ANALOG/MIXED-SIGNAL IP DESIGN FLOW FOR SOC APPLICATIONS

By

Marwa Mukhtar Hamour

BEng, Imperial College of Science, Technology & Medicine 2000

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

The Faculty of Graduate Studies
Department of Electrical and Computer Engineering

We accept this thesis as conforming to the required standard:

The University of British Columbia

November 2003

© Marwa Mukhtar Hamour, 2003
Library Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Marwa Mukhtar Hamour
Name of Author (please print)

1\textsuperscript{st} December 2003
Date

Title of Thesis: Analog/Mixed-Signal IP Design Flow for SoC Applications

Degree: Master of Applied Science
Year: 2003
Abstract

System-on-chip (SoC) with reuse of intellectual property (IP) is gaining acceptance as the preferred style for integrated circuit (IC) designs. Increasing demand for analog/mixed-signal (AMS) cores on SoCs is creating a need for new design methodologies and tools that facilitate the creation and integration of reusable AMS IP. In this work, a practical definition of AMS IP and an associated design-for-reuse process is proposed.

This thesis begins by investigating the issues of reusable AMS IP using a phase-locked loop (PLL) as the design vehicle. In the first pass, a PLL is designed without reusable IP. In the process of designing this PLL, the issues of how the IP should be defined for improved reusability and ease of design is addressed. Then a suitable design methodology is proposed to use the IP in a design flow. Firm IP is suggested as the most appropriate format to deliver the AMS IP library components. Unlike hard IP, this form allows ease of migration of IP from foundry-to-foundry, customer-to-customer and application-to-application. Then, a design methodology is developed for constraint-driven top-down design and model-driven bottom-up characterization of the cores. Once the design topology is selected, an IP hardening flow is described to produce a physical layout.

To validate the approach, the primary specifications of the first PLL were then changed and a second architecture was generated using the proposed design methodology. The time needed to design the second PLL is shown to be greatly reduced (by a factor of four) for comparable performance.
# Table of Contents

Abstract ................................................................................................................................. ii  
Table of Contents ................................................................................................................... iii  
List of Figures ........................................................................................................................... v  
List of Tables ........................................................................................................................... vii  
Acknowledgements ................................................................................................................ viii  

Chapter 1 ................................................................................................................................ 1  
INTRODUCTION ....................................................................................................................... 1  
1.1 Motivation ............................................................................................................................ 1  
1.2 Types of IP blocks .................................................................................................................. 3  
1.3 Applications of Reusable Analog IP ....................................................................................... 5  
1.4 Research Goals ...................................................................................................................... 7  
1.5 Thesis organization ............................................................................................................... 8  

Chapter 2 ................................................................................................................................ 9  
AMS DESIGN, SYNTHESIS AND REUSE .............................................................................. 9  
2.1 Traditional AMS Design Approach ....................................................................................... 9  
2.2 Proposed AMS Block for Reuse ........................................................................................... 12  
2.3 PLL Design Basics ............................................................................................................... 14  
2.4 Analog/Mixed-Signal Design for Reuse .............................................................................. 17  

Chapter 3 ................................................................................................................................ 25  
REFERENCE PHASE-LOCKED LOOP DESIGN ..................................................................... 25  
3.1 Introduction ........................................................................................................................... 25  
3.3 A Reference PLL Design ...................................................................................................... 26  
3.4 Block Authoring Methodology for Analog IP ...................................................................... 36  

Chapter 4 ................................................................................................................................ 38  
DIFFERENTIAL VCO IP ......................................................................................................... 38  
4.1. Introduction ......................................................................................................................... 38  
4.2. Architecture Analysis ......................................................................................................... 39
List of Figures

Figure 1.1 Mixed-signal in SoC trend ................................................................. 2
Figure 1.2 Different Types of IP Blocks ............................................................ 4
Figure 1.3 Multiple applications .................................................................... 5
Figure 1.4 Technology migration .................................................................... 6
Figure 1.5 Multiple foundries .......................................................................... 6
Figure 1.6 Specification change ....................................................................... 7
Figure 2.1 Traditional analog/mixed-signal design flow .................................. 10
Figure 2.2 Bluetooth system ........................................................................... 12
Figure 2.3 Bluetooth SoC ................................................................................ 13
Figure 2.4 Basic phase-locked loop block diagram ......................................... 14
Figure 2.5 Linear model of a phase-locked loop .............................................. 15
Figure 2.6 Behavioral view of sub-blocks ....................................................... 19
Figure 2.7 Characteristic resulting from analytic view ..................................... 20
Figure 2.8 Library of different components ..................................................... 21
Figure 2.9 Schematic view representation of charge pump ............................ 21
Figure 2.10 Matching constraints associated with the schematic .................... 22
Figure 2.11 Test bench presented to the designer .......................................... 23
Figure 2.12 Proposed analog/mixed-signal IP hardening flow ....................... 24
Figure 3.1 Charge pump PLL ......................................................................... 27
Figure 3.2 Current-starved delay cell .............................................................. 29
Figure 3.3 Current starved oscillator characteristic with varying W .............. 30
Figure 3.4 PFD architecture .......................................................................... 31
Figure 3.5 PD Operating frequency with reset time ....................................... 32
Figure 3.6 CP schematic ............................................................................... 33
Figure 3.7 Simple filter .................................................................................. 34
Figure 3.8 Ripple suppressed filter .................................................................. 34
Figure 3.9 Second order PLL magnitude response ....................................... 34
Figure 3.10 Control voltage transient response to a 26MHz input ................ 35
Figure 3.11 Input reference and PLL output signals ....................................... 36
Figure 3.12 AMS system design process ....................................................... 37
Figure 4.1 Single delay stage of VCO architecture ......................................... 40
Figure 4.2 Centre frequency variations with number of stages .................. 42
Figure 4.3 Centre frequency variations with number of stages for a different W.L 43
Figure 4.4 Frequency Characteristic for different number of stages ............ 44
Figure 4.5 Frequency space for 11 stage differential architecture for a range of W 45
Figure 4.6 Deterministic jitter test bench ....................................................... 46
Figure 4.7 Simulated phase noise characteristics for 11 stage VCO ............... 47
Figure 4.8 PSRR test bench ........................................................................... 48
Figure 4.9 VCO behavioral view .................................................................... 49
Figure 4.10 Jitter optimization goal ................................................................. 50
Figure 4.11 Optimizer output ......................................................................... 51
Figure 4.12 Physical view constraints ............................................................. 52
Figure 4.13 Placement of single delay cell ...................................................... 52
# List of Tables

Table 2-1 Typical PLL Specifications.................................................................................................. 16  
Table 3-1 VCO jitter values for different centre frequencies................................................................. 31  
Table 4-1 Power supply rejection ratio calculation.................................................................................. 48  
Table 4-2 Fabricated phase noise values ............................................................................................... 56  
Table 4-3 PSRR calculated values for fabricated architecture................................................................. 56  
Table 5-1 Modified PFD logical table ..................................................................................................... 60  
Table B-1 Oscillation frequency and delay for inverter oscillator......................................................... 91  
Table B-2 Frequency characteristic calculation and comparison.......................................................... 96  
Table B-3 Current starved configuration delay characteristic............................................................... 96
Acknowledgements

Firstly, my thanks are extended to my supervisor, Dr Res Saleh, for his constant support and valuable expertise. I would also like to thank Dr Shahriar Mirabbasi and Dr Andre Ivanov for their help and for their participation on the examining committee.

Throughout my time in the SoC lab, I have had the distinct pleasure of working with a great set of people, who provided constant technical and moral support; special thanks in that category go to Roberto, Roozbeh and Sandy. This would not be complete without mentioning Roopa, Ihab, Fatih, Muna, Haytham, Sarah, and Abdul, who made sure I got through this in one piece.

I would like to dedicate this work to my family, who have been extraordinarily supportive throughout my whole life. Without their encouragement, I would not be where I am today.

This research is supported by PMC-Sierra, Canadian Microelectronic Corporation, and Advanced Systems Institute.
1 INTRODUCTION

1.1 Motivation

The increasing capacity for integration due to advances in submicron CMOS integrated circuit (IC) fabrication has led to an era of complete Systems-on-Chip (SoC) designs. In this emerging area, complex digital blocks and embedded memory dominate a large percentage of the chip area [5]. Less widely recognized in all the publicity over "going digital" is that analog and mixed-signal (AMS) functions have become more critical than ever. In most leading applications, high-performance AMS functions and digital blocks need to be integrated together on a single chip in order to fully utilize the levels of integration made possible by deep submicron (DSM) technology.

Figure 1.1 shows the trend of analog/mixed-signal design content in SoC's [1]. The proportion of chips containing analog/mixed-signal blocks is expected to exceed 70% by the year 2006. This increase is consistent with the expected growth of the wireless industry over the same period, implying the placement of the AMS circuitry on the same substrate as digital blocks. It is a common belief that the incorporation of analog "cores" into the SoC serves as a competitive advantage, especially in applications such as personal digital assistants (PDAs), wireless local area networks (LAN), etc.

In order to keep pace with rapidly evolving markets, the productivity of both analog and digital designers must be increased. The development process or computer-aided design (CAD)

1 The term mixed-signal refers to designs that include both analog and digital signals. For example, A/D and D/A converters would fall into the mixed-signal category.
methodology employed plays a critical role in the productivity of the designers and the resulting time-to-market\(^2\) (TTM) and time-to-volume\(^3\) (TTV) of the product. This has placed enormous pressure on IC designers and CAD engineers to develop better and more efficient approaches to the design process.

![Mixed Signal Trend](image)

**Figure 1.1** Mixed-signal in SoC trend

To date, digital IP development has had a more accelerated growth in industry as compared with AMS IP due, in part, to the fact that digital blocks have a well-established design flow and associated CAD tools [2]. As a result, digital blocks can be re-used with a higher degree of confidence in the resulting performance. While digital design has a well-defined top-down design methodology, AMS design has traditionally had an ad-hoc, custom design approach but never a generic well-defined one. Furthermore, compared to digital IPs, analog/mixed-signal IPs must provide a greater degree of flexibility in the design parameters and performance characteristics. While the general function of an analog block may be the same in different applications, the design specifications may vary widely between the applications. *For analog, one size does not fit all.*

---

2 Time-to-market is the length of time it takes to get a product from idea to market place.

3 Time-to-volume is the required time needed to drive a product from the prototype stage to the manufacturing stage.
characteristics of the IP are usually a function of the IP circuit architecture and its interaction with the surrounding environment. This makes analog IP design for reuse (DFR) quite difficult, as many designers will attest.

In order to promote and to proliferate the use of AMS IP, a proper definition of reusable analog IP itself must first be established. This definition should encompass the circuit structure and the information needed to successfully reuse the analog IP. Then, a flow for reusable analog IP design and IP hardening must be defined and supported with the associated CAD tools. The purpose of this thesis is to investigate the issues associated with reusable analog IP and provide definitions and flows that advance its use in SoC design.

1.2 Types of IP blocks

The actual form of an IP core can vary depending on the way in which the IP designer/vendor chooses to provide the core to the system designer. There are three definitions by which IP can be identified [3]: soft, firm and hard. These forms are described below:

**Soft IP** are specified by behavioral level descriptions. They are more suitable to digital cores where the description in a hardware description language (HDL) is process-independent and can be synthesized to the gate level. This form has the advantage of flexibility, portability and reusability, with the drawback of not having a guaranteed result since implementation in different processes and applications produces variations in performance.
**Hard IP** have fixed layouts and are highly optimized for a given application in a specific process. They have the advantage of having predictable performance. This, however, comes with additional effort and cost, and that may greatly limit the areas of application.

**Firm IP** are less flexible than soft IP. They have some predefined structure of the circuit in order to make performance more predictable. They also have a large number of parameters available to optimize the design. Firm IP offers a compromise between soft and hard, being more flexible and portable than hard IP, yet more predictable than soft IP.

The relative properties of each form of IP are illustrated in Figure 1.2 [4]. The majority of AMS IP today is available in the form of hard IP. This work provides a framework for the use of firm IP and the corresponding flow needed to generate hard IP from a firm IP representation. This is an important intermediate step toward AMS IP synthesis directly from a soft IP representation.
1.3 Applications of Reusable Analog IP

As mentioned above, most of the AMS IP in use today is in the hard IP form and this tends to limit its application. Before describing the details of firm AMS IP and the design flows, we review a number of possible scenarios in which hard analog/mixed-signal IP may not be appropriate.

1. Different applications (same specifications)

The main purpose of reusable AMS IP is illustrated in Figure 1.3 for three different applications. Ideally, the IP should not have to be modified to suit the three cases. However, the surrounding circuitry frequently changes with different applications. This will lead to a change in the resulting performance of the design and should be accounted for in the initial design of the core. Hard IP is suitable only if the specification and environment are identical.

