RF Mixed Signal Design and Layout Synthesis

with Object-Oriented C++ for Nanometre SOI CMOS

by

Victor F. Karam

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Applied Science.

Ottawa-Carleton Institute for Electrical Engineering
Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, Canada

© Victor Karam, 2004
NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Canada

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Abstract

In this thesis, a new top-down methodology for radio frequency (RF) mixed-signal design is presented. This methodology is based on a new IBM electronic design automation (EDA) tool for layout synthesis. A general description of this EDA tool, as well as its capabilities and features in a RF mixed-signal design flow, is presented. Then the methodology is demonstrated by the design of a ring oscillator circuit in a 90nm silicon-on-insulator (SOI) IBM CMOS process for technology benchmarking. The ring oscillator is composed of 11 CMOS inverter circuits and the design is optimized with the EDA synthesis tool for maximum oscillation frequency. Figures of merit such as unity current gain frequency and unity power gain frequency are developed for the inverter to provide high-frequency performance insight. Inverter propagation delay in a ring is formulated to show that, for this CMOS process, an optimum device finger width of 1.4μm exists for minimizing delay in the presence of wiring parasitics.
Acknowledgements

I wish to extend my upmost thanks to my thesis supervisors, Dr. Calvin Plett and Dr. Neric Fong for their guidance, support and patience. Dr. Plett was a pillar of wisdom, and Dr. Fong was a source of extraordinary ideas.

I would like to thank Dr. Jean-Olivier Plouchart and Dr. Jonghae Kim from IBM’s Microelectronics Semiconductor Research and Development Centre (SRDC) in Hopewell Junction, NY, USA. It was a pleasure working for these gentlemen during my 6 month internship, which allowed me to access an advanced SOI CMOS technology and to assist in the development of a new electronic design automation tool.

To my electrical engineering comrades in the department, John Danson, Igor Miletic, and my younger brother Vincent Karam, thank you for your good and sometimes not-so-good advice – the intention is what counts.

The financial support of the Carleton University, Natural Sciences and Engineering Research Council of Canada (NSERC) and Micronet is gratefully acknowledged.

Finally, my fondest regards go to my parents, Phillip and Fadia Karam. Mom and Dad gave me the opportunity to attend Carleton University, encouraged me in my studies, and gave me the support that I needed to finish my thesis.
# Table of Contents

Abstract ........................................................................................................................................... ii

Acknowledgements ......................................................................................................................... iii

Table of Contents ............................................................................................................................... iv

List of Tables ....................................................................................................................................... vii

List of Figures and Illustrations ......................................................................................................... viii

List of Symbols, Abbreviations and Nomenclature ........................................................................... x

1. Introduction ..................................................................................................................................... 1

1.1. Why Synthesis? ....................................................................................................................... 1

1.2. Motivation ............................................................................................................................... 2

1.3. Contributions ......................................................................................................................... 4

1.4. Thesis Outline ....................................................................................................................... 5

2. RF Mixed-Signal IC Design ........................................................................................................... 7

2.1. RF Mixed-Signal Design Flow ............................................................................................... 7

2.1.1. Layout Generation ........................................................................................................... 9

2.2. RF Mixed-Signal Layout Synthesis ....................................................................................... 10

2.2.1. Procedural Module EDA ............................................................................................. 10

2.2.2. Template EDA .................................................................................................................. 11

2.2.3. Macro-cell Place-and-Route EDA .................................................................................. 11

2.3. Summary ............................................................................................................................... 13
3. A New RF Mixed-Signal Design Methodology................................. 14

3.1. Object-oriented C++: Introduction and Concepts........................... 14

3.2. A New IBM EDA Layout Synthesis Tool........................................ 16

3.2.1. GDSText Generator............................................................. 16

3.2.2. GDSII Translator............................................................... 19

3.3. Layout Synthesis of an nFET.................................................. 19

3.4. New RF Mixed-Signal Design Flow........................................... 22

3.5. Summary................................................................................. 26

4. Ring Oscillator............................................................................. 27

4.1. A 90nm SOI CMOS Technology................................................ 29

4.1.1. Interconnects......................................................................... 31

4.1.2. Diffusion Pitches................................................................. 32

4.1.3. Unity Current Gain Frequency and Unity Power Gain Frequency... 34

4.2. CMOS Inverter.......................................................................... 36

4.2.1. Propagation Delay – Inverter................................................. 37

4.2.2. Unity Current Gain Frequency and Unity Power Gain Frequency... 41

4.2.3. Propagation Delay - Ring..................................................... 44

4.3. Ring Oscillator.......................................................................... 52

4.4. Simulation................................................................................. 54

4.4.1. Unity Current Gain Frequency and Unity Power Gain Frequency... 55

4.4.2. Propagation Delay............................................................... 56

4.4.3. Observations........................................................................ 57
List of Tables

Table 4-1 Relevant FET Capacitances for Determining Delay ............................................ 49
Table 4-2 FET Finger Width and Total Device Width for the Ring Netlists ......................... 53
Table 4-3 Inverter Device $W_f$ and $f_{T, \text{inv}}$ ............................................................... 55
Table 4-4 Inverter Device $W_f$ and $f_{\text{max, inv}}$ ............................................................... 55
Table 4-5 Ring Device $W_f$ and $t_{pd, \text{inv, chain}}$ ............................................................... 57
Table 4-6 Relevant Inverter Chain Wiring Parasitics .......................................................... 64
Table 4-7 Extracted Inverter Device $W_f$ and $f_{T, \text{inv}}$ ....................................................... 65
Table 4-8 Extracted Inverter Device $W_f$ and $f_{\text{max, inv}}$ ....................................................... 65
Table 4-9 Extracted Ring Device $W_f$ and $t_{pd, \text{inv, chain}}$ ...................................................... 66
Table 4-10 Percentage Difference from Optimum Extracted Inverter Delay ....................... 67
Table 4-11 Ring's Dominant Time Constants for High-to-Low Propagation Delay ............... 68
Table 4-12 Device Capacitances $C_{IN}$ and $C_{OUT}$ Over the H-to-L Delay Time Interval .... 73
List of Figures and Illustrations

