase detector. Figure A.2 shows the modified triwave phase detector in its simplest form. The modified triwave phase detector uses one more XOR gate (U4) and one more D-type flip-flop\(^1\) (U8) than the standard triwave phase detector.

\[\text{Phase Detector} \quad \text{Output}\]

\[
\begin{array}{c}
\sum \\
\downarrow \\
\uparrow \\
\end{array}
\]

\[
\begin{array}{cccc}
U1 & U2 & U3 & U4 \\
D \quad Q & D \quad Q & D \quad Q & D \quad Q \\
U5 & U6 & U7 & U8 \\
\end{array}
\]

\[
\begin{array}{c}
\text{Delayed Data} \\
\downarrow \\
\text{Clock} \\
\end{array}
\quad 
\begin{array}{c}
\text{Retimed Data} \\
\downarrow \\
\text{Clock} \\
\end{array}
\quad 
\begin{array}{c}
\text{Clock} \\
\end{array}
\]

Figure A.2: The Modified Triwave Phase Detector

The operation of the modified triwave phase detector is similar to that of the standard triwave phase detector. Assuming that the clock has a 50% duty cycle, the output of U2 is a positive pulse half a clock period wide for each data transition. The same is true regarding the outputs of U3 and U4. The output of U1 is also a positive pulse for each data transition, but its width depends on the phase error between the

\(^1\)As discussed in Chapter 5, implementing U6–8 as D-type latches instead of D-type master-slave flip-flops does not alter the operation of the phase detector but does reduce circuit complexity and power dissipation.
delayed data and the clock; its width equals half a clock period when the delayed data and the clock are optimally aligned. Hence, the phase error can be obtained by comparing the variable width of the pulse out of U1 with the fixed widths of the pulses out of U2, U3, and U4.

Figure A.3 is the timing diagram for the modified triwave phase detector with the delayed data and the clock optimally aligned. In this case, the output of the phase detector has zero average value; consequently, there is no net change in the output of the loop integrator. As in the standard triwave phase detector, the presence or absence of a data transition does not change the average output of the loop integrator.

If the delayed data is ahead of the clock, the output of the phase detector has a positive average value, as shown in Figure A.4. As a result, the loop integrator’s output exhibits a net increase. Conversely, if the delayed data is behind the clock, the output of the phase detector has a negative average value, and the loop integrator’s output exhibits a net decrease.

Figure A.3: Timing Diagram for the Modified Triwave Phase Detector
A.4.2 Effect of Clock Asymmetry

The offset of the modified triwave phase detector in the presence of clock asymmetry is easily calculated. Just as in the case of the standard triwave phase detector, the width of the pulse on the output of XOR gate U2 equals DT, the width of the pulse on the output of XOR gate U3 equals \((1-D)T\), and the width of the pulse on the output of XOR gate U1 equals \(\delta t\), the time between a data transition and the following rising edge of the clock. Since the pulse on the output of XOR gate U4 is initiated upon a rising edge of the clock and terminated upon the following falling edge, the width of the pulse on the output of U4 equals DT. Since the average output of the phase detector must equal zero in the steady state, one can write:

\[
1 \times \delta t - 1 \times DT - 1 \times (1 - D)T + 1 \times DT = 0 \quad (A.10)
\]

Solving for \(\delta t\), one finds:

\[
\delta t = (1 - D)T \quad (A.11)
\]
The difference between this quantity and half a clock period is the offset of the modified triwave phase detector in seconds:

\[(1 - D)T - 0.5T = -(D - 0.5)T\]  \hspace{1cm} (A.12)

Accordingly, the offset of the modified triwave phase detector equals in radians:

\[\phi_{\text{offset}} = -(D - 0.5)2\pi\]  \hspace{1cm} (A.13)

The offset of the modified triwave phase detector equals the offset of the multiplier-based phase detector with single-balanced modulator (cf. Equation A.3). Like the multiplier-based phase detector with single-balanced modulator, the modified triwave phase detector aligns the data transitions with the falling edge of the clock, even in the presence of clock asymmetry.
Appendix B