![Figure 1.3 Multiple applications](image)

2. Process scaling/migration

When a new CMOS technology is developed, any pre-existing hard IP in the previous technology node must be migrated to the new technology. Figure 1.4 shows the technology characteristic for the past 8 years and the expected trend for the next 6 years [5]. This presents the designer with a turnaround time of about 3 years before the next generation of the design is expected. As an example, a large proportion of verified AMS IP are available in 180nm technology today but they
must be remapped to 130nm technology. Rapid migration would be facilitated if the IP was in a reusable firm IP form. In fact, this problem will occur again between 130nm and 90nm in the near future.

3. Multiple Foundries

Different foundries have different specifications related to a given process technology; these may lead to different design characteristics and performance. In order to serve the largest possible customer base, the IP should be easily re-targeted for each foundry. This should be possible regardless of the foundry as illustrated in Figure 1.5. Under certain circumstances, the design house may select the foundry based on the available IP or the IP vendor based on the foundry supported.
4. Design specification change

Occasionally, a designer is faced with a change in design specifications late in the design process as illustrated in Figure 1.6. However, if a hard IP is used, it would be difficult to accommodate this change. On the other hand, if a firm IP flow is constructed properly, it would only require another design iteration to select the proper parameters to tune the design to meet the new specifications.

Old Spec

AMS IP Specifications I

- Input Range: ±150MHz
- Output Range: ±300MHz
- Static Phase error: ±200 ps
- Output period Jitter: ±100ps

New Spec

AMS IP Specifications II

- Input Range: ±30MHz
- Output Range: ±600MHz
- Static Phase error: ±200 ps
- Output period Jitter: ±100ps

Figure 1.6 Specification change

1.4 Research Goals

There are a number of commonly occurring analog blocks in standard SoC designs such as the analog-to-digital converter (ADC), digital-to-analog converter (DAC), operational amplifiers, phase locked loops (PLLs) and filters. In this work, PLLs are chosen as the vehicle to understand and address issues concerning AMS IP. They are widely used in both digital and analog domains and exhibit both linear and nonlinear behavior. PLL blocks can be found in SoCs for wireless applications such as the Bluetooth and 802.11b standards. They can also be found in interface designs such as Ethernet applications. They are standard components for clock and data recovery, system synchronization and clock generation. Because of their widespread use, they provide a suitable candidate for our investigation.
The goals of this work can be summarized as follows

1. Establish a definition for firm AMS IP to provide flexibility for reuse while retaining the required characteristics of the design
2. Develop an analog IP for a PLL and demonstrate its reusability
3. Develop a suitable AMS flow for IP hardening (firm IP to hard IP) for SoC applications

1.5 Thesis organization

Chapter 2 of the thesis provides an overview of the state of AMS design for reuse. It introduces the traditional design flow and the different directions being pursued to improve productivity in the design cycle both commercially and academically. It also presents the proposed design flow for analog/mixed signal blocks and the format for its deliverables. In Chapter 3, the basics of PLL theory and design are introduced together with the specifics of the design of the first PLL architecture. Chapter 4 follows the IP library entry creation for a selected differential VCO architecture. This design is taken from concept through to silicon; it also presents a comparison of simulation results and actual behavior. In Chapter 5 the concept of reusability of the PLL as a system and the creation of library entries for its different sub-blocks is presented. This includes the measures that need to be taken to facilitate the reusability of each block. Chapter 6 provides the specifics of the introduced reusable IP design flow for a second PLL design and reports on its simulation results. Finally Chapter 7 summarizes the results and possible future direction for this work.
2 AMS DESIGN, SYNTHESIS AND REUSE

2.1 Traditional AMS Design Approach

Figure 2.1 shows the traditional flow for AMS design; it relies heavily on the expertise and experience of the designer. The design process begins with a performance specification of the component for a target application. System level design is performed to find an architecture that implements the required functionality while meeting specifications at minimum cost. System level designers traditionally use tools such as MATLAB. These tools allow the designers to explore various algorithms and evaluate tradeoffs.

Analog/mixed-signal hardware description languages (AMS-HDL) are used to model the circuit. These languages are a relatively new addition to the design process, and the most commonly used are Verilog-AMS [7] and VHDL-AMS [8]. Automatic generation of AMS architectures from AMS-HDL is still in the infancy stage because of the large number of variables associated with AMS design. The current contribution of these AMS-HDL to system level design is highly significant. They provide the necessary platform for system-level verification, an important part of design quality. Verification of SoCs requires co-simulation of behavioral models and transistor-level circuits to reduce simulation costs [9].
The next step is to map the design to an architecture (topology selection) such that the blocks at the system level represent the correspondingly mapped circuit cells. During topology selection, transistor sizing and layout, the time taken for the design is directly correlated with the experience of the designer. At this stage of the design, the system is usually subdivided into different blocks with the specifications for each block being derived from the overall system level specifications.
The specifications at the system level are propagated to the blocks as constraints for their detailed schematic design. Each block is then implemented using circuit elements such as transistors, capacitors, resistors, etc. Proper modeling of the interfaces between the different blocks is important in the design process to account for different effects such as loading and coupling. This is needed to achieve correct performance for the overall system-level design.

The performance specification of each block is mapped to specifications of lower level blocks until the basic cell-level sizing parameters are reached. Once the transistor-level circuits are created, transistor sizing is needed. The circuit is then continually tweaked and re-simulated for optimal performance. This is the design stage in which most of the designer's effort is concentrated. The process of design optimization is time-consuming because of the nonlinear relationships between the different specifications and transistor sizes. Reducing the time needed to size the circuit can greatly reduce the overall design time.

Research into automation of the sizing stage of the design flow has produced a variety of algorithms. Earlier approaches employed were more topology specific [10]-[12] while, more recently, a number of generalized tools [13],[14] have been developed. There are two main approaches to transistor sizing: knowledge-based and optimization-based. In knowledge-based sizing, detailed analog design knowledge (such as topological and analytical knowledge, rules of thumb, heuristics and simplified models) is used to perform sizing. In the optimization-based approach the design problem is formulated as a mathematical one aiming to determine the solution by minimizing with respect to some cost function. During optimization, the design parameters are updated using either deterministic or statistical updating. Circuit performance is then evaluated using either symbolic equations [15] or a circuit simulator.
The final stage of the design cycle is the layout of the circuit schematic produced by the front-end of the flow in Figure 2.1. The layout of the circuit can either be done manually or automated. For automatic layout, there are two levels the generation procedure; cell layout and system layout. There are two approaches to layout which are the same as in sizing, knowledge-based and optimization-based. For custom high-performance designs, manual layout is more frequently used. The layout is based on a floorplan that has proved to be successful in a prior design.

AMS design automation has been extensively researched [2], [16]-[32]. For brief descriptions of the different approaches developed for both analog synthesis and layout, refer to Appendix A.

2.2 Proposed AMS Block for Reuse

For the selection of an analog block for reuse, we focus on the wireless communication area as it is expected to dominate in the years ahead. An example of a wireless SoC application is the Bluetooth standard. This is an emerging short range (10m-100m) wireless technology that has been receiving its due share of attention since its debut in 1998. It operates in the unlicensed ISM band at 2.4GHz employing a frequency-hopping modulation scheme. To minimize interference with other devices in the ISM band, Bluetooth uses a pseudo-random hopping sequence of 79 hop frequencies, each separated by 1MHz. The gross data rate is 1 Mbps, although overhead usually limits it at 721 Kbps. Bluetooth is designed for low power, dynamic configurability, and support for both voice and data communications. The system consists of a radio unit, a link control unit and a support unit for link management and host terminal interface illustrated in Figure 2.2.

![Bluetooth System Diagram](image)
A number of papers have been published in different areas of its design process and different implementation [34]-[37]. There are already Bluetooth compatible products developed, some one-chip solution and a number of two-chip solutions. The two-chip solution consisting of an RF front end and a baseband processor is the more common solution at the moment because of mixed-signal integration issues at high frequencies [38].

Bluetooth lends itself to implementation using SoC design methodologies, and is expected to appear on many hand-held and portable devices, as well as wired devices. However, the design of a mixed-signal SoC such as Bluetooth is an enormous undertaking. As a data point for the design effort involved, a single-chip Bluetooth transceiver required 30 designers over 2.5 years to complete, that is, 75 person-years of effort [39]. Recently, SoC platforms for the baseband portion of Bluetooth have been designed with reusable IP blocks [40], [41] to greatly reduce the overall development time. However, the main design bottleneck continues to be the AMS portions of the design.

Figure 2.3 shows the Bluetooth baseband SoC, which is traditionally thought of as the digital part of the system. As can be seen from the figure, the chip is approximately 90% digital but there is a 10%
component that is analog/mixed-signal. The ADC and DAC are needed for communication with the RF portion and the PLL can be used to generate multiple clocks or to de-skew the clock. Although the analog blocks may occupy only a small part of the area in these mixed-signal SoCs, they typically require a large part of the design time and cost, and are often responsible for design errors and expensive re-design iterations [26]. Ideally these should be implemented as reusable IP to reduce the design times required but at the moment, most AMS IP vendors provide hard IP for these three components.

All three AMS blocks are of equal importance and complexity and are suitable as IP blocks for our investigation. To investigate the feasibility of the design of reusable AMS IP, the PLL was chosen as the design vehicle. Although extensive research has gone into the design automation of PLLs, comparatively less work has centred on design for reusability of PLLs. We now explore this topic in more detail after a brief overview of PLL design.

### 2.3 PLL Design Basics

PLLs are widely used circuits in modern electronic systems. The PLL IC is a highly integrated mixed-signal circuit that synchronizes the frequency generated by an oscillator with the frequency of a reference signal. As shown in Figure 2.4, a basic PLL contains three major blocks, a phase-detector (PD) a loop-filter (LF), and a voltage-controlled oscillator (VCO).

![Figure 2.4 Basic phase-locked loop block diagram](image-url)
The PD compares the phase of the input signal, $x(t)$, against the phase of the VCO output signal, $y(t)$. The output of the PD is typically a voltage proportional to the phase difference between its two inputs. The difference voltage at the PD output is filtered by a low-pass loop filter. The smoothed output of the LF is then applied to the VCO as the control voltage. This control voltage changes the frequency of the VCO in a direction that reduces the phase difference between the input and output signals. When the loop is locked, the control voltage is such that the frequency of the VCO is equal to the average frequency of the input signal.

Figure 2.5 shows the basic PLL block diagram with the linear transfer function of each block. The linearity assumption is valid for systems in the locked or nearly-locked condition. The input signal has a phase $\theta_i$ and the VCO output has a phase $\theta_o$. $K_{PD}$ is the PD gain factor, $F(s)$ is the LF transfer function, and $K_{VCO}$ the VCO gain factor.

![Figure 2.5 Linear model of a phase-locked loop](image)

The closed loop transfer function of the PLL is given by

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_{PD}K_{VCO}F(s)}{s + K_{PD}K_{VCO}F(s)}$$

(2.1)
There are several figures of merit on which the PLL performance is measured. In this work, specifications for lock time, period jitter, static timing error and power are used in the design of PLLs. These are defined in Table 2-1.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lock Time</strong></td>
<td>This is a measure of the settling time of the PLL once a change in frequency has been initiated. For this measurement, the frequency step and accuracy must be predefined. In this work, the lock time for a PLL is defined for the PLL initiated to its centre frequency to when the output frequency is within 5% of the input reference signal.</td>
</tr>
<tr>
<td><strong>Jitter/Phase Noise</strong></td>
<td>Each block within the PLL contributes to noise within the loop. There are two types of noise, intrinsic noise and deterministic noise. Intrinsic noise consists of <em>shot noise, thermal noise</em> and <em>flicker noise</em>. Deterministic jitter is due to noise in the supply and/or at the input. For this work, periodic jitter due to both types of jitter is used as a measure of the PLL's performance. This is the deviation of output signal's period from the ideal reference period.</td>
</tr>
<tr>
<td><strong>Static Phase Error</strong></td>
<td>Once the PLL is locked, there is an allowable static phase/timing difference between the input reference signal and the output of the PLL. A well designed loop will aim to minimize this error.</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>The target application family is for wireless functions, and therefore the power dissipation of the IP needs to be minimized.</td>
</tr>
</tbody>
</table>

Table 2-1 Typical PLL Specifications
2.4 Analog/Mixed-Signal Design for Reuse

This section presents design for reuse (DFR) concepts and defines the form and characteristics of a reusable AMS core. The purpose of DFR is to carry out a given design with the sole objective of providing enough supporting information so that the derivate designs can be quickly and easily created for other applications. As described in Chapter 1, usually the other applications involve another system environment and/or in a different fabrication process, with either the same or slightly modified performance requirements. The degree of reusability of a core can be measured by the amount by which the design cycle is shortened from the base design to derivative design.