Figure 2-1 RF Mixed-Signal Design Flow .......................................................... 9
Figure 2-2 Layout Synthesis Tools ..................................................................... 13
Figure 3-1 OOP Inheritance Example............................................................... 16
Figure 3-2 Class Hierarchy for Devices.............................................................. 17
Figure 3-3 GDSText Generator Tool Flow Chart .................................................. 20
Figure 3-4 GDSII Translator Flow Chart ............................................................ 21
Figure 3-5 Layout-based RF Mixed-Signal Design Flow ....................................... 25
Figure 4-1 An 11-Inverter Stage Ring Oscillator ................................................ 28
Figure 4-2 Cross-Section Device Structure of a SOI nFET .................................... 31
Figure 4-3 Interconnect Sidewall Coupling Parasitic ......................................... 32
Figure 4-4 90nm SOI FET Pitch Sizes ............................................................... 33
Figure 4-5 MOSFET Small-Signal Hybrid-pi model ............................................. 34
Figure 4-6 CMOS Inverter .................................................................................. 36
Figure 4-7 Inverter Voltage Transfer Function ................................................... 37
Figure 4-8 CMOS Inverter Static Switch Model Representation .......................... 38
Figure 4-9 Input and Output Signals of a CMOS Inverter Switch Model ............... 39
Figure 4-10 Equivalent Small Signal Model of CMOS Inverter............................. 41
Figure 4-11 Two Stage Inverter Chain – Switch Model ......................................... 45
Figure 4-12 Average nFET On-Resistance .......................................................... 46
Figure 4-13 Inverter VTC with FET Operation Regions ......................................... 48
Figure 4-14 Two Stage Inverter Chain – Parasitic Capacitances ............................. 49
## List of Symbols, Abbreviations and Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5G</td>
<td>GSM enhanced with General Packet Radio Services technology</td>
</tr>
<tr>
<td>3G</td>
<td>3rd Generation GSM</td>
</tr>
<tr>
<td>A</td>
<td>area [m]</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>C</td>
<td>capacitor</td>
</tr>
<tr>
<td>C_{bd}</td>
<td>body-to-drain capacitance [F]</td>
</tr>
<tr>
<td>C_{bd,n}</td>
<td>nFET body-to-drain capacitance [F]</td>
</tr>
<tr>
<td>C_{bd,p}</td>
<td>pFET body-to-drain capacitance [F]</td>
</tr>
<tr>
<td>C_g</td>
<td>equivalent inverter input capacitance [F]</td>
</tr>
<tr>
<td>C_{gd}</td>
<td>gate-to-drain capacitance [F]</td>
</tr>
<tr>
<td>C_{gd,n}</td>
<td>nFET gate-to-drain capacitance [F]</td>
</tr>
<tr>
<td>C_{gd,p}</td>
<td>pFET gate-to-drain capacitance [F]</td>
</tr>
<tr>
<td>C_gS</td>
<td>gate-to-source capacitance [F]</td>
</tr>
<tr>
<td>C_{gs,n}</td>
<td>nFET gate-to-source capacitance [F]</td>
</tr>
<tr>
<td>C_{gs,p}</td>
<td>pFET gate-to-source capacitance [F]</td>
</tr>
<tr>
<td>C_{IN}</td>
<td>inverter input capacitance [F]</td>
</tr>
<tr>
<td>C_{OUT}</td>
<td>inverter output capacitance [F]</td>
</tr>
<tr>
<td>C_{ox}</td>
<td>gate-oxide capacitance per unit area [F/m²]</td>
</tr>
<tr>
<td>C_p</td>
<td>coupling capacitance [F]</td>
</tr>
<tr>
<td>C_w</td>
<td>wiring parasitic capacitance [F]</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal-Oxide-Semiconductor</td>
</tr>
<tr>
<td>diffL</td>
<td>FET left diffusion pitch</td>
</tr>
<tr>
<td>diffM</td>
<td>FET middle diffusion pitch</td>
</tr>
<tr>
<td>diffR</td>
<td>FET right diffusion pitch</td>
</tr>
<tr>
<td>DRC</td>
<td>Design-Rule-Check</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>effective permittivity [F/m]</td>
</tr>
<tr>
<td>EDA</td>
<td>Electronic Design Automation</td>
</tr>
<tr>
<td>f_{max}</td>
<td>unity power gain frequency [Hz]</td>
</tr>
<tr>
<td>f_{MAX_inv}</td>
<td>unity power gain frequency [Hz]</td>
</tr>
<tr>
<td>f_{osc}</td>
<td>oscillation frequency [Hz]</td>
</tr>
<tr>
<td>f_{T}</td>
<td>unity current gain frequency [Hz]</td>
</tr>
<tr>
<td>f_{T_inv}</td>
<td>inverter unity current gain frequency [Hz]</td>
</tr>
<tr>
<td>FET</td>
<td>Field Effect Transistor</td>
</tr>
<tr>
<td>g_d</td>
<td>output conductance [mhos]</td>
</tr>
<tr>
<td>g_{d,n}</td>
<td>nFET output conductance [mhos]</td>
</tr>
<tr>
<td>g_{d,p}</td>
<td>pFET output conductance [mhos]</td>
</tr>
<tr>
<td>GDSII</td>
<td>Graphic Design System II</td>
</tr>
<tr>
<td>g_m</td>
<td>transconductance [S]</td>
</tr>
<tr>
<td>g_{m,n}</td>
<td>nFET transconductance [S]</td>
</tr>
<tr>
<td>g_{m,p}</td>
<td>pFET transconductance [S]</td>
</tr>
<tr>
<td>GND</td>
<td>ground [0V]</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>HDL</td>
<td>Hardware Description Languages</td>
</tr>
<tr>
<td>I</td>
<td>current [A]</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
<tr>
<td>I_{DS}</td>
<td>drain-to-source current [A]</td>
</tr>
<tr>
<td>I_{DS,n}</td>
<td>nFET drain-to-source current [A]</td>
</tr>
<tr>
<td>L</td>
<td>MOS gate length [nm]</td>
</tr>
<tr>
<td>LVS</td>
<td>Layout-Versus-Schematic</td>
</tr>
<tr>
<td>MOS</td>
<td>Metal-Oxide-Semiconductor</td>
</tr>
<tr>
<td>MOSFET</td>
<td>Metal-Oxide-Semiconductor Field Effect Transistor</td>
</tr>
<tr>
<td>MS</td>
<td>Mixed-Signal</td>
</tr>
<tr>
<td>N</td>
<td>number of inverter stages</td>
</tr>
<tr>
<td>nFET</td>
<td>n-channel Field Effect Transistor</td>
</tr>
<tr>
<td>NM_{H}</td>
<td>high logic level noise margin [V]</td>
</tr>
<tr>
<td>NM_{L}</td>
<td>low logic level noise margin [V]</td>
</tr>
</tbody>
</table>
OOP  
Object-Oriented Programming

P-cells  
Parameterized Cells

PAN  
Processing Area Network

pFET  
p-channel Field Effect Transistor

R  
resistor

$R_g$  
gate resistance [Ω]

$R_{g,n}$  
nFET gate resistance [Ω]

$R_{g,p}$  
pFET gate resistance [Ω]

$R_{g, np}$  
equivalent inverter input resistance [Ω]

$R_{\text{drain}}$  
drain connection resistance [Ω]

$R_{\text{on,n}}$  
nFET on-resistance [Ω]

$R'_{\text{on,n}}$  
nFET on-resistance per square [Ω]

$R_{\text{on,p}}$  
pFET on-resistance [Ω]

$R'_{\text{on,p}}$  
nFET on-resistance per square [Ω]

$R_{\text{sheet}}$  
MOS gate sheet resistance [Ω/square]

$R_{\text{strap}}$  
poly gate strap resistance [Ω]

$R_W$  
wiring parasitic resistance [Ω]

RC  
resistor-capacitor network

RF  
Radio Frequency

Si  
Silicon

SiO$_2$  
Silicon Dioxide

SOI  
Silicon-on-Insulator

STI  
Shallow Trench Isolation

t  
time [s]

$\tau_{ED}$  
Elmore delay [s]

$t_p$  
average ring propagation delay [s]

$t_{pd\_wire}$  
distributed propagation delay for a wire [s]

$t_{pd\_inv\_chain}$  
average propagation delay for an inverter in a chain [s]

$t_{\text{PHL}}$  
propagation delay when the output goes from high-to-low [s]

$t_{\text{PHL\_inv\_chain}}$  
high-to-low propagation delay for an inverter in a chain [s]

$t_{\text{PLH}}$  
propagation delay when the output goes from low-to-high [s]
\( t_{pLH\_inv\_chain} \) | low-to-high propagation delay for an inverter in a chain [s]
---|---
\( \mu_n \) | nFET carrier mobility [m/s/J]
\( V_{DD} \) | supply voltage [V]
\( V_{DS} \) | drain-to-source voltage [V]
\( V_{gs} \) | small-signal gate-to-source voltage [V]
\( V_{GS} \) | gate-to-source voltage [V]
\( V_{III} \) | input voltage when slope of VTC equals 1 for logic high [V]
\( V_{IL} \) | input voltage when slope of VTC equals 1 for logic low [V]
\( V_{IN} \) | input voltage [V]
\( V_M \) | inverter switching threshold [V]
\( V_{OH} \) | maximum output high logic voltage [V]
\( V_{OL} \) | minimum output low logic voltage [V]
\( V_{OUT} \) | output voltage [V]
\( V_t \) | threshold voltage [V]
\( \text{VHDL} \) | VHSIC Hardware Description Language
\( \text{VHSIC} \) | Very High-Speed Integrated Circuit
\( \text{VLSI} \) | Very Large Scale Integration
\( \text{VTC} \) | Voltage Transfer Characteristic
\( W \) | MOS gate width [m]
\( W_f \) | MOS gate finger width [m]
\( W_n \) | nFET Device width [m]
\( W_p \) | pFET Device width [m]
\( \text{WLAN} \) | Wireless Local Area Network
1. Introduction

In this thesis, a new top-down methodology for radio frequency (RF) mixed-signal (MS) design is developed and the related electronic design automation (EDA) tool for layout synthesis is presented. The synthesis methodology is demonstrated by the design and performance optimization of a ring oscillator circuit in a nanometre silicon-on-insulator (SOI) CMOS process.