Triwave Phase Detector Modification to Avoid Equivocation

As explained in Chapter 4, even phase detectors with "sawtooth" transfer characteristics do not eliminate the problem of metastability during acquisition. When the initial phase error is near \( \pm \pi \) radians, the phase detector may produce alternating positive and negative error voltages and may not decisively drive the loop away from its metastable point. By equivocating, the loop can dwell in the vicinity of the unstable null for a long time, thereby violating acquisition time requirements. The severity of this problem was demonstrated in Chapter 6, which presented the experimental results of the DLL and the D/PLL. The following sections describe a modification of the triwave phase detector which should, in principle, solve this problem. The design counts on the availability of a preamble code consisting of alternating ones and zeros at the beginning of each data burst. It should be noted that the modification to be described can also be applied to the self-correcting phase detector.

B.1 Basic Operating Principle

In the standard triwave phase detector, the D input of the first flip-flop is connected to the delayed data (refer to Figure 4.15). During the preamble, the delayed data consists of alternating ones and zeros. Assuming that the phase error between the delayed data and the clock is small enough for the flip-flop to read the delayed data correctly, the Q output of the first flip-flop also consists of alternating ones and zeros.

Consider now connecting the D input of the first flip-flop to its own \( \bar{Q} \) output during acquisition. Such a modification can be implemented with a digital multiplexer, as shown in Figure B.1. Since the Q output of flip-flop U4 still consists of alternating
ones and zeros, the operation of the phase detector is not altered – at least for phase errors less than $\pm \pi$ radians. Figures B.2 – B.5 are the timing diagrams of this circuit for phase errors of 0, $\pi/2$, $3\pi/2$, and $2\pi$ radians. As with the standard triwave phase detector, the average output of the phase detector is zero when the phase error is zero and is positive when the phase error is $\pi/2$. A more interesting result is that the average output of the phase detector is now also positive for a phase error of $3\pi/2$ radians. (The average output of the standard triwave phase detector is negative for a phase error of $3\pi/2$ radians.) Instead of the “sawtooth” characteristic of Figure 4.12, the phase detector now has a “triangular” transfer characteristic, which is presented in Figure B.6.
Figure B.2: Timing Diagram: Phase Error = 0 radians

Figure B.3: Timing Diagram: Phase Error = π/2 radians
Figure B.4: Timing Diagram: Phase Error $= 3\pi/2$ radians

Figure B.5: Timing Diagram: Phase Error $= 2\pi$ radians
Figure B.6: Transfer Characteristic of Modified Phase Detector During Acquisition

B.2 Metastability

At first glance, the modification to the triwave phase detector may seem to have exacerbated the problem of metastability. As indicated in Figure B.6, the null at $2\pi$ radians is a point of unstable equilibrium. Because of the finite negative slope at this point, the loop is more likely to dwell in the vicinity of this null than it was to dwell in the vicinity of the unstable null of the standard triwave phase detector.

It is important to realize, however, that initial phase errors larger than $\pm \pi$ radians can be avoided if the state of flip-flop U4 is initialized properly. As shown in the timing diagram of Figure B.5, when the phase error equals $2\pi$ radians, the delayed data and the clock are optimally aligned, but the state of flip-flop U4 is incorrect in the sense that it is the opposite of what it would be if the D input of the flip-flop were connected to the delayed data. This situation can be avoided if the D input of U4 is connected to the $\overline{Q}$ output of U4 after at least one bit of the delayed data has been loaded into U4. Provided that this precaution is taken, the initial phase error is guaranteed to lie between $-\pi$ and $\pi$ radians, and acquisition will occur rapidly. Sequential logic can be used to control the multiplexer in the appropriate manner. As a final note, such logic must also switch the D input of the flip-flop back to the delayed data before the end of the preamble, as the modification described in this appendix only works when the delayed data consists of alternating ones and zeros.
Bibliography


163