Currently, because of the complexity of analog/mixed-signal design and its sensitivity to the surrounding environment, it is most commonly presented in the form of hard IP. However, for the reasons described in Chapter 1, this form has limited scope of applications. Hard IP will reduce the design cycle greatly when the specifications and fabrications processes are identical, but will hardly improve the design cycle if it is required to be migrated to new processes. This calls for a more flexible definition for the format in which the AMS IPs are provided.

The most suitable IP format at the current time is firm IP. This is not a new suggestion as it has been proposed elsewhere [4]. However, the details of this form of IP have not been fully elaborated to identify the advantages and disadvantages of this form of IP delivery, and the associated IP hardening flows have not been fully developed. Our aim is to use the PLL to explore these issues relative to firm IP.

When developing reusable AMS cores, the usual precepts of a good design must be followed. That is, there should be a good formal specification, a good architecture and a good implementation of
the design. In addition, there are a number of additional steps that must be followed to achieve reusability, as discussed below. A successful IP block should be parameterized, easily verified through reusable test benches, well-documented, and have associated views to ease the derivative design process. These design strategies should be followed at all times during design. We now describe the models, specifications, views and test benches in more detail.

1. Development of Parameterized Models

Specification is an important part of the design; this will characterize the intended design space for the user in terms of both its external and internal behavior. The same as for digital designs [29], analog specification requirements need to be addressed during design. The specifications of the design should have a detailed description of the following:

- Functionality of the circuit
- Performance range specifications
- External interface to other hardware
- Physical design issues such as area and power.

When developing the different models associated with each core, a few things should be kept in mind. The functionality should be modeled in a generic, parameterized way allowing the model to cover a wide range of design space and the underlying lower level implementations. When a certain specification is to be evaluated, all circuit aspects that influence this specification should be included in the model. In general, a tradeoff can be made between the accuracy of the model and the necessary evaluation time. In the preliminary stages of the design, a top-down implementation is employed where less accurate models can be allowed in order to calculate a rough estimate of the design space on which further analysis can be performed to refine the results.
2. Behavioral/Analytical view

This is a description of the different blocks of the design in AMS-HDL. It is a behavioral/analytical model of the circuit used as an initial base for the analysis and design of the circuit. This model can also be further refined to reflect non-idealities of the circuit for a more accurate representation of the circuit. This behavioral/analytical model is also used to co-simulate the AMS design with the digital parts of the SoC during both the creation and verification stages of the design process [42]. A survey of the methods for behavioral modeling of analog circuits is presented in [43]. Behavioral simulation specific to phase-locked and delay-locked loop architectures are provided in [44]-[46].

The guidelines for good coding in HDL for digital design can be found in [29], and these can also be extended to analog/mixed-signal designs. It is important to establish a good choice for partitioning in the design. In the example PLL architecture, these partitions are at the natural boundaries of the design that are associated with the different sub-blocks of the circuit, the phase frequency detector (PFD), loop filter, VCO and divider as illustrated in Figure 2.6.

![Figure 2.6 Behavioral view of sub-blocks](image-url)

The analytical view is a set of design equations associated with individual sub-circuits at a higher level than the transistor level but representative of the key relationships at the transistor level. This bottom-up view is generated directly from the transistor level whereas the behavioral view is a top-
down view. For example, the entire PLL can be represented at the behavioral level while each of the blocks in Figure 2.4 would have its own analytical view. It takes into consideration some of the non-idealities associated with the design and can be used for block selection. This analytic representation can be constructed in many different forms depending on the modeling effort required or the simulation tools available. In this work, some models are developed by hand and other are the result of simulations in MATLAB and Spectre.

Behavioral simulation was performed using MATLAB. The ideal behavioral representation of the PLL and its sub-blocks are readily modeled with this tool; all that is required is the interface definitions and the addition of any non-idealities that may arise such as extra capacitance from wiring in the final layout stage. Figure 2.7 shows the two frequency characteristics for a 3-stage VCO, one in which an estimate of the parasitic capacitance resulting from the layout is taken into consideration and another model where it is not.

![Figure 2.7 Characteristic resulting from analytic view](image)

The behavioral view together with the analytical view is generated by the IP developer. For the end user, they are the means by which a choice of configuration/topology is made from the library (Figure 2.8) of different architectures using the specifications for the design.
### 3. Schematic view

This is a transistor level representation of the different sub-blocks within the system. A default list of transistor sizes and the resulting performance parameters of the circuit is included with the schematic view and is used as an initial data sheet for the design as shown in Figure 2.9. When characterizing the schematic view of the circuit, care must be taken to include the possible ranges of the transistor sizes and element values. However, the specifics of the transistor sizes for the design are viewed as a restriction that should be included with the test bench view related to the schematic rather than with the schematic view.

![Figure 2.8 Library of different components](image)

![Figure 2.9 Schematic view representation of charge pump](image)
4. **Physical view**

This view presents the optimization driven layout constraints tied to the schematic view of the circuit. This view presents the designer with the constraints related to the parasitics associated with the design. For detailed analog layout methodologies refer to [50]. Some of the constraints associated with analog layout are briefly summarized in Chapter 5 under the section describing charge pump design. A representative floorplan is required to convey placement and proximity constraints for the design.

![Matching Constraints](image)

**Figure 2.10** Matching constraints associated with the schematic

The matching constraints to the layout of the system can be incorporated into the netlist as shown in Figure 2.10. The use of symmetry and common-centroid layout approaches can be used to implemented the specified matching constraints.

5. **Test benches**

The test benches are needed to validate the performance of the circuit under different design and process variations. They are used as the basis for verification of specifications and for exploration of the design space for the system. In the Cadence tools, the test benches are constructed in Analog
Artist and the calculator tool is used to evaluate the outputs of the different test scenarios to which the circuit under test (CUT) is subjected (Figure 2.11).

The optimizer tool employed simulates the CUT and in conjunction with the calculator functions explores the design space for the best fit solutions. A number of different test benches may be required for different specifications of the circuit.

![TEST SUITE](image)

**Figure 2.11** Test bench presented to the designer

6. **IP Hardening Flow**

A firm IP hardening flow is illustrated in Figure 2.12 to generate the GDSII layout of the design. The starting point of the flow is the set of selected library components that comprise the unoptimized schematic view of the design. The parameters of the models are set by an optimization tool to achieve the derivative design specifications. The final design can be selected by a tool or by
the user. The physical design is carried out using the information in the physical view of the IP to produce the GDSII layout.

Once a specifically sized architecture is chosen, the firm IP is taken through the above hardening flow for optimization of the circuit to maximize performance and to generate the GDSII layout of the core.
3 REFERENCE PHASE-LOCKED LOOP

DESIGN

3.1 Introduction

This chapter describes the design of a practical integrated circuit phase-locked loop to further develop the concepts of reusable IP design. The overall strategy is to first select the appropriate topology for the PLL based on the design specifications, and then optimize a given set of parameters for each of the associated blocks to deliver the desired specifications. When designing a reusable PLL, it is important to consider the types of changes that are likely to occur when going from one version of the design to the next. The range of possibilities must be captured in the parameterized blocks available in the IP component library.

The changes in PLL specifications and performance metrics can be subdivided into two categories: primary changes and secondary changes. In the primary change category are the fundamental specifications of the PLL which lead to the selection of the appropriate architecture based on the blocks from the library. Secondary changes are considered as more of an optimization of the blocks selected. This category includes changes that do not require architectural modifications but rather another iteration of the optimization procedure. For a given IP library, it must be possible to quickly assess whether a set of design specifications can be satisfied by the elements available in that library. If so, the proper architecture must be selected and then optimized. If not, then new elements must be added to the library before the design can proceed.
The goal of this chapter is to describe the PLL design process and understand the requirements of reusable library components. In the next chapter, the voltage-controlled oscillator is used to illustrate the steps needed to develop a reusable library component. Additional library components are described in Chapter 5. A complete PLL design based on a set of library components is described in Chapter 6.

### 3.3 A Reference PLL Design

The design of various PLL architectures and their optimization has been extensively covered in [51], [53] with special emphasis on design for improved noise performance introduced in [54], [56]. When designing a PLL, performance and stability considerations must be accounted for in the design procedure. Power is an increasingly important consideration in design [57] for which models have been developed [59]. The entire procedure for a fully integrated PLL is generally an iterative process. Typically, design parameters are propagated from the mathematical model to the system level model to the transistor level design. Here, we illustrate a constraint-driven top-down approach along with a model-driven bottom-up approach to design the PLL.

The first step in the process is to establish the design specifications. The target application for our PLL is the Bluetooth baseband processor running at a frequency of 26MHz. Since actual design specifications were not readily available, design constraints were assigned as follows:

- Centre frequency = 26MHz
- Locking time ≤ 3us
- Static phase error < 0.01ns
- RMS jitter < 300ps
- Power dissipation < 1mW
From a top-down perspective, these specifications can be translated into constraints that will affect the design of each of the sub-blocks. In particular, the centre frequency, RMS jitter and power dissipation will affect the VCO design. The locking time and stability requirements will affect the filter design. The static phase error constraint will be propagated to the PD and CP design. The task at hand is to construct a suitable architecture of the design needed to achieve these specifications. We now describe how these constraints would be used to design such a PLL using a well-known architecture, and show how the bottom-up models would be used to establish the feasibility of the design.

The most commonly used IC PLL architecture is the charge pump PLL. The charge pump PLL is a digital PLL that uses a charge pump as the output of the PFD as shown in Figure 3.1. The PFD compares the input reference signal and the feedback signal to produce two control signals, UP and DOWN. These control signals control how much error current flows into the loop filter.

![Figure 3.1 Charge pump PLL](image)

The PFD allows the PLL to have a pull in range that is only limited by the VCO tuning range [60] with a static phase error of zero between the input reference signal and the feedback signal even if the reference signal is not equal to the centre frequency of the VCO [61]. In an integrated circuit,
switching signals from the digital portions of the circuit can couple into sensitive AMS circuit nodes and directly degrade the overall signal to noise ratio. This coupling can happen through power supply lines as well as substrate. The charge pump PLL architectures display increased immunity to power supply variations [62].

**VCO Block**

The design of the VCO is dependent on both PLL frequency range and jitter/phase noise specifications, and it must also satisfy the power requirements. The VCO is the block that will primarily determine the overall PLL jitter/phase noise performance. There are many different configurations for VCOs, the choice of which is dependent on the specifications of the individual design. Oscillator configurations can be categorized into one of two categories, tuned oscillators and nonlinear oscillators. For PLL applications, the two most common types used are the LC oscillators, part of the tuned oscillator family, and ring oscillators which are members of the nonlinear oscillator family. Both types of oscillators are easily integrated on chip. A ring oscillator is realized by placing an odd number of open-loop inverting amplifiers in a feedback loop, while LC oscillators are defined by the use of on-chip inductance.

For this low-frequency application, ring oscillators are used in the design of the PLL. The choice of the VCO is made dependent on the frequency and tuning range requirements. Ring oscillator architectures have been extensively researched [65], [67], [69]-[74]. Several methods for modeling of the VCO characteristics as well as ways of calculating phase noise have been developed [64], [65]. Assume that we are using a voltage-controlled ring oscillator with each stage implemented using the current starved configuration shown in Figure 3.2. This is simply an inverter with two additional transistors in the pull-up and pull-down paths that are used to adjust the delay through the stage.
The resistance of the two extra transistors is adjusted by the control voltage $V_{\text{control}}$ which, in turn, changes the delay of the gate and therefore the frequency of the ring oscillator. This VCO would have parameters of device width ($W$) and length ($L$), supply voltage ($V_{\text{DD}}$) and number of stages ($N$).

![Figure 3.2 Current-starved delay cell](image)

We require a bottom-up model to characterize the VCO so as to be able to decide whether such an architecture can satisfy the top-down constraints. This model may be in the form of equations, tables or a set of simulation results. Taking the centre frequency of the PLL as an example, this constrains the operating frequency at which the VCO is biased. There are two parameters that will affect this frequency for a specific architecture, $N$ which is the number of stages and $W$, the widths of the transistors and $L$ the length of the transistor channel. For example, the parameter $N$, the number of stages, can be computed from the equation relating oscillation frequency to the specific delay [67]:

$$f = \frac{1}{2N\tau}$$  \hspace{1cm} (3.1)

The stage delay $\tau$ is a function of the control voltage and the parameters $W$ and $L$. For example, if the delay per stage $\tau = 370\text{ps}$ the ring oscillator we require 47 stages to achieve the target frequency
of 26MHz. If we fix the parameter, \( N \) the number of stages in the VCO and then we sweep \( W \), the width of the transistors, and plot the frequency vs. control voltage, we obtain a region of the feasible solution for the VCO as shown in Figure 3.3. These characteristics capture the VCO characteristics for a range of \( W \) values. They can be converted to analytical equations (using curve fitting) or generated using detailed equations based on the VCO schematic. As an illustration of the process, the detailed model equations for this case are provided in Appendix B.