1.1. Why Synthesis?

Synthesis is a means of enabling a designer to automate integrated circuit (IC) design based on a set of rules. As ICs were being engineered to accommodate a large number of transistors (from hundreds of thousands up to millions) enabling the circuit to perform more functions, software tools, known as electronic design automation (EDA) tools, were created to aid the designer in the development of these very large scale integration (VLSI) circuits. First appearing in the late 1970s, EDA synthesis tools began to significantly improve the efficiency of the physical layout design by automatically placing and routing digital gates for minimizing chip area or maximizing chip speed. Then, in the late 1980s, hardware description languages (HDLs) were introduced, such as VHDL and Verilog, giving the designer the ability to model a digital system at a behavioural-level or gate-level abstraction. This allowed for high-level synthesis EDA tools which would synthesize the behavioural HDL description into a gate-level representation. Logic syn-
thesis EDA tools were then used to translate this gate-level representation into a gate-level netlist. The advent of these advances in synthesis allowed design productivity to keep up with the increasing complexity of VLSI coupled with the shortening of design periods from time-to-market pressures. [1]

1.2. Motivation

Mixed-signal circuits are needed to interface the continuous-valued analog circuits with the digital signal processing (DSP) circuits of a system. Common mixed-signal circuits are sample-and-hold circuits for sampling of analog signals, analog-to-digital converters for quantization of analog signals, digital-to-analog converters for reconstruction of amplitude signals, and phase-locked loops for frequency synthesis or timing synchronization purposes. [2]

The constant growing market for wireless networking and communication systems is providing the impetus for the semiconductor market in the first decade of the 2000s. Wireless transmissions contain significant amounts of voice and sound content, requiring IC designs to include an ever-increasing amount of mixed-signal circuitry. It is estimated that by 2005, 67% of IC design will be in mixed-signal, up from 22% in 1999 [3]. Furthermore, evolving wireless systems for 2.5G and 3G cellular, Wireless Local Area Network (WLAN) and Processing Area Network (PAN) are moving from “voice-centric” to “voice + internet” and “voice + multimedia” capable wireless terminals [4]. This entails higher data rates for voice and high-bandwidth multimedia, demanding an
increase in the signal processing speeds and requiring the system's mixed-signal circuitry to operate at RF.

The market demands for performance, integration, and cost reduction have driven the semiconductor industry to continually move to the newest CMOS technology. With 90nm gate-length CMOS designs in production, and 65nm in the near future, designers are contending with deep submicron factors which could critically impact the success of mixed-signal designs: shrinking feature sizes, thinner line widths and more metal layers [5]. These factors, along with the ever-increasing RF mixed-signal content, are making the design performance more dependent on the physical layout because of the resulting parasitic effects.

Designers of high frequency mixed-signal (and RF) circuits are lacking the support of EDA tools available to both digital and low frequency mixed-signal circuits. Time-to-market pressures on semiconductor design centres and the increasing content and complexity of RF mixed-signal wireless systems are demanding a new design methodology to decrease design cycle times [6]. To achieve this, the design methodology must be layout-centric, as the RF mixed-signal layout implementation is one of the most time consuming steps in the design cycle and the most sensitive aspect of performance. This entails that the new methodology must employ an EDA tool capable of synthesizing RF mixed-signal layouts. Furthermore, design reuse is another means accessible to digital and low frequency mixed-signal designers to shrink development cycles. Therefore, as
1. Introduction
technologies advance to the next process node, this methodology should allow for direct technology transfer of layout designs with minimal porting or migration effort.

The benefits of this new methodology are not limited to semiconductor design centres, but also to semiconductor fabrication facilities for technology benchmarking. As CMOS processes evolve towards the 65nm node, monitoring an immature technology’s yield and performance becomes crucial for development. An EDA layout synthesis tool can aid in the optimization of benchmark circuits for accurate and efficient performance evaluation of a technology’s full potential.

1.3. Contributions

The contributions of this work are as follows:

Methodology:

- worked with IBM in the formulation of a new RF mixed-signal design methodology based on layout synthesis;
- demonstration, verification and application of the new methodology by the design and optimization of a ring oscillator for technology benchmarking.

Electronic Design Automation:

- demonstration of a EDA tool for RF mixed-signal layout synthesis in 90nm SOI CMOS – the GDSText Generator with the GDSII Translator, and the
1. Introduction

EDA development in this thesis is for both academic research and industrial applications.

Inverter (basic building block of the ring oscillator):

• theory and analysis of unity current gain frequency and unity power gain frequency for a CMOS inverter;

• development of the EDA tool’s architecture database for the synthesis of a lithography-independent and hierarchal CMOS inverter layout.

Ring Oscillator (methodology demonstration):

• theory and analysis of ring propagation delay in the presence of wiring parasitics;

• development of the EDA tool’s architecture database for the synthesis of a lithography-independent and hierarchal 11-stage ring oscillator layout;

• design and optimization of an 1V 11-stage ring oscillator circuit in a 90nm SOI CMOS for maximum oscillation frequency by using the methodology.

1.4. Thesis Outline

There are five chapters in this thesis, and they are organized as follows:

Chapter 1 is the introduction for this thesis.
Chapter 2 gives a brief overview of the current RF IC mixed-signal design methodology and a background of existing automatic layout generation tools for RF mixed-signal circuits.

Chapter 3 discloses a new RF IC mixed-signal design methodology. This new methodology employs a new IBM electronic design automation tool (EDA) for layout generation – the GDSText Generator with the GDSII Translator. A description of this synthesis tool, highlighting its features and capabilities, and its application in a new RF mixed-signal design flow are presented.

Chapter 4 contains a demonstration of the new methodology with the design of an 11-stage ring oscillator circuit in a 90nm SOI CMOS technology. The ring is optimized for maximum oscillation frequency with the automatic layout generation tool presented in Chapter 3. Related theories such as inverter unity current gain frequency, inverter unity power gain frequency and ring oscillator propagation delay are analyzed and discussed.

Chapter 5 provides a summary, concluding remarks and suggestions of future research work.
2. RF Mixed-Signal IC Design

In the previous chapter, the case was made for a new design methodology, employing an EDA layout synthesis tool, to address the growing market and nanometre-technology implementation challenges for RF mixed-signal IC design. This chapter provides an overview of a common RF mixed-signal design flow, and presents some of the existing EDA tools for automatic layout generation, as well as their shortcomings.

2.1. RF Mixed-Signal Design Flow

The possible design flow for mixed-signal IC operating at RF is illustrated in Figure 2-1. The flow is based on a top-down performance-driven design methodology [2]. At each stage, the required task is performed while satisfying the target specifications for the design. If the specifications cannot be met, a redesign or backtracking iteration may be needed to a point in the flow where the design can be corrected. The closer this point is to the current stage, the less time it takes for the redesign – usually, but not always. One of the major differences between a digital and a RF mixed-signal design flow is that the RF mixed signal design flow is expected to have a much higher number of design iterations as it is more heuristic. A summary of the tasks performed by each stage is described in the following paragraphs.

The first stage in the design process, topology selection, is where the most suitable circuit topology is chosen from a set of known architectures to meet the target speci-
fications. If none of the existing architectures are adequate, there is the need for innovation and the designer can derive a novel topology.

The next stage is optimization by simulation. Here, the designer uses a CAD tool that accesses the fabrication technology's library for device models to optimize the circuit's netlist (or schematic) for the performance specifications, yield and environmental conditions. In this stage, all of the device sizes or values and bias parameters in the circuit are determined.

Layout generation is the third stage of the design flow, where the designer, with the help of a CAD tool, creates a mask-level layout from the optimized circuit's netlist. During this stage, the designer is insuring that the layout occupies a minimum amount of chip area.

The fourth stage is verification by design-rule-check (DRC) and layout-versus-schematic (LVS). The designer uses a CAD tool to verify that the dimensions, spacing, overlaps etc. of the layout conform to the design rules specified by the semiconductor fabrication facility. Then, another CAD tool is used to extract a netlist, containing the intentional devices and unintentional parasitics, from the layout. This extracted netlist is compared against the original circuit's netlist for verification.

The last stage in the design flow is post-layout optimization by simulation, where the designer use a CAD tool that accesses the technology's library to verify that the extracted netlist, with its parasitics, meets the target performance specifications, yield and environmental conditions.
2.1.1. Layout Generation

Layout generation is one of the most time consuming stages in an RF mixed-signal design flow as the designer needs to understand and address the critical wiring parasitics which could affect the circuit’s performance. Typically, in order to increase
the quantity, quality and speed of performing this low-level implementation, a semiconductor design centre's only solution would be to hire more layout designers. This is also true for a semiconductor fabrication facility doing technology benchmarking. However, there have been developments in EDA tools for the automatic generation of RF mixed-signal layouts, and some of these tools are described in the following section.