![Figure 3.3 Current starved oscillator characteristic with varying \( W \)](image.png)

For our target centre frequency, \( f_{\text{cm}} = 26 \text{MHz} \), the solid line in the figure is the characteristic for the oscillator that meets this specification. The exact parameters of the design are \( L = 1 \mu m \) and

\[
W_1 = 2.5 \ W_4 = 5 \mu m, \\
W_2 = 2.5 \ W_1, \\
500 \text{nm} \leq W_3 \leq 40 \mu m.
\]
A similar model could be constructed for the RMS jitter for the current-starved configuration. This characterization for the jitter can only account for the intrinsic jitter and not the deterministic jitter. Alternatively, a more practical approach is through quotation of the typical values for total jitter for a number of centre frequencies for the VCO architecture as shown in Table 3-1. This gives the jitter minimum value for the architecture, since the jitter of the PLL is usually greater than the jitter of the VCO alone. To calculate the jitter values for the chosen architecture, Spectre simulations are run to generate the phase noise plot of the architecture. These phase noise values are then used to calculate the time domain jitter values for the VCO by integration.

<table>
<thead>
<tr>
<th>Centre frequency (MHz)</th>
<th>RMS Jitter (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>187</td>
</tr>
<tr>
<td>250</td>
<td>360</td>
</tr>
</tbody>
</table>

Table 3-1 VCO jitter values for different centre frequencies

**PD Block**

For the PD block, the PFD architecture shown in Figure 3.4 is used.

![Figure 3.4 PFD architecture](image-url)
This implementation requires a delay buffer in the reset path of the architecture to eliminate operation of the PFD in the dead zone. Dead zone occurs when the phase difference between the reference and feedback clock is minimal, such that the charge pump does not have time to turn on and correct the phase difference.

The delay buffer/reset time while eliminating dead zone places a constraint on the maximum operating frequency of the PFD, \( f_{\text{max}} = \frac{1}{2T_{\text{mer}}} \). This is the frequency design constraint set by the PLL specifications. This constraint is a function of the PD parameters, \( W \), width of the transistors, \( L \) the length, and \( N \), the number of stages in the delay buffer

\[
\tau_{\text{reset}} = C + \sum_{n=1}^{N} \left( \frac{a}{W_n} [bW_{n+1} + cW_n] \right)
\]

where \( a, b, c \) and \( C \) are dependent on the transistor dimensions.

The maximum frequency characteristic for a range of \( \tau_{\text{reset}} \) is shown in Figure 3.5. The delay through the PD and its power dissipation can not exceed the phase error and power specifications.

![Figure 3.5 PD Operating frequency with reset time](image-url)
CP Block

The CP block, to ensure correct biasing, is mostly custom designed. However its characterization is still possible using circuit analysis. With this block, the current source forms a large part of the power dissipation of the overall system, therefore for power constrained PLLs, the value of $I_{cp}$ is chosen carefully. For our example, the simple switch in source charge pump architecture, shown in Figure 3.6, is used.

The power specification of the PLL would bound the bias current as follows:

$$P > V_{DD}I_{p}$$

$$I_{p} = f(W, L)$$

![Figure 3.6 CP schematic](image)

The switching time of the CP will affect the speed of the circuit which affects the lock time though not significantly when compared to the filter effects on the lock time.

Loop Filter

The loop filter and the loop bandwidth affect the stability, transient response and lock time of the circuit. A simple implementation of the filter is shown in Figure 3.7. However, interaction between
the charge pump and the filter causes a ripple on the VCO control voltage, a parallel capacitor, $C_2$, is added to suppress it as shown in Figure 3.8.

![Figure 3.7 Simple filter](image)

![Figure 3.8 Ripple suppressed filter](image)

This extra capacitor adds a pole to the PLL transfer function making it a third order system. However if $C_2 < C_1/20$, the system can be approximated as a second order system [63]. The magnitude response for the system is shown in Figure 3.9.

![Magnitude response](image)

The zero of the system is: $\omega_z = 1/RC_1$ and the pole is: $\omega_p = (C_1 + C_2)/RC_1C_2$. The loop bandwidth for stability must lie between those two values to ensure sufficient phase margin.

Using the blocks introduced in this Chapter, a PLL was designed and its performance is investigated in the next section.
PLL Performance

The designed PLL has the following performance when simulated in Spectre

- Locking time, $t_{\text{lock}} = 2.5 \mu s$
- Static phase error, $\tau_{\text{err}} = 0.005 ns$
- RMS jitter, $\sigma = 216.2 ps$
- Power dissipation in lock, $P_{\text{lock}} = 0.5 mW$

The transient control voltage response to an input frequency of 26MHz is shown in Figure 3.10. The spikes in the response are due to the charging and discharging of the filter by the CP. Figure 3.11 shows the input and output signals of the PLL.

![Control voltage transient response to a 26MHz input](image)

*Figure 3.10 Control voltage transient response to a 26MHz input*
3.4 Block Authoring Methodology for Analog IP

With a design of a PLL now complete, the design process described in the previous sections is shown in the flow diagram in Figure 3.12. Specifications are first converted into a PLL architecture with constraints mapped to each component in block diagram of the PLL. For each block, there are a set of library elements that have adjustable parameters. The library elements are fully characterized in their analytical view using equations, tables, and/or simulation results. These elements are selected based on their suitability in delivering the required performance based on the top-down constraints. Once the appropriate blocks are selected, parameter optimization is used to
obtain the final circuit. The resulting parameters must lie within the bounds specified for each of
the blocks for a valid design. This design is now ready for automatic or manual layout.

Figure 3.12 AMS system design process
4 DIFFERENTIAL VCO IP

4.1. Introduction

To facilitate the choice of the best possible implementation for each core, a library of different architectures and their respective specifications should be compiled. Before we choose an oscillator configuration to be used in a PLL from this library, detailed knowledge of its operation and expected performance is needed. This information must be conveyed to the system designer in the simplest and most efficient format. In this chapter, the steps taken in the development of a reusable VCO core, for the IP library, to facilitate multi-application and/or multi-specification designs is presented.

As presented in Chapter 2, there are five views needed to characterize each core. These are the schematic of the architecture, the behavioral for system-level simulation, the analytical, the layout view, and the test benches. During the development stage of each core, the designer will need to analyze the circuit and develop simple mathematical performance models to be documented as the analytical view of the IP. These are models for the block properties that will affect the system level specifications. Second, is the creation of test benches for the circuit performance. The final part of the IP development is the generation of the physical view for the IP. This will define the layout constraints for the IP to ensure its correct operation when fabricated.
The following sections introduce the primary specifications for the VCO block that are needed to define its design for reusability, and its development as an IP core. In particular, Section 4.2 deals with the validation of a specific architecture, and explains the development of the corresponding behavioral and analytical models that are incorporated into the IP. Section 4.3 outlines the development of the core test benches, and Section 4.4 presents the physical view of the IP. For experimental validation, the core described here was fabricated in 0.18um CMOS technology. A comparison of Spice simulation and experimental measurements is presented in the final section.

4.2. Architecture Analysis

When selecting a particular configuration for a circuit block, the IP developer must understand tradeoffs, limitations, and range of operation. For example, in CMOS technology, ring oscillators are frequently chosen over LC oscillators for low frequency VCO design because of savings in area. In Chapter 3, a PLL employing a single-ended ring oscillator, was presented in the design of the reference PLL. However, in integrated circuits, differential configurations are more widely used for their noise rejection characteristics. For the single-ended ring oscillator, phase noise for a given power dissipation and frequency is independent of the number of stages while for a differential architecture, the phase noise grows with the number of stages [64]. Analysis of the characteristics and phase noise of differential architectures has been extensively documented [65], [71], [72], [74], [75], [77]. The differential architecture does however present an advantage as compared with single-ended architectures when placed near digital circuitry for its rejection of common mode noise, that is, it has a lower sensitivity to power supply and substrate noise [72], analysis of which is presented in [73], [76]. Still, among differential ring oscillators, there are several architectures to choose from, the choice of the specific VCO architecture to be implemented in a PLL design is dependent on the target PLL specifications, especially the frequency and jitter specifications.
When considering suitable architectures for reusable AMS IP, the selected architectures must be robust and easily synthesizable and optimizable. One such architecture is the VCO architecture [69] with the delay stage shown in Figure 4.1. This architecture adjusts the delay with the resistance of a transmission gate. This configuration also has a built in feedback circuit to make the delay insensitive to common-mode voltage variations. The variable resistance of the transmission gate adjusts the delay of each stage.

![Figure 4.1 Single delay stage of VCO architecture](image)

Once such a block is selected for inclusion in the library, it must be thoroughly analyzed. The goal, at this stage of the design, is the analysis of the primary properties of the VCO architecture that will determine the system level performance, as well as its key behavioral characteristics. The key properties considered here are the frequency characteristics, jitter/phase noise, and power supply rejection performance.

### 4.2.1. Frequency Characteristics

This characteristic shows the range of operational frequencies for the VCO as well as the linearity of its response. This is one of the first properties of the architecture that needs to be documented.
Bottom-up models are developed to capture this frequency characteristic for the VCO from simulation results and using mathematical models from circuit analysis [66]-[68]. For this application, numerical modeling was done in MATLAB. The delay characteristics of each cell were modeled carefully so the frequency of oscillation could be derived without the need for transistor level simulation.

The number of stages (which determines the frequency range) in the VCO is bounded by a minimum of 3 required for oscillation and a maximum that is dependent on the parameters W and L. For specific W and L, the parameters of the design, there is a maximum value of N after which the linearity of the transfer characteristic for the VCO degrades. In that case, if a lower target frequency is needed, changing the parameters of the design, W and L, as well as the number of stages is required.

Figure 4.2 shows the variation in the centre frequency (for specified transistor widths and lengths) with change in the number of stages in the VCO, the circles represent the simulation results and the solid line is the model-generated results. The centre frequency is specified as the oscillation frequency resulting from a control voltage of 1.4V, the midpoint of the allowable swing of the control voltage. As expected, the centre frequency of oscillation reduces with increasing number of stages. The rate of change of centre frequency decreases with increasing number of stages, indicating that a minimum value will eventually be reached. Also physically, there is a maximum number of stages after which the penalty of increasing physical size will outweigh the advantages of choosing this configuration over others that may have a lower frequency range.
From the simulation results, the dependence of the centre frequency on the number of stages, $n$, can be captured in a function of the form

$$f(n) = \frac{A}{n} + \frac{B}{n^2} + \frac{C}{n^3} + K$$  \hspace{1cm} (3.3)

The values of the constants $A$, $B$, $C$, and $K$ are dependent on the parameters $W$ and $L$. From analysis of the experimental circuit, and using simulation results, the centre frequency as a function of the number of stages is found to be

$$f(n) = \frac{2.6 \times 10^3}{n} - \frac{0.8 \times 10^3}{n^2} + \frac{1.8 \times 10^3}{n^3} - 0.9$$  \hspace{1cm} (3.4)
To verify the model, a differently sized VCO architecture is analyzed and spice simulation results are compared with those calculated using the model. These results for 3, 9 and 15 stage VCO are shown in Figure 4.3.

![Graph showing centre frequency variations with number of stages for different W/L](image)

**Figure 4.3** Centre frequency variations with number of stages for different W/L

To model the dependence of operational frequency on the parameters of the design and the control voltage, circuit analysis is used. An approximation for the values of $R$ and $C$, for each delay stage where the resistance is variable and the capacitance fixed is derived. This is then used to derive the operational frequency of the architecture as a function of the widths, lengths and control voltage. The exact equations used to derive the capacitance are given in Appendix B. These equations are of the form

$$R = f\{R_{eq}, R_{eqn}, W, L, V_{control}\}$$  \hspace{1cm} (3.5)

$$C = f\{C_J, C_{Jsw}, C_g, W, L\}$$  \hspace{1cm} (3.6)
The delay, $\tau = RC$, is used to calculate the frequency using $f = 1/(2N\tau)$. Sizing of the transistors is such that both high-low and low-high transitional delays are equal. The resistance of the transmission gate is variable depending on the applied control voltage and the load capacitance at the output is assumed constant and independent of the applied control voltage. The fixed part of the resistance is equated to ensure equal delay in the pull-up and pull-down process of the oscillator which means that

$$R_{M5/M6} \parallel R_{Mn} = R_{M1/M2} + R_{MP}$$

where $R_n$ is the resistance of the transistor x.

Figure 4.4 shows the model-generated frequency characteristic for a 3, 5, 13, and 15 stage VCO. This transfer characteristic shows desirable linearity characteristic and the large range for each VCO configuration simulated.
The model is used to generate the frequency characteristic for a range of $W$ for an 11-stage oscillator. The results are shown in Figure 4.5.

![Graph showing frequency vs control voltage for different $W$ values](image)

**Figure 4.5** Frequency space for 11 stage differential architecture for a range of $W$

### 4.2.2 Jitter/Phase noise

The jitter of the VCO constitutes a significant part of the system level PLL jitter. For this specification, a minimum/typical value is provided with the IP. The designer is now able to choose the appropriate architecture to be optimized without needing to run simulations. However, during the actual design process of the VCO, jitter simulation for deterministic phase noise/jitter is required and the designer is provided with Spectre test benches for this measurement.
The test bench setup for supply induced jitter is shown in Figure 4.6. This jitter results from noise in the power supply lines. The assumption that a clean input signal is used has been made. The transient simulation is run for a long period of time, usually about 500 times the period. A set of deviation values is collected for post-processing to calculate the RMS jitter.

In the frequency domain, Spectre analyses are run to generate the phase noise (PN) plot for the architecture. Typically, PN values are given as numerical values at an offset with respect to the centre frequency. The offset at which the phase noise is measured is dependent on the specifications of the particular application for the VCO. For our work, the configuration is simulated in the worst-case corner and the phase noise is plotted for a full range of offsets. The jitter is calculated by integration of the phase noise plot.

![Deterministic jitter test bench](image)

Figure 4.6 Deterministic jitter test bench

The phase noise for an 11-stage VCO based on the differential configuration is shown in Figure 4.7. From the graph, we can extract the two points:

@ 500 KHz offset: $PN \approx -147dBc/Hz$

@ 1MHz offset: $PN \approx -149dBc/Hz$
4.2.3. Power Supply Rejection Ratio

Integration of the PLL into an SoC exposes it to switching noise from digital circuits that will introduce noise on the supply lines. The core's sensitivity to this supply variation is defined as its supply rejection ratio (PSRR). The changes in the power supply will cause the range of the frequency characteristic of the VCO to shift. To measure the PSRR, the VCO central frequency is measured for a nominal supply voltage of 1.8V. The supply is then varied ±10% of the nominal and the central frequency re-measured. The PSRR is given by

$$PSRR = 20 \log \left( \frac{\Delta VDD}{\Delta f_{center}} \right) K_{vco}$$  \hspace{1cm} (3.8)

$K_{vco}$ is the gain of the VCO and the result has units of dB. This equation translates changes on the supply voltage to changes at the input. The test bench for measurement of PSRR is shown in Figure 4.8.
The PSRR results are shown in Table 4-1 for the 11 stage differential VCO architecture. If we consider the range of frequency deviations, the variations are within 5% of the values of the normalized central frequency.

<table>
<thead>
<tr>
<th>Power Supply (V)</th>
<th>Frequency (MHz)</th>
<th>Power Supply Rejection Ratio (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.62</td>
<td>103</td>
<td>43.2</td>
</tr>
<tr>
<td>1.98</td>
<td>107</td>
<td>42.1</td>
</tr>
</tbody>
</table>

To summarize, three main properties of the VCO core need to be documented into the analytical view during the IP development. These are the frequency characteristic, jitter performance, and PSRR. The frequency characteristic is presented in the form of equations relating the frequency of oscillation as a function of the circuit parameters and a varying control voltage. The jitter is given as a minimum value together with the test benches needed for simulation to obtain accurate results.
The PSRR of the architecture is given as a worst case value. This gives the designer an indication of the sensitivity of the circuit to the surrounding environment.

As the PLL is targeted toward integrated applications, co-simulation with digital designs is required. This is made possible by the behavioral view of the IP shown in Figure 4.9. This can be a simple model of the architecture or a more detailed representation of the circuit considering issues such as phase noise and jitter [80]. The target is to allow for full automatic synthesis from this view, the possibility of which was investigated in [79], but which is still not feasible at this time.

```
// Differential quadrature voltage controlled oscillator

module quadVco (Plout, Nlout, PQout, NQout, Pin, Nin);

input Pin, Nin; voltage Pin, Nin;
output Plout, Nlout; voltage Plout, Nlout;
output PQout, NQout; voltage PQout, NQout;

analog begin
  @(initial_step) seed;
  freq = (V(Pin,Nin) - vmin) * (fmax - fmin) / (vmax - vmin) + fmin;
  freq = freq/(1 + dT*freq);
  phase = 2*M_PI*idtmod(freq, 0.0, 1.0, -0.5);
  @(cross(phase - 3*M_PI/4, +1, ttol)
   or cross(phase - M_PI/4, +1, ttol)
   or cross(phase + M_PI/4, +1, ttol)
   or cross(phase + 3*M_PI/4, +1, ttol))
    begin
      dT = 2*jitter*$dist_normal(seed,0,1);
      i = (phase >= -3*M_PI/4) && (phase < M_PI/4);
      q = (phase >= M_PI/4) && (phase < 3*M_PI/4);
    end
  V(Plout) <+ transition (i vh : vl, 0, tt);
  V(Nlout) <+ transition (i vl : vh, 0, tt);
  V(PQout) <+ transition (q vh : vl, 0, tt);
  V(NQout) <+ transition (q vl : vh, 0, tt);
end
endmodule
```

Figure 4.9 VCO behavioral view
4.3 Test Benches

There is a great deal of information gained during the analysis of the circuit. This should be used in the construction of the test benches associated with the IP and is used in its optimization. For our circuit, test benches for three primary properties to be optimized were constructed. These are jitter, power consumption, and the centre frequency. The jitter/phase noise will determine the performance of the PLL while power is a critical issue that must be monitored during design. The frequency test bench is needed to ensure that the oscillator is still operational in the required space.

The test benches should be designed so as to allow the user to tweak the circuit for maximum performance without the need to manually reiterate the sizing and simulation stage of the design procedure. This would mean having variables rather than exact values for every parameter. The goals for the optimization are set by specifying the desired design space and five points within that space. Three goals were set; Centre frequency (maximized), Jitter (minimized), and Power (minimized). The jitter goal setup is shown in Figure 4.10. The centre frequency and jitter goals are given a higher weight than the power goal.

![Figure 4.10 Jitter optimization goal](image-url)
The test benches allow the user to explore the tradeoffs between the different possible output designs. The output of the optimizer is a design space showing these tradeoffs as shown in Figure 4.11. In our example, the dark line is the solution chosen. The run time of the optimization, as with any optimization, is dependent on the constraints placed by the user and the number of variables fixed. The test benches must be constructed with an understanding of the technology to be used and the specifics of the design such as matching. In this work, Creative Genius™ [93] was used in the optimization procedure. It has the advantage of being able to handle a large number of variables. The user is able to see graphically the trade-offs between the different choices and is thus able to make a more informed decision for every specification of the core.

![Figure 4.11 Optimizer output](image)

4.4. Physical view

After the widths and lengths of the architecture are specified, layout of the circuit is the next step. The last view documented for the IP is the physical/layout view. This view gives the physical constraints in the layout of the architecture. These are the restrictions placed on the design to guarantee correct operation. One of the most common considerations in design is equating rise and fall times for the output. Another is ensuring symmetry between branches for differential VCO
architectures. Minimization of channel length modulation effects is achieved by setting a lower limit for the length of each transistor channel. The criteria needed to meet these conditions are described in terms of the configuration's transistor sizing. These constraints can be fed into an automatic layout tool or used in the manual layout process. For the differential architecture used here, the constraints illustrated in Figure 4.12 must be met.

![Matching Constraints](image)

**Figure 4.12** Physical view constraints

In addition to the physical constraints on the layout, this view should provide the user with the placement of the transistors in the circuit. Figure 4.13 shows the placement for one delay stage, which is replicated depending on the required number of stages, to form the VCO.

![Placement of single delay cell](image)

**Figure 4.13** Placement of single delay cell
4.4 0.18μm Fabricated Design

For verification of the design and models developed, this specific core was designed and fabricated in 0.18μm technology, shown in Figure 4.14. In this section, the experimental results from testing the chip are summarized. The three properties of the core, frequency characteristic, jitter and PSRR are measured.

![Figure 4.14 VCO layout](image)

**Frequency Characteristic**

Figure 4.15 shows the frequency characteristic for the fabricated 15-stage differential VCO architecture. The frequency of the simulated design is on average 1.7 times the frequency of the fabricated results. If we assume that all transistors are either on or off ($V_{control} = 1.8V$), the delay per stage is calculated from the measure frequency and is, $\tau = 0.234\, ns$. The resistance calculated using
the BSIM model is $R = 5.2\, \Omega$, and from $\tau = RC$, the capacitance of each stage for the fabricated architecture is calculated to $C = 45\, \mu F$.

![Fabricated VCO frequency characteristic](image)

Figure 4.15 Fabricated VCO frequency characteristic

The capacitance calculated using the model is, $C = 32\, \mu F$, while from simulations $C = 29\, \mu F$. There is a $+50\%$ error. This difference in capacitance between fabrication and simulation/model can be partly attributed to parasitic and wire capacitance. An approximation of the wiring and parasitic capacitance needs to be included in the models of the analytical view of the core. This extra capacitance factor is dependent on the layout of the design. A factor dependent on the parameters of the design, $W$ and $L$, is included in the analytical view of the IP to provide an error margin for the operational range of the core. Figure 4.16 shows the characteristic function of the 15 stage VCO with and without consideration of the extra parasitic factor. This error margin must be provided with the IP as part of its analytical view.
Phase noise/Jitter

Next a comparison of phase noise values for the fabricated architecture and simulation is done. Phase noise values are measured using a spectrum analyzer. The difference between the magnitude of the centre frequency and the magnitude at the offset frequency specified, $\Delta$, is used to calculate the phase noise in dBC/Hz.

$$\text{Phase noise}(dB) = \Delta - 10 \log(RESBW) + 2.5 \quad (3.9)$$

The resolution bandwidth ($RESBW$) term is needed to calculate the phase noise per Hertz bandwidth and the constant is needed to take into account the under measurement of random noise.
by the spectrum analyzer. Using equation 4.11, the phase noise is calculated for a varying supply voltage, the results are shown in Table 4-2. The error as compared with the average of simulation is 34%.

<table>
<thead>
<tr>
<th>VDD</th>
<th>Phase noise @ 500 KHz</th>
<th>Phase noise @ 1 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(dBc/Hz)</td>
<td>(dBc/Hz)</td>
</tr>
<tr>
<td>1.62</td>
<td>-94.0</td>
<td>-95.7</td>
</tr>
<tr>
<td>1.8</td>
<td>-97.1</td>
<td>-98.8</td>
</tr>
<tr>
<td>1.98</td>
<td>-88.9</td>
<td>-93.0</td>
</tr>
</tbody>
</table>

Table 4-2 Fabricated phase noise values

Using integration, the jitter is calculated, $J_{rms} = 22.7\text{ps}$.

**Power Supply Rejection Ratio**

PSRR values for the fabricated architecture are shown in Table 4-3 for ±10% variation in the nominal supply voltage of 1.8V.

<table>
<thead>
<tr>
<th>Power Supply(V)</th>
<th>Frequency(MHz)</th>
<th>PSRR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.62</td>
<td>88.75</td>
<td>35.0</td>
</tr>
<tr>
<td>1.98</td>
<td>90.05</td>
<td>24.1</td>
</tr>
</tbody>
</table>

Table 4-3 PSRR calculated values for fabricated architecture
The graphical representation of the PSR is shown in Figure 4.17. The figure shows the variation in operation frequency is within 10% of the nominal values, a 5% increase from simulation. Therefore, a ±5% factor should be included in the documentation to account for the physical variation.

![Figure 4.17 PSR for fabricated design](image)

To summarize, this chapter presented the steps needed to develop an IP core, the VCO. A core was then designed using the proposed approach and fabricated in 0.18μm technology. The experimental measurements from the fabricated architecture were then compared to the simulation results. In the next chapter, the IP cores for the other blocks of the PLL are presented in reusable format.
5 A SAMPLE OF PLL LIBRARY

COMPONENTS

It is difficult to generate a PLL directly from behavioral descriptions. Instead, for design of the PLL, a well-characterized library of different elements with their different views can enable PLL generation. The reusability of the PLL as a system comes from the reusability of its building blocks and their optimization. The PLL architecture presented here consists of four blocks, the VCO, PD, CP, and LPF. In Chapter 4, we presented the steps needed to characterize the VCO block as a reusable firm IP. In this chapter, a set of other PLL blocks are characterized for their inclusion in the library. This involves the analysis and generation of the different views for all architectures presented.