2.2. RF Mixed-Signal Layout Synthesis

There are two main approaches for automatic layout generation of RF mixed-signal circuits. The first approach is knowledge-based, which relies on mixed-signal designer knowledge to output a solution. This approach uses EDA tools which generate layouts by procedural modules or templates. The second approach is optimization-based, which uses computers to derive a solution according to some constraints. This approach uses EDA tools which generate layouts by macrocell-place-and-route. [2]

2.2.1. Procedural Module EDA

The procedural module EDA tool, presented in [7] for analog cell layout synthesis, is a software program that contains a coded definition of the cell's layout. At runtime, the program instantaneously generates the layout based on the values of the input parameters (e.g. device sizes). This approach is used frequently for layout generation of single devices. However, for the automatic layout generation of entire circuits, this approach may result in inefficient use of area for large device parameter variations in the circuit. The disadvantage with procedural module tool is that it must be updated for any
changes in the technology process, requiring a semiconductor design center to invest precious development time and resources. [2]

2.2.2. Template EDA

The template EDA tool for layout generation is more circuit-oriented. This approach uses a geometric template that defines the relative positions and interconnections of the devices in the circuit [2]. When the input circuit parameters are entered, the tool uses the template to place the devices such that an area efficient layout is generated. Parameterized layout templates of single devices have been developed. The most widely used in industry are Virtuoso Parameterized Cells (P-cells) created with the SKILL language [8], and are frequently included in the design kits from semiconductor manufacturers. Unfortunately, the construction of layout templates requires more time and effort than a fully-custom layout for a given circuit design, and the developmental cost is a factor of five times more expensive [9]. Therefore, repetitive use of the layout template of more than five times is required for this approach to be an economically viable choice. Lastly, like the procedural module approach, the layout information contained in the tool has to be maintained as the process’s design rules change.

2.2.3. Macro-cell Place-and-Route EDA

The optimization-based macro-cell place-and-route approach [2] was developed to generate fully-customized layouts. This approach uses a procedural module EDA tool to generate layout cells (called macros) of devices. Then, a placement tool optimally ar-
ranges the cells, a routing tool establishes the interconnections between the cells and a compaction tool condenses the cells together to minimize the area, generating a layout of an entire circuit. A flow for this approach is illustrated in Figure 2-2. The EDA tool KOAN [10] is a placement tool that performs qualitative optimization by maximizing drain-source merging through folding, abutting, and/or merging of MOS devices. However, for each variant of these grouped devices, such as stacked, interdigitated, etc., procedural module generators have to be developed and maintained. A router tool that qualitatively optimizes for minimal crosstalk is ANAGRAM II [11]. Advances in the development of optimization-based macro-cell place-and-route tools allowed for the generation of layouts which are optimized for explicit quantitative performance targets. In this method, the degradation of performance from layout parasitics is measured to ensure the design's specifications are within its tolerances after the layout implementation. The placement tool PUPPY-A [12], the routing tool ROAD [13], and the compaction tool SPARCS-A [14] are all tools of this type. A combined placement, routing and compaction tool called NEOCELL [15] is also of this type. The main drawback with the optimization-based approach is the high computation cost, and the average semiconductor design centre usually cannot afford the overhead.
2.3. Summary

RF mixed-signal design can be characterized by a top-down performance-driven design process, where EDA tools were developed to help automate the layout generation stage. There are three main types of EDA tools; procedural module, template and macro-cell-place-and-route. However, none of these have been widely accepted by the IC design industry because of the resources and time needed to support them.
3. A New RF Mixed-Signal Design Methodology

In nanometre scale design, the new physical effects associated with shrinking geometries are making the performance of RF mixed-signal circuits more dependent on the layout phase of the design process. According to an IC/ASIC study [16], more than half of analog mixed-signal designs fail to pass first silicon - functional flaws in the design being the main cause for failures. It is becoming more apparent to designers that a new design methodology is needed to address the manufacturability of their designs. Furthermore, they demand a methodology geared for a more efficient design process because of the ever-decreasing consumer product life-cycle market place. To satisfy their concerns, a new RF mixed-signal design methodology is developed and the related layout synthesis tool from IBM is presented. The synthesis tool is incorporated in a new layout-based design flow for RF mixed-signal ICs.

3.1. Object-oriented C++: Introduction and Concepts

The programming language C++ was chosen to implement the new layout synthesis tool because of its object-oriented nature - features and abilities. Object-oriented programming (OOP) [17] allows the programmer to create data structures - objects which have state and behaviour characteristics. The state of an object is defined by variables called attributes, and its behaviour is implemented with functions called methods. A program contains a collection of many objects which interact (i.e. communicate information) with one-another through the passing of parameters.
A programmer creates a software template, known as a class, to define the attributes and methods common to objects of a certain kind. Objects of the same kind are instances of a particular class. In addition, object-oriented programming, allows for the defining of classes in terms of other classes, thereby reusing existing code and decreasing development time. This feature is known as inheritance. For instance, if there is a class called “vehicle”, the programmer can define a class called “car” which will inherit the attributes and methods of class vehicle without the need to redefine them. Vehicle is the parent, known as a superclass, and car is its child, known as a subclass. Subclasses can override the attributes and methods of its superclass and can define new attributes and methods to specialize its state and behaviour. Further specialization is achieved by defining a subclass of the class car, called Ford, which will inherit everything defined for class car, as well as class vehicle. Now, class car will be the parent of the child class Ford. This relationship is shown in Figure 3-1. Finally, objects of the subclass can be used where objects of the corresponding superclass are expected.
3.2. A New IBM EDA Layout Synthesis Tool

A newly developed object-oriented C++-based layout synthesis tool from IBM, the GDSText Generator with the GDSII Translator, is presented for a nanometre SOI CMOS technology process.

3.2.1. GDSText Generator

The GDSText Generator tool is used to create data structures, implemented as C++ objects, which represent the physical layouts of devices. The blueprints for the data structures are derived from a class hierarchy of devices. The superclass to which all devices inherit from is the device_base class. To give its objects state and behaviour characteristics, the class device_base has attributes such as xy-location and pointers to an arbitrary number of terminals for wiring interconnects, and methods such as orthogonal an-
A class for a device is not limited to containing only layer information pertaining to the device's layout structure, but can also specify any interconnect layers on the struc-
A New RF Mixed-Signal Design Methodology

ture's terminals for routing purposes. For example, the nFET class can have a metal layer option attribute which would pre-wire the gate, drain, source and bulk terminals of its objects with a specified metal layer(s).

With the xy-location attribute inherited from the device_base class, final placement information, if known, can be specified for the C++ objects of the GDSText Generator to create a layout of an entire circuit.

The nFET (and pFET) subclass also defines output-write functions to enable its objects to communicate their layout attributes to the outside world. The layout information is sent out in an IBM developed ASCII language syntax called GDSText. With GDSText, C++ objects can describe the physical structures in a GDSII-binary format layout. For instance, the gate of an nFET layout would be described in GDSText as a rectangle-type polygon, having a poly-type layer with a drawing-type purpose, a coordinate position and a size dimension. Further information about the hierarchal position of this nFET in a layout containing subcircuits could be specified with GDSText. Also, GDSText can describe any text-shapes in the GDSII layout.

As it is necessary to use the newest technology for designing because of the advantages in performance, integration and cost reduction, GDSText Generator's classes risk becoming obsolete as CMOS production moves from the 90nm to the 65nm process node. However, since the layer masks for the device are not changing with technology, just their relative positioning and sizing between each other, the classes' write methods are coded to access relevant design rules from a database for a given technology. In this
3. A New RF Mixed-Signal Design Methodology

way, the write methods do not need to be updated for technology node migration and the resulting GDSText layout output will always lead to DRC clean layouts.

3.2.2. GDSII Translator

The GDSII Translator tool is used to translate to-and-from GDSText-ASCII format and GDSII-binary format. When a GDSText Generator’s data structure outputs its layout information to a GDSText-ASCII file (*.txt), it can be converted to a machine-readable GDSII-binary file (*.gds), and vice versa.

3.3. Layout Synthesis of an nFET

An example of GDSText Generator and GDSII Translator usage is illustrated in this section with the synthesis of a floating-body, partially-depleted, regular Vt nFET layout for a 90nm SOI CMOS process.

With a class definition for the nFET defined (in a GDSText class architecture database called devices.cpp), the GDSText Generator creates an nFET object. The user then sets the nFET’s attributes with the following design: a width of 3μm, a gate length of 90nm, a number of fingers of 2, right, middle and left diffusion pitches of 0.2975μm, 0.2275μm and 0.2975μm respectively, an xy co-ordinate location of 0,0, and a prewiring option of metal-1 for the gate and source terminals and of up to metal-2 for the drain terminal. Then, the nFET object’s write function is called, and then the program generates an output file (nfet.txt) containing the nFET’s layout information in the GDSText-ASCII format.
syntax. The GDSText Generator tool’s usage is illustrated with the flow chart in Figure 3-3.

The GDSText-ASCII output file (nFET.txt) is then imported into the GDSII Translator which converts the GDSText that describes nFET layout structure into a

---

**Figure 3-3 GDSText Generator Tool Flow Chart**

The GDSText-ASCII output file (nFET.txt) is then imported into the GDSII Translator which converts the GDSText that describes nFET layout structure into a
GDSII-binary format layout file (nFET.gds). The GDSII Translator tool’s usage is illustrated with the flow chart in Figure 3-4.

nFET.txt (GDSText input)

// RX rectangular diagonal axis (x1,y1) and (sx1,sy1)
RECT Layer=(RX,Drawing) xy=(0,0) sz=(9.3,10)

// PC rect diagonal array axes (xn,yn) and (sxn,syn)
RECT Layer=(PC,Drawing) xy=(1.7,0) sz=(0.5,10)
RECT Layer=(PC,Drawing) xy=(3.5,0) sz=(0.5,10)
...

GDSII Translator

nFET.gds (GDSII output)

Figure 3-4 GDSII Translator Flow Chart
3.4. New RF Mixed-Signal Design Flow

The GDSText Generator and the GDSII Translator form a new IBM EDA tool capable of synthesizing layouts based on a technology’s design rules. This EDA tool is incorporated and developed into a new design flow for mixed-signal IC operating at RF and is illustrated in Figure 3-5. The design flow is still based on a top-down performance-driven design methodology, with backtracking paths for redesign corrections. However, unlike the typical RF mixed-signal design flow presented in Chapter 2, the new EDA synthesis tool allows for a more layout-centric design flow, as proper and efficient layout optimization is critical for the IC’s performance and economic viability. A summary of the tasks performed by each stage is described as follows.

Like the traditional RF mixed-signal design flow, the first stage is to select a topology that is most suited to meet the specification requirements based on designer-knowledge.

The next stage is to use a simulation CAD tool, such as SPICE or SPECTRE, which accesses the technology’s model library, to optimize the circuit’s netlist for the performance specifications, yield and environmental conditions. Failing this, another topology can be chosen by the designer.

The third stage is to use the new IBM EDA tool for layout synthesis. If the circuit’s topology exists in the GDSText class architecture database, the designer can execute the GDSText Generator program, create an instance of the topology’s class (i.e. an object), enter the necessary information from the netlist optimization stage to set the ob-
ject attributes, and call the object's write method to have its layout information generated in the GDSText-ASCII syntax and saved to a file. However, if the circuit's topology does not exist in the GDSText class architecture database, then a class definition for the circuit has to be coded. The coding for this class can use any existing class architectures in the database, such as nFET, pFET, etc., to accelerate the development. Once the coding is done, the GDSText Generator is re-compiled and then executed as before.

The forth stage in the flow is to convert the file containing the circuit's layout information in GDSText-ASCII into a GDSII-binary format layout file using the GDSII Translator tool.

With the generated layout, the next stage is to use an existing layout CAD tool, such as Virtuoso, to do any necessary final placement and routing for the circuit's devices.

The sixth stage, like in the traditional RF mixed-signal design flow, is to use a DRC and LVS verification CAD tool, such as Mentor Graphics' Calibre. If the layout does not conform to the design rules or the extracted netlist is different from the circuit's original netlist, then the circuit's layout can be re-worked with a CAD tool or the circuit's topology can be re-coded for the GDSText Generator and then re-converted into a GDSII layout.

The last stage is to perform post-layout optimization simulations where the designer uses a CAD tool that accesses the technology's model library to verify the extracted netlist meets the target performance specifications, yield and environmental con-
ditions. If it does not, the circuit's layout can be re-synthesized with IBM's EDA tool, re-worked with a layout CAD tool or its netlist topology can be changed to one that is better suited to meet the required specifications.
3. A New RF Mixed-Signal Design Methodology

Figure 3-5 Layout-based RF Mixed-Signal Design Flow

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.5. Summary

This new IBM EDA synthesis tool is of the knowledge-based approach, in that it relies on designer knowledge for the placement and optional routing of devices to generate RF mixed-signal IC layouts. Also, this EDA tool is of the procedural module-type because the layout is generated by a program at run-time based on device input parameters. However, since the technology design rules are accessed at run-time by this tool, technology process changes can easily be made without having to re-code the layout classes for the GDSText Generator. Unlike the other procedural module tool presented in Chapter 2, this new EDA tool requires minimal time and resources to maintain. Also, since the GDSText Generator is based on object-oriented programming, code-reuse through inheritance will decrease development time of new layout data structures.

The contributions discussed in this thesis includes an overall development of this EDA tool, however, the main focus was on layout design and synthesis of the tool’s class architecture database which is shown in blue on Figure 3-5.

This new RF mixed-signal design methodology will be easily accepted by RF mixed-signal designers. Aside from its layout-centric flow and potential increase in design cycle efficiency, the main reason for this is because designers will be already familiar with significant parts of the design process as it relies on existing methods and industry CAD tools for simulation and verification. A demonstration of the new methodology is shown in the following chapter.
4. Ring Oscillator

Until recently, communication ICs operating in the gigahertz range were largely implemented with very fast technologies. Emerging as an attractive alternative is nanometre CMOS because of the impressive unity current gain frequency \( (f_T) \) and unity power gain frequency \( (f_{max}) \) performance now achievable by the transistors in this process in addition to its integration advantages. IBM’s 90nm SOI CMOS is one such process, as this technology’s partially-depleted nFETs achieve an \( f_{max} \) of 208GHz and a peak \( f_T \) of 243GHz (and pFETs achieve a peak \( f_T \) of 108GHz) [18]. SOI CMOS is able to provide further speed, as well as power, enhancements to digital and mixed-signal circuits over bulk-CMOS technologies mainly because SOI FETs have lower parasitic junction capacitances.

Propagation delay, defined in the general sense as the time required for the output of a logic gate to respond to a combination of input pulses, is another important benchmark that semiconductor foundries use to evaluate their technology because it directly impacts on the speed at which a digital or mixed-signal circuit can operate. The factors that contribute to logic circuit propagation delay are wiring resistance and capacitance, fan-out resistance and capacitance, and the on-resistance of the transistor [19]. Wiring resistance arises from the sheet resistance of interconnects between gates, and wiring capacitance is due to the capacitive coupling between interconnects and the substrate. Fan-out resistance and capacitance are from the input impedance of the device(s) loading the
driving gate’s output. And lastly, the on-resistance is the effective resistance between drain and source terminals of the transistor, and it is a non-linear function of the voltage across the FET.

The most common method to simulate or measure propagation delay is to measure it from the oscillation frequency of a ring oscillator. A ring oscillator is composed of an odd number of inverter cells chained together, with the output of the last inverter connected to the input of the first inverter, thus forming a ring. An illustration of a ring oscillator composed of 11 inverter stages is shown in Figure 4-1.

![Figure 4-1 An 11-Inverter Stage Ring Oscillator](image)

The ring will oscillate at a frequency determined by the average propagation delay of each inverter cell. From a transient simulation or time-domain lab measurement of the ring, the average propagation delay $t_p$ of each inverter cell will be given by

$$t_p = \frac{1}{2Nf_{osc}}$$  (4-1)

Where $N$ is the number of inverter cells (or stages), and $f_{osc}$ is the oscillation frequency.
4. Ring Oscillator

Ring oscillators are used in mixed-signal circuits such as phase-lock loops for frequency synthesis or timing synchronization, and these circuits are beginning to operate in the RF spectrum [20], making the layout design an increasingly important factor in the ring's performance. Furthermore, and most importantly, ring oscillators are used to benchmark a given technology. Benchmarking is used by semiconductor fabrication facilities to monitor a developing technology's yield and performance. Proper evaluation of a new technology's potential is crucial for a lower time-to-market of the technology and a wider acceptance by semiconductor design centres. Therefore, a non-optimized ring oscillator design overestimating a technology's propagation delay, and hence underestimating a technology's performance, could lead to unnecessary development time and costs for the fabrication facility.

Electronic design automation can aid in ring oscillator design through time efficient layout implementation and optimization. In this chapter, the newly developed RF mixed-signal design methodology will be demonstrated by the design of a ring oscillator in 90nm SOI CMOS. The ring will be optimized with IBM's EDA layout synthesis tool for minimum propagation delay.