5.1 Phase Detector Library Element

The PD detects the phase difference between the external reference signal and the PLL output signal. This difference is converted to voltage, or current, to control the output frequency of the VCO. The phase detector implementation can be a simple XOR gate, an SR-flip-flop, or a multiplier. A commonly used configuration in integrated applications today is the phase/frequency detector (PFD) which consists of two reset enabled D-flip-flops and an AND gate. This is shown in Figure 5.1 together with the state diagram representation of the circuit. The PFD is characterized by its ability to track phase and frequency errors. It is connected to the CP block whose output, \( V_{\text{control}} \), adjusts the VCO frequency.
There are three stable states for the PFD, when both outputs are low and when either one is low while the other is high. A positive transition in one of the inputs will cause the circuit to change state. As we mentioned in Chapter 3, a buffer is needed in the reset path of the PFD to avoid dead zone operation. The finite delay in the reset path introduces spurs at the output as shown in Figure 5.2.

![Figure 5.1](image)

**Figure 5.1** (a) Phase/frequency detector architecture (b) PFD state diagram

![Figure 5.2](image)

**Figure 5.2** PFD theoretical and practical signals

To eliminate the spurs at the output caused by the finite reset delay time while maintaining correct operation, extra gates are incorporated into the design as shown in Figure 5.3. The output of the PFD is now a delayed logical function of the basic PFD outputs UP and DOWN.
With this architecture, the two outputs of the PFD are guaranteed not to be high at the same time. Therefore the CP will always operate in the correct region. The modified outputs have the following functions

\[ UP_{\text{mod}} = UP \cdot DOWN \]  \hspace{1cm} (5.1)

\[ DOWN_{\text{mod}} = \overline{DOWN} \cdot UP \]  \hspace{1cm} (5.2)

\[ \begin{array}{|c|c|c|c|}
\hline
UP & DOWN & UP_{\text{mod}} & DOWN_{\text{mod}} \\
\hline
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
\hline
\end{array} \]

Table 5-1 Modified PFD logical table
There are different possible implementations for the D flip-flop. For this library element, we use the D flip-flop architecture shown in Figure 5.4. The highlighted transistors implement the reset for the flip-flop. This is a positive-edge triggered architecture with an asynchronous reset.

![Figure 5.4 D flip-flop with reset schematic](image)

The frequency specification of the PLL sets a constraint for the PFD design. The PFD maximum frequency must meet the specification

$$f_{\text{mix}} < f_{\text{in,max}}$$

for all input frequencies, where \(f_{\text{max}}\) is the maximum operational frequency of the PFD. To characterize the PFD, bottom-up modeling using circuit analysis is required. The resulting models characterize the operational space of the PFD in terms of the delay reset path and the time constant at its output.

The reset delay bounds the maximum PFD frequency through the relationship:

$$f_{\text{max}} = \frac{1}{2\tau_{\text{reset}}}$$

(5.3)

The reset delay, \(\tau_{\text{reset}}\), is calculated using an RC delay model for the reset buffer. This is shown in Figure 5.5. The multiplication factor used for the device widths is 4.
Figure 5.5 Reset path buffer chain

\[ W2 = 2.5*W1 \]
\[ M3 = 4*W1 \]
\[ M4 = 4*W2 \]
\[ Mx = 4*M(x-2) \]

\[ \tau_{reset} = t1 + t2 + t3 + ... + tx \] (5.4)

The total reset delay is expressed in the form

\[
\tau_{reset} = C + \sum_{x=1}^{X} \left( \frac{a}{W_x} \left[ bW_{x+1} + cW_x \right] \right)
\] (5.5)

where \( a, b \) and \( C \) are functions of the capacitance and resistance of each gate in the chain and is therefore technology dependent.

The second frequency constraint modeled is the time constant at the output of the PFD. For correct operation, the rise and fall times at the output of the PFD has to be less than half the period of the input signal. The output time constant is defined by \( \tau = R_{out}C_{out} \). The total capacitance at the output node of the PFD is the sum of the output capacitance of the PFD and the load capacitance,

\[ C_{out} = C_{out-PFD} + C_{load} \] (5.6)

Therefore there is a maximum allowable value for the load capacitance, \( C_{load} < C_{max} \). Detailed derivation of the time constant and frequency constraints for the basic PFD architecture is
presented in Appendix B. For the modified architecture of Figure 5.3, the output resistance and capacitance are those of the output AND gate which simplifies the derivation of the load capacitance value. The frequency constraint is now due to two time constants, that of the basic PFD and the response time for the AND gate. Standard library cells for the DFF are suitable for most applications, especially when considering frequencies less than 1GHz.

Behavioral view

The behavioral view of the PFD is shown in Figure 5.6. For this block, technology dependent timing response is the limiting characteristic.

```verilog
module pfd_cp (UP, DOWN, ref, track);
    if (Ref & Track) begin
        state = 'same;
    end else if (Ref) begin
        if (state == 'behind) begin
            state = 'same;
        end else if (state == 'same) begin
            state = 'ahead;
        end else if (Track) begin
            if (state == 'ahead) begin
                state = 'same;
            end else if (state == 'same) begin
                state = 'behind;
            end
        end
        if (state == 'ahead) begin
            UP = 0;
            DOWN = 1;
        end else if (state == 'same) begin
            UP = 0;
            DOWN = 0;
        end else if (state == 'behind) begin
            UP = 1;
            DOWN = 0;
        end
        end
endmodule
```

Figure 5.6 PFD behavioral view

Physical view

In the layout of the PFD block, care must be taken in equating the signal paths of the UP and DN signals. Equal delay for both output signals will mean faster response time during the pull in process of the PLL. The transistors are sized to ensure equal response times for high and low transitions and the architecture is laid out such that the metal paths of the two signals are as equal as possible.
A symmetrical NAND gate architecture is used as shown in Figure 5.7. Figure 5.8 shows a symmetrical layout for the PFD block.

![Symmetrical NAND gate](image)

**Figure 5.7 Symmetrical NAND gate**

![PFD layout](image)

**Figure 5.8 PFD layout**

**Test bench view**

The test bench for the PFD tests the response of the PFD to varying input frequencies, and calculates the reset time and dead zone of the PFD. The test benches are also used with an optimizer to maximize the performance of the architecture. The reset time is tested by applying high Reference and track signals simultaneously to the input of the PFD, as shown in Figure 5.9. The reset time is defined as the time from the trigger signal at 0.5Vdd to event at 0.5Vdd. The second test bench, shown in Figure 5.10, measures the dead zone of the architecture. For this test bench, an ideal CP is needed to generate the output current. Two input signals, having the same
frequency but different phase, are applied to the system. This phase difference is varied and the output current is measured.

**Figure 5.9 PFD reset time test bench**

**Figure 5.10 PFD dead zone test bench**
5.2 Charge Pump Library Elements

The charge pump provides the input control signal to the VCO. This block controls the variations in the VCO frequency in response to the PFD output. For our library element, a simple single-ended structure is used. Single-ended charge pumps are commonly used because of the simplicity of their design and their lower power dissipation [84].

The first CP library element is the architecture used in the design of the PLL of Chapter 3. This is the switch-in-source design, shown in Figure 5.11. In this architecture, the UP switch is implemented with a PMOS transistor while the DN switch uses an NMOS transistor. The current mirrors regulate the output current, which is zero when both input signals are low.

![Figure 5.11 Switch in source charge pump](image)

A more robust charge pump architecture is the single-ended architecture shown in Figure 5.12 [84]. In this architecture, the output impedance is increased by using cascode current mirrors (\(M_1 - M_6\) & \(M_{p1} - M_{p4}\)). This reduces the sensitivity of the output current to output voltage. \(M_{C1}\) and \(M_{C2}\) reduce the charge coupling to the gate, enhancing the switching speed. \(M_{R1}\), \(M_{R2}\) and \(M_{R3}\) are used for replica biasing to give the same bias condition when the charge pump is turned on.
The switching time for the CP is of the form, \( T_{sw} = f\{W_{UP}, W_{DOWN}, L_{UP}, L_{DOWN}, V_t\} \), the value of which bounds the frequency operational space of the core. The power dissipation of the CP is \( P = f\{I_{CP}, Vdd\} \), and sets a lower bound for the overall power dissipation of the PLL.

The most critical design consideration for the charge-pump is current matching. The charge and discharge current for the same error signal should always be equal. This is achieved through careful design and layout. The biasing conditions to optimize current matching requires high effective gate voltages. Assuming the biasing conditions have been considered, the matching is now dependent on the layout process. There are several factors that must be considered during layout [50], some of which are discussed in the physical view section for the CP.
Behavioral view

This behavioral view for the CP is shown in Figure 5.13. It assumes current matching and instantaneous switching. A more detailed representation will take into account a user-defined error for matching as well as incorporate the finite switching time of the transistors.

```verilog
module charge_pump(UP, DN, vref);
  input UP, DN;
  inout vout;
  if (UP == DN) begin
    i_direction = 0.0;
  else if (UP) begin
    i_direction = 1.0;
  else if (DN) begin
    i_direction = -1.0;
  end
  analog begin
    @ (initial_step ) begin
      i_cp = i_mag*i_direction(U, DN, vref);
    end
    @ (cross(V(UP), 0.9)) begin
      i_cp = iamp*i_direction(V(UP),V(DN),vref);
    end
    @ (cross(V(DN), 0.9)) begin
      i_cp = iamp*i_direction(V(UP),V(DN),vref);
    end
  end
endmodule
```

Figure 5.13 CP behavioral view

Physical view

For the CP, current matching is an important consideration during layout. Reference [50] details the steps needed to produce a matched layout. Three important factors are: use of identical finger geometries, orientation of transistors in the same direction and close proximity placement of transistors. This reduces the vulnerability to gradients in temperature, stress and oxide thickness.

The placement and layout of the transistors for the CP architecture is shown in Figure 5.14. To minimize mismatch, transistors are folded around the axis of symmetry in an ABBA configuration.
Both PMOS and NMOS transistors should be placed far from the Nwell edge to avoid diffusion variation.

![Figure 5.14 CP transistor placement](image)

**Test bench view**

The test bench is designed to ensure that the block meets the target specifications. It is also used with an optimizer to maximize the performance of the circuit. For the CP, careful biasing is required which translates to fewer degrees of freedom in the optimization process. The test bench constructed measures the response of the CP to varying input signals in different simulation corners as is shown in Figure 5.15.
For all simulation corners:
1. \( f(\text{UP}) > f(\text{DOWN}) \)
2. \( f(\text{UP}) < f(\text{DOWN}) \)
3. \( f(\text{UP}) = f(\text{DOWN}) \)
4. \( \text{UP} = \text{DOWN} = 0 \)

Figure 5.15 CP test bench

5.3 Loop Filter Element

The element considered is a second order passive filter shown in Figure 5.16. Because of the simplicity of this architecture, the different views as defined are not useful.

The values of the filter components determine the positions of the poles and zeros in the system and therefore bound its bandwidth. These are given by: the zero is at \( \omega_z = 1/(RC) \) and the pole is at \( \omega_p = (C_1+C_2)/(RC_1C_2) \).

The next chapter presents the PLL architecture generated through the use of the concepts introduced in this work. The primary specifications of the PLL are changed for this second architecture and the blocks are restricted to the library elements characterized thus far.
Chapter 6

6 LIBRARY GENERATED PLL IP

In Chapter 5, a sample of library components for PLL design was assembled. In this chapter, a new PLL generated from the IP library is designed to illustrate the concept of PLL reusability. For this purpose, the primary specifications of the PLL designed in Chapter 3 are changed to target another application. The system clock for the analog front end required for Ethernet applications. These changes require architectural modifications of the design to meet these new specifications. The flow as defined in Chapter 2 is used to harden the generated PLL firm IP. The design time for both architectures is then compared. This design is currently in fabrication in 0.18μm CMOS technology.

The specifications for the new PLL architecture are:

- Centre frequency = 250MHz
- Locking time ≤ 5μs
- RMS jitter < 100ps
- Static phase error < 250ps
- Power < 1mW

6.1 PLL Design

The first stage in the design process is the top-down constraint propagation to lower level system sub-blocks. The second step is the selection of the cores needed to meet the specifications and their
optimization to maximize performance. The optimized architecture is then taken through the hardening flow to generate the hard IP from the firm IP.

**VCO Design**

The design specifications can be met by both VCO elements in the library. However, the PSRR value for the differential architecture is higher than for the current starved one. This implies a higher tolerance for variations in the supply, and the differential architecture has the advantage of common mode noise rejection making it a more robust one. The architecture used is shown in Figure 6.1. Using the analytical models of the core, the transistor sizes needed to meet the system level specifications are generated and the circuit optimized to maximize performance using the test benches presented in Chapter 4.

![VCO delay cell](image-url)

*Figure 6.1 VCO delay cell*

The frequency characteristic for this configuration is shown in Figure 6.2.
PFD Design

The PFD architecture used in this design is the modified one shown in Figure 6.3.