4.1. A 90nm SOI CMOS Technology

The SOI CMOS technology is emerging as a commercially viable alternative to bulk-CMOS because of its performance advantages, and the exponentially rising development costs and scaling challenges of bulk-CMOS technology beyond the 0.25μm node.
For a comparison between bulk and SOI CMOS devices, a cross-section of the physical structure of an nFET SOI device is shown in Figure 4-2. An SOI device has a buried oxide (SiO$_2$) layer that acts as a lateral isolation barrier between the silicon substrate and the implanted device. The active region of the device is known as the body. The thickness of the body depends on the type of SOI device, ranging from 100nm for the partially-depleted type, to around 20nm for the fully-depleted type. The 90nm SOI CMOS technology from IBM in this thesis features only partially-depleted SOI devices. Partially-depleted means that the depletion region under the gate does not extend all the way down to the buried oxide, thereby occupying the entire body of the device. In other words, partially-depleted devices have a neutral region formed between the depletion region and the buried oxide. The source/drain metallurgical junctions of the devices extend all the way down to the buried oxide. This results in a reduction of the area junction capacitance under the source/drain, and is one of the reasons for the improvement in SOI device performance over the traditional bulk-CMOS FET devices. This 90nm SOI CMOS technology is meant for circuits operating with a 1V power supply.
4. Ring Oscillator

4.1.1. Interconnects

Sub-wavelength lithography techniques for nanometre gate length scaling allowed the semiconductor IC industry to reap the benefits of Moore’s Law – increased speed, lower power dissipation per function, increased transistor density, and lower cost per function. As illustrated in Figure 4-3 for the case of sidewalls, the increased device and interconnect density from lateral scaling results in a larger coupling parasitic capacitance \((C_{p,180nm}\) for 180nm process, and \(C_{p,90nm}\) for a 90nm process) between the interconnect wires.

**Figure 4-2 Cross-Section Device Structure of a SOI nFET**
4. Ring Oscillator

4.1.2. Diffusion Pitches

The FETs in the 90nm IBM SOI CMOS kit have 2 diffusion pitch sizes, nominal and relaxed. As shown in Figure 4-4 for the 2-finger FET with nominal pitches, the length of the left, middle (between gate fingers) and right diffusion is 0.2275μm, 0.2975μm and 0.2275μm, respectively. For the 2-finger FET with relaxed pitches, the length of the left, middle and right diffusion is 0.4550μm, 0.5425μm and 0.4550μm, respectively.

\[ C_{P_{180nm}} = \frac{\varepsilon A}{x} \]

180nm Process

\[ C_{P_{90nm}} = \frac{2\varepsilon A}{x} \]

90nm Process

Figure 4-3 Interconnect Sidewall Coupling Parasitic
With the relaxed pitch FET, the coupling parasitic capacitance between the lowest source/drain interconnect layer and the gate fingers of the device is approximately 5 times smaller than the nominal pitch. Although it was expected that a near doubling of diffusion pitch sizes would result in a capacitance decrease by a factor of 2, modelling of the FET structure revealed a further capacitance reduction. With the relaxed pitch however, the diffusion areas are increased by a factor of around 2, resulting in an increase in area junction capacitance under the source/drain of the same factor. As well, there is a slight increase in the sidewall junction capacitance with the increased perimeter of the relaxed diffusion pitches. The lowering of the coupling capacitance is more significant than these increases in junction capacitances, resulting in the relaxed pitch FET exhibiting better high-frequency performance than the nominal pitch. Therefore, the relaxed pitch FET was the device chosen to implement the ring oscillator.
4.1.3. Unity Current Gain Frequency and Unity Power Gain Frequency

The unity current gain frequency, commonly referred as the transit frequency, is a figure of merit for the high-frequency operation of a MOSFET. It is defined as the input frequency at which the small-signal current gain of the MOSFET becomes unity. The small-signal current gain is the ratio of small-signal drain current to small-signal gate current. [21] Figure 4-5 shows the small-signal hybrid-π model for the MOSFET.

\[ f_T = \frac{g_m}{2\pi (C_{gs} + C_{gd})} \]  

(4-2)

The equation is derived when the MOSFET was configured as common-source amplifier with an input current source at the gate and its output short-circuited, resulting in \( f_T \) being independent of \( R_g, g_d \), and \( C_{bd} \).
Since the $f_T$ is proportional to the FET's transconductance and inversely proportional to the gate-to-source and gate-to-drain capacitances, a higher $f_T$ denotes better FET high-frequency amplification. Increasing the source-drain bias current with the device width held constant can increase its $g_m$, and hence $f_T$, but a rate of diminishing returns as eventually $g_m$ becomes independent of the bias current. The $g_m$ also scales with the width of the device, but the gate-to-source and gate-to-drain capacitances do as well, so the device's $f_T$ will not increase.

The unity power gain frequency, commonly known as the maximum frequency of oscillation, is another benchmark for the high-frequency operation of a MOSFET. It is defined as the input frequency at which the maximum available power gain of the MOSFET becomes unity. The maximum available power gain is the power gain of the device when its input and output impedances are conjugately matched to the source and load, respectively. [21]

For a MOSFET operating in the saturation region and with the small-signal model as shown in Figure 4-5, an estimate for its unity power gain frequency [22] is given by

$$f_{MAX} = \frac{g_m}{4\pi \sqrt{R_s g_m C_{gsd} + g_d (C_{gsd} + C_{psd}) (C_{gsd} + C_{psd})}} \quad (4-3)$$
4. Ring Oscillator

4.2. CMOS Inverter

The CMOS inverter is the basic building block of CMOS digital circuits. As illustrated in Figure 4-6, it is composed of 2 complementary transistors (nFET and pFET) with the gates of these transistors connected together to form the input of the inverter, and the drains of these transistors connected to form the output.

The voltage transfer characteristic (VTC) of the inverter is plotted in Figure 4-7a). The point where input voltage ($V_{IN}$) equals the output voltage ($V_{OUT}$) is known as the inverter switching threshold ($V_m$). The noise margin of the inverter is defined as the maximum amount of noise applied at the input while the output remains in the correct logic state. The high and low noise margins are given by $NM_H = V_{OH} - V_{IH}$ and $NM_L = V_{IL} - V_{OL}$, respectively. To optimize the inverter for noise immunity, the switching threshold must be equal to $V_{DD}/2$. [23]
4. Ring Oscillator

The inverter architecture presented is used in ring oscillator design in this chapter, and it is implemented in IBM’s 90nm SOI CMOS technology with floating-body, partially-depleted, relaxed diffusion pitch, regular V_T FETs. The inverter is constructed with a 2-finger nFET device and a 4-finger pFET device. From SPECTRE simulations, the inverter achieves a switching threshold of approximately 0.5V (i.e. \( V_{DD}/2 \)) when the inverter’s pFETs are twice the size of its nFETs.

4.2.1. Propagation Delay – Inverter

To understand the underlying behaviour of a CMOS inverter, assume that the MOS transistor can be modelled as a switch with an infinite off-resistance (for \( |V_{gs}|<|V_{d}| \)) and a finite on-resistance (for \( |V_{gs}|>|V_{d}| \)). When the inverter’s input \( V_{IN} \) is high (equal to \( V_{DD} \)), the nFET is on and the pFET is off. Replacing the nFET by the on-resistance model (\( R_{on,n} \), a path is formed between the output \( V_{OUT} \) and ground, as illustrated in

Figure 4-7 Inverter Voltage Transfer Function

The inverter architecture presented is used in ring oscillator design in this chapter, and it is implemented in IBM’s 90nm SOI CMOS technology with floating-body, partially-depleted, relaxed diffusion pitch, regular V_T FETs. The inverter is constructed with a 2-finger nFET device and a 4-finger pFET device. From SPECTRE simulations, the inverter achieves a switching threshold of approximately 0.5V (i.e. \( V_{DD}/2 \)) when the inverter’s pFETs are twice the size of its nFETs.

4.2.1. Propagation Delay – Inverter

To understand the underlying behaviour of a CMOS inverter, assume that the MOS transistor can be modelled as a switch with an infinite off-resistance (for \( |V_{gs}|<|V_{d}| \)) and a finite on-resistance (for \( |V_{gs}|>|V_{d}| \)). When the inverter’s input \( V_{IN} \) is high (equal to \( V_{DD} \)), the nFET is on and the pFET is off. Replacing the nFET by the on-resistance model (\( R_{on,n} \)), a path is formed between the output \( V_{OUT} \) and ground, as illustrated in
4. Ring Oscillator

4.1 CMOS Inverter

Figure 4-8 a), which sinks current from $V_{OUT}$ and brings its potential to 0V. Similarly, when the input is low (equal to 0V), the pFET is on and replaced by the on-resistance $R_{on,p}$, and the nFET is off. The path now exists between $V_{DD}$ and $V_{OUT}$, as illustrated in Figure 4-8 b), sourcing current to bring the potential of $V_{OUT}$ to $V_{DD}$. The capacitor $C_{OUT}$ between $V_{OUT}$ and ground represents a lumped equivalent of the capacitors at the output node of the inverter, namely gate-to-drain and body-to-drain capacitances. [19]

![Figure 4-8 CMOS Inverter Static Switch Model Representation](image)

To gain some insight into the transient response of the CMOS inverter, assume that its transistors turn on-and-off instantaneously in the switch model representation. In reality, transistors do not turn on-and-off instantaneously and their equivalent on-resistance is not constant, but this assumption will lead to simple and still meaningful results. The error introduced by this assumption will be discussed later. Therefore, for the case when $V_{IN}$ goes from low-to-high (as in a stepped input), the response time is determined by the time required to discharge the capacitance at $V_{OUT}$ (modelled by $C_{OUT}$) from $V_{DD}$.
4. Ring Oscillator

$V_{DD}$ to 0V through the on-resistance. The propagation delay for a high-to-low voltage level transition at the output, denoted by $t_{pHL}$, is measured as the time difference between the $V_{IN}$ and $V_{OUT}$ signals when they both cross 50% of $V_{DD}$. This is illustrated in Figure 4-9, as well as the case for a low-to-high $V_{OUT}$ transition, denoted by $t_{pLH}$.