Figure 6.3 PFD schematic

Figure 6.4 shows the operational behavior of the chosen PFD.
CP & Filter Design

Both CP library elements satisfy the PLL specifications. However, the architecture shown in Figure 6.5 is more robust and is hence used in the design. Only one filter element has been characterized and used in the design, i.e. the passive filter shown in the figure.
The output behavior of the CP & filter architecture is shown in Figure 6.6.

![Figure 6.6 CP output behavior](image)

This design is currently being fabricated in 0.18μm CMOS technology. The layout of the PLL is shown in Figure 6.7.

![Figure 6.7 PLL layout](image)
6.2 Test Benches

In this section, the test benches designed to ensure correct performance of the PLL are described. Each test bench targets one of the specifications.

**Lock time**

The testbench for lock time of the system monitors the transient control voltage. If for a span of 1μs, the control voltage is within 5% of a specified value (Specified approximately using the analytical view of the VCO), this time step is considered equal to the lock time.

**Static timing error**

The static timing error of the system is measured by measuring and comparing the zero crossings of the reference input and PLL output signals. At 1μs intervals, the zero crossing of the output signal of the PLL is measured against the theoretical zero crossing of the reference signal.

**RMS Jitter**

The same approach used for determining the static timing error is used to calculate and minimize the jitter of the system. By measuring the zero crossings of the PLL output signal, the characteristic for the period of the output signal can be measured. To calculate the RMS jitter of the system, the output period is compared with an ideal reference signal and a distribution of these values is collected. The jitter is calculated by finding the standard deviation of this distribution.

**Power dissipation**

To measure the power dissipation of the system, the current drawn from the power supply is measured and multiplied by Vdd.
6.3 Results

The designed PLL architecture has the following performance specifications:

- **Locking time**, $t_{\text{lock}} \approx 0.35 \mu s$
- **Static timing error**, $r_{\text{err}} = 0.025 \text{ns}$
- **RMS jitter**, $J_{\text{RMS}} = 29 \text{ps}$
- **Power dissipation in lock**, $P_{\text{lock}} = 0.63 \text{mW}$

Figure 6.8 Control voltage of generated PLL

Figure 6.8 shows the transient response of the control voltage for PLL as it is locking to an input frequency of $f_{\text{in}} = 250 \text{MHz}$. The lock time for the PLL for 5% error is approximately $0.35 \mu s$ which
is about 90 reference periods. Figure 6.9 shows the input reference, the PLL output signal, and the buffered output of the PLL after a lag of 600ns. The output buffer needed to drive the output load introduces an extra delay and reduces the amplitude of the driven signal. To calculate the RMS jitter of the system, the output signal period is recorded as the system locks; this is shown in Figure 6.10.

![Figure 6.9 Input and output signals of PLL](image)

The distribution of the difference between the ideal reference signal and the output signal is shown in Figure 6.11, the standard deviation of which is the RMS jitter of the PLL. The standard deviation of this distribution is the value of the jitter. For this system, $J_{RMS} = 29\text{ps}$. 
Figure 6.10 Output signal period variation as the loop locks

Figure 6.11 Output signal period deviation from ideal reference
The time required to take this design from concept to silicon was approximately 16 days or about 175 man-hours as compared with 750 man-hours for the Bluetooth PLL design. This is a significant reduction in the design time, a factor of 4.

The knowledge gained from going through the initial design plays a large role in this reduction in design time. However, this increased expertise was used to capture better IP characterization and can then be used in future design versions.
Chapter 7

7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This thesis has addressed the issues of reusable AMS IP using a phase-locked loop as the design vehicle. It is clear that the development of AMS IP must take a different approach compared to digital IP development. The IPs must be able to handle and transfer both design experience and heuristics from the original design to subsequent design derivatives. This is really the reusable part of the analog design process. The steps involved in AMS IP creation are much more labor-intensive than the digital case, even for relatively small blocks. The development process or design methodology employed in the development of these AMS IPs plays a critical role in the degree of reusability possible.

PLLs can not be easily synthesized from a high level specification without low-level support. This observation is also true for many other AMS functions. Therefore, a library of parameterized reusable components is an essential part of the design flow. In this work, a design flow based on top-down specification and bottom-up characterization was proposed. This proposed method has constraints flowing top-down from the specifications, while bottom-up custom models are used to enable quick selection from the different possible configurations within an assembled library. These bottom-up models completely characterize the IP core for the library. Once a specific architecture
is selected, it is taken through the IP hardening flow which involves optimization of the design to maximize performance and to generate the final GDSII layout.

Firm IP was concluded to be the most appropriate format to deliver the AMS IP library components. Unlike hard IP, this form allows ease of migration of IP from foundry-to-foundry, customer-to-customer and application-to-application. In the first phase of this work, the firm IP format for AMS cores was defined together with a design-for-reuse methodology for the development of AMS library components. The firm IP completely characterizes the core, while the hardening flow generates the GDSII layout guided by the firm IP physical views.

The second phase of the work applied these concepts to the case study, a PLL. Design for reuse was investigated first by designing (from specifications to layout) a PLL architecture. The different blocks within the PLL were then designed for reusability and a small IP library is created. The primary specifications of the PLL were then changed and a second architecture was generated using the design methodology presented. This PLL architecture had the same topology but was designed using other library elements. The time needed to design the second PLL is shown to be greatly reduced (by a factor of four) for comparable performance.

This reduction in the design time could be attributed to two interdependent factors: the increased expertise of the designer and the captured design knowledge in the different firm IPs. However, most of the experience gained by the designer was used to improve on the documentation of the firm IP which leads to increasing gains in productivity. This work shows that design experience acquired during the design process for AMS cores can be efficiently captured in firm IP for eventual use in derivative designs.

82
7.2 Research Contributions

The goal of this research was to offer guidelines and methodologies for analog/mixed-signal IP design and reuse. In this work, we have defined the most suitable AMS IP format for reuse as firm IP. This work presented the information necessary to define firm AMS IP and showed that this method results in reduced design time for a PLL case study. The design methodology presented here exploits the benefits of both top-down and bottom-up design approaches. There is higher confidence in the fabricated chip performance with an increased reuse value of the design. A summary of the contributions are as follows:

- Top-down design and bottom-up component selection methodology
- Analog IP hardening flow
- Design for reuse process for firm IP
- Design of library components for PLL design

7.3 Future Work

The proposed approach can significantly reduce PLL design time. However, it has focused on manual implementation using existing tools. There is potential for this approach to be automated so that the design times can be reduced even further. Therefore, a future goal for this research is to develop a CAD flow based on this design methodology.

Currently, only a handful of building blocks have been characterized and used as library elements. A significant effort is needed to provide a rich catalog of library elements for a variety of different types of PLLs. This would broaden the scope of PLLs that could be designed by the methodology described here. Further, this work has concentrated on the design of PLL cores but it needs to be
extended to cover other AMS applications such as ADCs, DACs, and OPAMPS. This would serve as another area of fruitful research and development.
Appendix A

Developments in AMS Design & Layout

AMS Design

The following section gives an overview of research in the area of AMS design and synthesis, for more detailed information, the reader is directed to the associated references.

*IDAC* [84]

Specific heuristic design knowledge about the circuit topology under design is encoded in some computer executable form. This design form is then executed during synthesis (transistor level simulation) for a given set of specifications to obtain the design solution. The design equations for a particular topology together with the degrees of freedom had to be derived. These are then solved explicitly using simplifications and design heuristics. The main advantage of this approach is the fast execution time but with the disadvantage of long plan development time and lack of flexibility.

*OASYS* [86]

This tool has a sizing based design approach but with a hierarchical element introduced in the procedure. This tool also added a heuristic approach for topology selection but still has the disadvantage of a long design time.

*WiCked* [18]

This tool presents a simulation based analog synthesis platform that incorporates mismatch using a customized sequential quadratic programming, deterministic, approach to achieve the solution.
**ASF (Analog Synthesis Framework)** [31]

Simulation based methodology for the synthesis of custom analog circuits. The synthesis engine within this tool works at the cell level. The optimizer is based on parallel re-combinative simulated annealing that allows parallel annealing type stochastic optimization.

**ADSA (Analog Design Synthesis Assistant)** [87]

This is a top down design methodology to accommodate circuit interaction within an analog synthesis environment. The end product for the ADSA system is a fully designed circuit schematic and high-level design specification. A key feature of the system is the ability to co-simulate both behavioral and circuit level but a major task is the breaking down of the circuit into the different blocks. To accommodate the effects of loading, the circuit is grown from the critical seed and impedance constraints are propagated to the neighboring blocks. In this approach, all system blocks are designed in stages, progressively refining their behavioral models, using parameters extracted following circuit-level design.

**AMGIE** [26]

This synthesis environment covers the complete design flow from specifications over topology selection and optimal circuit sizing down to automatic layout generation and performance characterization. It has a hierarchical, process independent design approach. This system delivers fast results but with the disadvantage of a long and tedious plan creation stage.

**ASTRX/OBLX** [24]

This synthesis tool automates the design flow to produce a fully sized netlist. It is an equation based optimization tool using simulation-based optimization for linear small-signal characteristics. It relies on asymptotic waveform evaluation to predict circuit performance and simulated annealing to solve the optimization formulation.
Based on redesign, this tool starts its simulation from the previous design solution stored in the system database. Therefore it requires an extensive first set of simulations from which the first solution is developed on which the consecutive solutions are based and is specific to the design of OPAMPS.

**Layout strategies:**

The earliest layout approaches to automatic layout relied on procedural module generation. Changes in the circuit may require full custom layout, which can be modeled using a macro cell strategy. Module generation techniques are used to generate the layout of individual devices. A problem with this approach is extending these primarily digital algorithms to handle all the low level geometric optimizations that characterize expert manual design. The next generation of tools kept the macrocell style but reinvented the necessary algorithms from the bottom-up incorporating many manual design optimizations. The next version of cell level tools shifted the focus to quantitative optimization of performance goals and implementing improved cost-based schemes.

In the next generation, device placement has been divided into, device stacking and stack placement. The approach identifies clusters of devices that ought to be merged, stacks, to minimize capacitance. This is done using either, exact algorithms to extract all the optimal stacks or a placer that extends an algorithm to dynamically choose the right stacking and the right placement. Instead of extracting all stacks, this technique extracts one optimal set of stacks.

Recent tools introduced commercially as automatic layout generators compatible with the analog/mixed-signal design domain are Silicon Canvas Laker [89], Sagantec Anaconda [90] which allows designers to reuse existing layout topologies and implement modified device parameters and design
constraints from a schematic. CiraNova [91] allows users to choose the best combination of automation and manual editing. Controlling the amount of automation interactively, designers can create optimized, high performance analog physical layout from schematics. This tool relies on propagating the layout designer knowledge on to the next version of the design making it similar to Sagantec Anaconda. InternetCad [92] is a timing driven placement and routing tool applicable to row based and building block design styles. NeoCell [13] is a constraint-driven layout-synthesis tool that captures these vital constraints interactively. It allows the user to flexibly specify your cell's geometric architecture, synthesize the geometry for individual devices, and place and route them automatically.

The quality of the final design is dependent on the ability of the device generators to take into account analog-specific constraints such as mismatching, symmetry and capacitance minimization by merging. A performance-driven approach to layout for analog design is still in its infant stages.
Appendix B
Analytical model development

B.1 VCO Modeling

To calculate the capacitance and resistive characteristics of the oscillator, the capacitance and resistance of each individual transistor must be estimated. For this estimation, the three dimensional layout of a single transistor shown in Figure B.1 is used.

![Transistor Capacitance
Cgdo: Overlap capacitance
Cj:Junction capacitance
Cjsw: Sidewall junction capacitance
Cjswg: Bulk source/drain junction capacitance](image)

Figure B.1 3D view of transistor layout & capacitance associated with it

The resistance of a single transistor is dependent on the values of the width and length, W and L, and the capacitance that is dependent on W, L, X and Y.

To analyze the frequency characteristics of the different oscillator configurations, the most basic oscillator configuration, as shown in Figure B.2, is used to validate the modeling technique employed. This oscillator is comprised of an odd number of inverters in a loop. As there is no direct control over the delay per stage, only one frequency of oscillation for each configuration is possible. The frequency of oscillation for ring oscillators is given by $f = 1/2N\tau$ [67]. Where $N$ is the number of stages and $\tau$ is the delay per stage. To change the oscillation frequency, the number of
stages in the oscillator, or the delay per stage is varied. Therefore the parameters of the design are 
N, W and L.

![Inverter ring oscillator](image)

**Figure B.2 Inverter ring oscillator**

The delay of each stage can be modeled quite simply and accurately using an RC model, i.e. \( \tau = RC \).

In this approach, the capacitance and resistance at the output node of the inverter (assuming sized for equal pull-up and pull-down) determines the delay of the individual cell and hence the frequency of the VCO can be calculated. The inverter cell is modeled as shown in Figure B.3.

![Inverter delay cell schematic and model](image)

**Figure B.3 Inverter delay cell schematic and model**

For equal rise and fall times, the transistors are sized such that \( W_p = 2.5W_n \), the values for the resistance, \( R_{up} = R_{dn} \), can then be calculated from...
\[
R_{ap} = R_{eqp} \frac{L_p}{W_p} \\
R_{down} = R_{eqn} \frac{L_n}{W_n}
\] (B.1)

The capacitance at the output node of the inverter is given by:

\[
C = C_{dp} + C_{dn} + C_{gp} + C_{gn} \\
= (W_n Y_n C_{j} + 2Y_n C_{jswn} + W_n C_{gdon}) + (W_p Y_p C_{j} + 2Y_p C_{jswp} + W_p C_{g dop}) \\
+ C_{ox} (W_p L_p + W_n L_n)
\] (B.2)

Using equations 4.1 and 4.2, the capacitance and resistance are calculated for the experimental architecture, from which the delay is also calculated.

\[
C = 4.2 \, fF \\
R = 6.5K\Omega
\] (B.3)

\[\Rightarrow \tau = 27.3 \, ps\]

These model results are compared with the Spice simulation results. The simulation results for three configurations, 3, 9 and 15 stage oscillators are shown in Table B-1.

<table>
<thead>
<tr>
<th>Number of stages</th>
<th>Oscillation frequency (GHz)</th>
<th>Delay (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5.607</td>
<td>29.7</td>
</tr>
<tr>
<td>9</td>
<td>1.788</td>
<td>31.0</td>
</tr>
<tr>
<td>15</td>
<td>1.073</td>
<td>31.1</td>
</tr>
</tbody>
</table>

*Table B-1 Oscillation frequency and delay for inverter oscillator*
The average delay for the simulated configurations is $T = 31\text{ps}$. The error obtained for the model calculation is $11\%$. This is an acceptable error margin as the calculation is for the estimation of the approximate architecture configuration that is then further refined in the transistor-level stage of the design process.

**Differential Configuration**

For the differential configuration, the model is designed to calculate the delay per stage, $T$. For the VCO delay cell shown in Figure B.4.

![Differential VCO delay cell](image)

Figure B.4 Differential VCO delay cell

The load capacitance in-between consecutive stages is given by:

$$C_L = C_{d1} + C_{d2} + C_{d3} + C_{d4} + C_{d5} + C_{d6} + C_{d7} + C_{d8} + C_{g4} + C_{g6}$$

(B.4)

where each drain capacitance can be expressed as

$$C_d = C_{gdo} W + W Y C_j + x_j W C_{jswg} + x_j Y C_{jsw} + W x_j C_{jsw}$$

(B.5)
Defining the following

\[
C_N = C_{gdoN} + YC_{jN} + x_{jN} C_{jswN} + x_{jN} C_{jswH}
\]
\[
C_P = C_{gdoP} + YC_{jP} + x_{jP} C_{jswP} + x_{jP} C_{jswP}
\]

(B.6)

The load capacitance during a low-high transition is expressed as:

\[
C_{L(l-h)} = C_P (W_1 + W_2) + 2x_jYC_{jswP} + 2C_P (W_5 + W_6) + 4x_jYC_{jswP}
+ C_N (2W_3 + 2W_4) + 2x_jYC_{jswN} + C_N (W_7 + W_8) + 4x_jYC_{jswN}
+ C_{oH} l(W_4 + W_6 + W_9)
\]

(B.7)

And similarly for a high-low transition

\[
C_{L(h-l)} = C_P (W_1 + W_2) + 2x_jYC_{jswP} + 2C_P (W_5 + W_6) + 4x_jYC_{jswP}
+ C_N (2W_3 + 2W_4) + 2x_jYC_{jswN} + C_N (W_7 + W_8) + 4x_jYC_{jswN}
+ C_{oH} l(W_4 + W_5 + W_8 + W_{10})
\]

(B.8)

The delay, \(\tau = RC\), is therefore given by

\[
\tau_{(l-h)} = R_{eqP} \frac{L_2}{W_2} C_{L(l-h)}
\]

(B.9)

\[
\tau_{(h-l)} = R_{eqN} \left[ \frac{1}{2} + \frac{1 + L_2}{2W_4} \right] C_{L(h-l)}
\]

(B.10)
To model the frequency space dependency on the number of stages of the oscillator, a fixed value for the control voltage, $V_{\text{ctrl}} = 1.4V$, is assumed. The current through each delay stage, assuming operation in the saturation region, can be calculated using

$$I_D = \frac{\mu_n C_{ox} W_n}{2L_n} (V_{GS} - V_m)^2 (1 + \lambda V_{DS})$$  \hspace{1cm} (B.11)

The derivative of equation (B.11) is the impedance of an individual NMOS transistor when operating in the saturation region and is equal to

$$\frac{\partial I_D}{\partial V_{DS}} = \frac{\mu_n C_{ox} W_n}{2L_n} (V_{GS} - V_{tp})^2 \lambda$$  \hspace{1cm} (B.12)
Equation (B.12) shows the variable impedance of the transistor, M4, as a function of the applied gate voltage when mobility, \( \mu \), and threshold voltage, \( V_t \), is assumed constant. For the variable resistance of M1, the PMOS equivalent of the impedance is given by

\[
\frac{\partial I_D}{\partial V_{DS}} = \frac{\mu_p C_{ox} W_p}{2L_p} (V_{GS} - V_{tp})^2 \lambda
\]  

(B.13)

For both NMOS and PMOS transistors, the value of the channel length modulation index is approximated to \( \lambda \approx 1 \).

The capacitance at the output node is given by

\[
C = C_d + C_s + C_g
\]

(B.14)

And the resistance in the pull down path is given by \( R_{ds} = R_m + R_s \) and in the pull up path \( R_{up} = R_{tp} + R_s \).

The average delay from the simulation results is \( \tau = 54 \text{ps} \). This gives the error in model calculation as compared with the average of the simulated delay values is 3.0%.

The second VCO frequency characteristic modeled is the dependency of the operation frequency on the applied control voltage. To model this frequency characteristic of the oscillator, the value of \( V_{control} \) is extended to cover a range of values. Using the derived equations, the variable resistance value is calculated for varying control voltage. These calculated values are then compared with values from simulation of the circuit in Spectre. Table B-2 shows these results for three applied control voltage values for a 3-stage ring oscillator based on the current starved delay cell. As can be seen from the table, the error for mid-range control voltage is lower than at when the control voltage is closer to the supplies. This can be explained by the change in the threshold voltage and
mobility variation for the transistor as this was not accounted for in the model. As can be seen from the table, the maximum error is 13%.

<table>
<thead>
<tr>
<th>Control Voltage (V)</th>
<th>Delay (ps)</th>
<th>Calculated Frequency (GHz)</th>
<th>Simulated Frequency (GHz)</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>76.0</td>
<td>2.2</td>
<td>2.5</td>
<td>12%</td>
</tr>
<tr>
<td>1.1</td>
<td>52.1</td>
<td>2.9</td>
<td>3.1</td>
<td>6.4%</td>
</tr>
<tr>
<td>1.2</td>
<td>52.1</td>
<td>3.2</td>
<td>3.3</td>
<td>3.0%</td>
</tr>
<tr>
<td>1.3</td>
<td>46.3</td>
<td>3.6</td>
<td>3.3</td>
<td>10.8%</td>
</tr>
</tbody>
</table>

Table B-2 Frequency characteristic calculation and comparison

The model generates for the experimental circuit a delay, \( \tau = RC = 52.34\text{ps} \). For a 3-stage oscillator configuration, the modeled oscillation frequency, \( f = 3.18\text{GHz} \). The model results are then compared with simulations in Spectre in 0.18\(\mu\)m, shown in Table B-3

<table>
<thead>
<tr>
<th>Number of stages</th>
<th>Oscillation frequency</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3.25GHz</td>
<td>51.3ps</td>
</tr>
<tr>
<td>9</td>
<td>977.5MHz</td>
<td>56.8ps</td>
</tr>
<tr>
<td>15</td>
<td>586.5MHz</td>
<td>56.8ps</td>
</tr>
</tbody>
</table>

Table B-3 Current starved configuration delay characteristic
B.2 PFD Modeling

The reset delay, \( \tau_{\text{reset}} \), is calculated using an RC delay model for the buffer chain.

\[
\tau_{\text{reset}} = t_1 + t_2 + t_3 + \ldots + t_x
\]  
(B.15)

The reset delay of each stage in the buffer is, \( t_i = R \cdot C_{\text{load\_buffer}} \) where

\[
R = R_{\text{eqn}} \frac{L_n}{W_n} = R_{\text{eqp}} \frac{L_p}{W_p} \]  
(B.16)

\[
C_{\text{load\_buffer}} = C_{gn} + C_{gp} + C_{dn} + C_{dp} \]  
(B.17)

To calculate the output capacitance of the PFD architecture, the output transistors of the basic PFD architecture are highlighted in Figure B.6.

Figure B.6 PFD output capacitance
For the basic PFD architecture, the total capacitance at the output node is the sum of the output capacitance of the DFF, the input capacitance of the NAND gate, and the load capacitance (the input capacitance of the next stage and wiring capacitance).

\[ C_{\text{out(DFF)}} = C_{\text{out(INV)}} + C_{\text{out(MUX)}} \]  

\[(B.18)\]

The NAND gate is designed such that both pull up and pull down delays are equal. This capacitance is equal to

\[
C_{\text{out(DFF)}} = C_{\text{dp(INV)}} + C_{\text{dn(INV)}} + C_{\text{dp(MUX)}} + C_{\text{dn(MUX)}} \\
+ 3W_p(INV) C_{jn} + 6W_p(INV) C_{jwp} + W_p(INV) C_{gdop} \\
+ 3W_n(INV) C_{jn} + 6W_n(INV) C_{jwn} + W_p(INV) C_{gdon} \\
+ 3W_p(MUX) C_{jn} + 6W_p(MUX) C_{jwp} + W_p(MUX) C_{gdop} \\
+ 3W_p(MUX) C_{jn} + 6W_p(MUX) C_{jwn} + W_p(MUX) C_{gdon}
\]

\[(B.19)\]

\[ C_{\text{in(NAND)}} = W_p L_p C_{\text{exp}} \]  

\[(B.20)\]

The total capacitance at the output of the PFD that will determine the maximum operating frequency of the PFD is

\[ C_{\text{out(PFD)}} = C_{\text{in(NAND)}} + C_{\text{out(DFF)}} \]  

\[(B.21)\]

The pull up resistance at the output of the PFD is

\[ R_{\text{out}} = R_{\text{sp}} \left[ \frac{L_p(INV)}{W_p(INV)} + \frac{L_p(NAND)}{W_p(NAND)} \right] \]  

\[(B.22)\]
which is equal to the pull down resistance

\[ R_{\text{out}} = R_{\text{SN}} \left[ \frac{I_{\text{n(INV)}}}{W_{\text{n(INV)}}} + \frac{I_{\text{n(NAND)}}}{W_{\text{n(NAND)}}} \right] \]  

(B.23)

The time constant for the PFD is calculated using

\[ \tau = R_{\text{out}}(C_{\text{out}(\text{PFD})} + C_{\text{load}}) \]

The time taken for the rising and falling edges to register is calculated using

\[ V_f = V_0 e^{-\frac{R_{\text{out}}}{C_{\text{out}(\text{PFD})} + C_{\text{load}}}} \]  

(B.24)

With the final voltage value taken as the midpoint, \( V_f = (3V_{dd}/4)V \) and the initial voltage \( V_0 = (VDD/4)V. \) This value of time, \( \tau \), has to be less than half the period of the input signal for a 50% duty cycle signal to ensure correct operation of the PFD.

For an example basic PFD architecture in 0.18\( \mu \)m gives the following results

\[ \begin{align*}
V_f &= 1.35V \\
V_0 &= 0.35V \\
C_{\text{out}(\text{PFD})} &= 5.03fF \\
R_{\text{out}} &= 11.66K\Omega \\
\text{Input frequency} &= 1GHz \\
\Rightarrow 1.35 &= 0.35e^{-5e-10/11.66e3*5.033e-15+11.66e3*C_{\text{load}}} \\
\ln(3) &= (5e-10/5.87e-11+11.66e3C_{\text{load}}) \\
C_{\text{load}} &\leq 44.0fF
\end{align*} \]
References

[1] SIA. www.semichips.org


[40] Silicon Wave: www.siliconwave.com


[93] ADA: [www.analogsynthesis.com](http://www.analogsynthesis.com)