![Graph showing input and output signals of a CMOS inverter switch model](image)

**Figure 4-9 Input and Output Signals of a CMOS Inverter Switch Model**

Referring to the resistor and capacitor network of Figure 4-8 a) for $t_{pHL}$, the current $I$ through the capacitor $C_{OUT}$ is related to the changing output voltage by

$$I = C_{OUT} \frac{dV_{OUT}}{dt}$$

(4-4)

Substituting the following expression for current $I$ and noting its direction through the resistor $R_{on,n}$

$$I = \frac{V_{OUT}}{R_{on,n}}$$

(4-5)

And then isolating for $dt$, results in
4. Ring Oscillator

\[
dt = - \frac{\text{R}_{\text{on-n}} \cdot C_{\text{OUT}}}{V_{\text{OUT}}} \, dV_{\text{OUT}} \tag{4-6}
\]

From Figure 4-9, integrating the left-hand side of equation (4-6) from time \( t_1 \) to \( t_2 \), and the right-hand side from voltage \( V_{\text{DD}} \) to \( 0.5V_{\text{DD}} \), the propagation delay for a high-to-low transition is given by

\[
t_{\text{pHL}} = t_2 - t_1 = -\text{R}_{\text{on-n}} \cdot C_{\text{OUT}} \ln \left( \frac{0.5V_{\text{DD}}}{V_{\text{DD}}} \right) = 0.69 \text{R}_{\text{on-n}} \cdot C_{\text{OUT}} \tag{4-7}
\]

Similarly, the propagation delay for a low-to-high transition can be shown to be

\[
t_{\text{pLH}} = 0.69 \text{R}_{\text{on-p}} \cdot C_{\text{OUT}} \tag{4-8}
\]

For every signal cycle, there is a downward as well as an upward transition, and since the high-to-low and low-to-high propagation delays are not usually the same, the estimated propagation delay of the CMOS inverter \( t_p \) is the average delay and given by

\[
t_p = \frac{t_{\text{pHL}} + t_{\text{pLH}}}{2} \tag{4-9}
\]

Therefore, for high-frequency digital and mixed-signal applications, the CMOS inverter should have a small output capacitance and its transistors should have a low on-resistance for minimal propagation delay. The transistor on-resistance can be reduced by increasing its \( W/L \) ratio, but this also increases the gate-to-drain and body-to-drain capacitances at the output, so it depends on how much these factors scale with transistor size as to whether the delay increases or decreases. This will be investigated in the following sections when the inverter is used in a ring oscillator.
4.2.2. Unity Current Gain Frequency and Unity Power Gain Frequency

To gain some insight into the high-frequency limitations of the CMOS inverter, the unity current gain frequency and unity power gain frequency figures of merit are derived. Figure 4-10 a) depicts the small-signal model for the CMOS inverter based on the transistor parameters in Figure 4-5 (parameters belonging to the nFET are denoted by the subscript "\_n", and similarly for the pFET, the subscript "\_p").

![Small Signal Model of CMOS Inverter](image)

**Figure 4-10 Equivalent Small Signal Model of CMOS Inverter**

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
To simplify the small-signal circuit in Figure 4-10 a), assume that points A and B can be shorted together, and then the equivalent circuit shown in Figure 4-10 b) is formed for the CMOS inverter. This assumption is valid since the delay is approximately matched between the nFET and the pFET path. This is based on the fact that the pFET is twice the size of the nFET, having 4 gate-fingers of width $W_f$ instead of 2. This pFET to nFET size ratio was determined earlier for the inverter, as it yielded a switching threshold of approximately 0.5V. Therefore, the pFET’s gate resistance will be about one-half of the nFET, and its gate-to-source capacitance will be approximately twice of that of the nFET. The gate resistance ($R_g$) and the gate-to-source capacitance ($C_{gs}$) form a resistor-capacitor network, whose propagation delay is approximately equal to $0.69R_gC_{gs}$. This is based on a similar derivation for the switch model inverter experiencing a voltage step transition (high-to-low or low-to-high) at its input. Hence, a signal from $V_{in}$ will have an approximate delay $t_{pb}$ at point B given by

$$t_{pb} = 0.69R_{g-p}C_{gs-p} = 0.69\frac{R_{g-n}}{2}C_{gs-n} = 0.69R_{g-n}C_{gs-n}$$  (4-10)

Which is equal to the delay experienced by the same signal at point A. Thus, points A and B can be considered the same point from the signal’s perspective, and shorted together to simplify the small-signal equivalent circuit of the inverter.

The CMOS inverter small-signal model of Figure 4-10 b) has the same form as the MOSFET hybrid-$\pi$ model of Figure 4-5. Therefore, an estimate of the inverter’s
4. Ring Oscillator

unity current gain frequency $f_{T,\text{inv}}$ can be formulated by replacing the MOSFET's parameters of (4-2) with their equivalent counterparts in the inverter model, and is given by

$$
f_{T,\text{inv}} = \frac{g_{m_n} + g_{m_p}}{2\pi (C_{gs_n} + C_{gs_p} + C_{gd_n} + C_{gd_p})} \tag{4-11}
$$

Similarly, an expression for the estimated unity power gain frequency of the inverter $f_{MAX,\text{inv}}$ can be formulated from (4-3) and shown to be

$$
f_{MAX,\text{inv}} = \frac{g_{m_n} + g_{m_p}}{4\pi \sqrt{(R_{s,p})(g_{m_n} + g_{m_p})(C_{gs_n} + C_{gs_p}) + (g_{d_n} + g_{d_p})(C_{gd_n} + C_{gd_p})}} \tag{4-12}
$$

Where

$$
R_{s,p} = R_{s,n} // R_{s,p} \tag{4-13}
$$

And

$$
C_g = C_{gs_n} + C_{gs_p} + C_{gd_n} + C_{gd_p} \tag{4-14}
$$

The unity current gain frequency and unity power gain frequency of the inverter are figures of merit for the small-signal performance of the inverter at high-frequencies. Whether or not these provide an indication of the inverter's large-signal performance, as in a ring oscillator, will be investigated in this chapter.
4. Ring Oscillator

4.2.3. Propagation Delay - Ring

To form a ring oscillator, CMOS inverter cells (or stages) must be chained together in a loop, where each stage must drive the following stage to propagate an oscillating signal around the loop at a frequency determined by the delay of the inverter and the number of inverters in the loop (4-1). From equation (4-9), the propagation delay is proportional to the on-resistances of the inverter’s transistors. The value of the on-resistance is based on a non-linear function of the voltage across its transistor.

Consider the case of the two stage inverter chain illustrated in Figure 4-11 a), where the total wiring resistance and capacitance connecting the two are represented by $R_w$ and $C_w$, respectively. The resistive-capacitive wire is distributed in nature, and will be modelled as such. At time $t = 0$, the voltage at $V_{IN}$ is $0V$, and $V_X$ is close to $1V$. Then, $V_{IN}$ experiences an instantaneous voltage transition from $0V$ to $1V$. The charge on inverter $II$’s output capacitance $C_{OUT}$, the wiring capacitance $C_w$, and inverter $I2$’s input capacitance $C_{IN}$ will discharge as a current through the nFET’s on-resistance of $II$ as depicted in Figure 4-11b). The fan-out resistor $R_{g,n}/R_{g,p}$ and capacitor $C_{IN}$ represent the equivalent input impedance of $I2$ loading the output of $II$. This impedance is formulated by replacing $I2$ with its equivalent small-signal model from Figure 4-10 b), and is also treated as a distributed resistor-capacitor network.
Point A in Figure 4-12 [23] represents the operating point of the I1's nFET before the voltage transition at $V_{IN}$. Point B is the operating point immediately afterwards, where now the nFET's $V_{DS} = 1V$ and $V_{GS} = 1V$. As $V_Y$'s potential is brought towards ground, the drain-to-source voltage across the nFET decreases (since $V_{DS} = V_Y$). The discharging current will follow the arrowed path along the nFET's $I_{DS}$-$V_{DS}$ characteristic curve to point C. The on-resistance between the drain and source is also changing along this path. An estimate of the average on-resistance of the nFET is given by the reciprocal of the slope of the line BC [23], and is expressed by
Where \( \frac{W}{L} \) is the transistor’s width-to-length ratio, \( \mu_n \) is the nFET’s mobility, \( C\text{ox} \) is the nFET’s gate-oxide capacitance per unit area, and \( R_{\text{on, } n} \) is the nFET’s on-resistance per square. From SPECTRE simulations, which includes channel length modulation effects, the relaxed diffusion pitch nFET and pFET exhibit \( R_{\text{on, } n} \) of 13.07k\( \Omega \)/square and \( R_{\text{on, } p} \) of 30.54k\( \Omega \)/square respectively.

![Graph of Average nFET On-Resistance](image)

**Figure 4-12 Average nFET On-Resistance**

While it may not be intuitively obvious why a line is drawn from B-to-C and not from D-to-C for a better approximation of the average on-resistance, it should be noted that this analysis is based on a voltage step at the input, when, in reality it is a voltage ramp generated by the proceeding inverter stage in the ring. Later in the chapter, the in-
verter’s nFET $I_{DS}$-$V_{DS}$ simulation behaviour in a ring will be explored, and it will show that this approximation for on-resistance is reasonable. This approximation is expected to help provide some propagation delay insights as the size of the inverter’s transistors are scaled.

The parasitic capacitances of the FET are dependent on the voltages across its terminals, and therefore its operation mode; cut-off, triode or saturation. Figure 4-13 is a plot of a typical VTC of a CMOS inverter, and depicts the operating modes of the nFET and pFET for a given input voltage $V_{IN}$. As stated earlier, the propagation delay is defined as the time difference between the 50% transitions of the input and output signals. When the inverter’s input is set to $V_{DD}/2$, the output should be around $V_{DD}/2$, since the switching threshold $V_M$ was chosen to be at $V_{DD}/2 = 0.5V$. Here, both of the inverter’s transistors are operating in saturation, and it is at this point where the values of their capacitances were chosen for use in the propagation delay estimate.
As illustrated in Figure 4-11 b), the capacitances which will affect the transient response time of inverter II are $C_{OUT}$, $C_W$ and $C_{IN}$. $C_{OUT}$ represents a lumped equivalent capacitance between $V_I$ and ground of the parasitic FET capacitances at II’s output. From Figure 4-14, $C_{OUT}$ is the parallel combination of $C_{gd,n}$, $C_{gd,p}$, $C_{bd,n}$ and $C_{bd,p}$. Similarly, the FET capacitances at the input of II are lumped together to form $C_{IN}$ from $V_X$ to ground. $C_{IN}$ is the parallel combination of $C_{gd,n}$, $C_{gd,p}$, $C_{gs,n}$ and $C_{gs,p}$. This analysis assumes that $V_{IN}$ is driven by an ideal voltage source with zero rise and fall times, as in a step.
From SPECTRE simulations, the relevant parasitic capacitances per unit width of the FET are listed in Table 4-1 for the relaxed diffusion pitch FET of minimum gate length (90nm).

**Table 4-1 Relevant FET Capacitances for Determining Delay**

<table>
<thead>
<tr>
<th>Parasitic Capacitor</th>
<th>Relaxed Diffusion Pitch Capacitance (fF/µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{gs_n}$</td>
<td>0.69</td>
</tr>
<tr>
<td>$C_{gd_n}$</td>
<td>0.33</td>
</tr>
<tr>
<td>$C_{bd_n}$</td>
<td>0.21</td>
</tr>
<tr>
<td>$C_{gs_p}$</td>
<td>0.58</td>
</tr>
<tr>
<td>$C_{gd_p}$</td>
<td>0.32</td>
</tr>
<tr>
<td>$C_{bd_p}$</td>
<td>0.11</td>
</tr>
<tr>
<td>$C_w$</td>
<td>From Extraction</td>
</tr>
</tbody>
</table>
The sheet resistance \( R_{\text{sheet}} \) of the gate material for the relaxed diffusion pitch FET is 7.1 \( \Omega \) per square. The FET's gate resistance \( R_g \) is therefore given by

\[
R_g = \frac{R_{\text{sheet}} \, W_f}{n \, L}
\]  

(4-16)

Where \( n \) is the number of gate fingers, \( W_f \) is the width of the gate finger, and \( L \) is the length of the gate. For the CMOS inverter, the nFET is expected to be a 2 gate finger device and the pFET a 4 gate finger device. It is also worth mentioning that the wiring resistance \( R_w \) between the inverters in a chain will be determined by parasitic extraction tools.

Earlier, the propagation delay for a single resistor-capacitor network excited by a high-to-low voltage stepped-input was derived (4-7). To do the same analysis for the delay of a given RC chain of Figure 4-15, characterized by a number of time constants, becomes too complex for practical design efforts.

Figure 4-15 RC Chain Network
A first-order approximation for an RC chain network was developed and presented in [24]. Known as the Elmore delay equation, it determines the value of the dominant time constant (i.e. pole) in the network, and is given by

$$\tau_{ED} = R_1 C_1 + (R_1 + R_2)C_2 + \ldots + (R_1 + R_2 + \ldots + R_N)C_N$$

(4-17)

A lumped RC model for wiring parasitics results in an overly pessimistic estimate for the wire’s propagation delay. With reference to Figure 4-16 where $R_w$ and $C_w$ represent the total resistance and total capacitance of the wire respectively, to account for the distributed nature of these parasitics along the wire, the following equation given in [19] is used to approximate its delay,

$$t_{pd\_wire} = 0.38 R_w C_w$$

(4-18)

\[\text{Figure 4-16 Distributed RC Wire}\]

With the Elmore delay theory and the distributed RC delay theory, an average delay estimate for the switch model two stage inverter chain, shown again in Figure 4-17 for convenience, can be formulated.
The high-to-low propagation delay is given by

\[ t_{PHL_{\text{inv-chain}}} = 0.69 R_{on,n} C_{OUT} + (0.69 R_{on,n} + 0.38 R_W) C_W 
+ (0.69 R_{on,n} + 0.69 R_W + 0.38 (R_{g,n} \parallel R_{g,p})) C_{IN} \]  

(4-19)

Similarly, the low-to-high propagation delay is given by

\[ t_{PLH_{\text{inv-chain}}} = 0.69 R_{on,p} C_{OUT} + (0.69 R_{on,p} + 0.38 R_W) C_W 
+ (0.69 R_{on,p} + 0.69 R_W + 0.38 (R_{g,n} \parallel R_{g,p})) C_{IN} \]  

(4-20)

Finally, the average propagation delay for an inverter in a chain is given by

\[ t_{pd_{\text{inv-chain}}} = \frac{t_{PHL_{\text{inv-chain}}} + t_{PLH_{\text{inv-chain}}}}{2} \]  

(4-21)

Through the substitution of \( t_{pd_{\text{inv-chain}}} \) for \( t_p \) in equation (4-1), the oscillating frequency of a ring can be predicted.

4.3. Ring Oscillator

An 11 inverter-stage ring oscillator is chosen as the test vehicle for optimization because of simulation time. The time for an 11-stage ring to start-up and reach a steady state oscillation frequency is tens of nanoseconds. The standard industry benchmarking...
ring circuit contains 101 inverter-stages since its frequency of oscillation is slow enough for oscilloscope measurements. However, a 101-stage ring would require hundreds of nanoseconds of simulation time.

The CMOS inverter is constructed with a 2-finger nFET device and a 4-finger pFET device, both devices are implemented with relaxed diffusion pitches. To maximize the operating frequency of the FET, minimum gate lengths should be chosen [21]. The main reason for this choice is that the time required for a carrier to cross from the drain to the source scales with the channel’s length, and shorter cross times imply high frequency. Therefore, the gate length $L$ for the inverter’s FETs is 90nm.

The FET finger width $W_f$ is the remaining device parameter which could influence the average propagation delay of the inverter. Since the aim is to optimize the ring for minimal propagation delay (and hence, maximize oscillation frequency), parametric variations of 11-stage ring oscillator designs with varying device finger widths are studied. Table 4-2 lists the FET finger widths for each ring. In total, there are 6 different rings which are analyzed and simulated.

**Table 4-2 FET Finger Width and Total Device Width for the Ring Netlists**

<table>
<thead>
<tr>
<th>Ring Netlist Designation</th>
<th>FET Finger Width $W_f$ ($\mu$m)</th>
<th>nFET Device Width $W_n$ ($\mu$m)</th>
<th>pFET Device Width $W_p$ ($\mu$m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>1.05</td>
<td>2.1</td>
</tr>
<tr>
<td>2</td>
<td>1.05</td>
<td>2.1</td>
<td>4.2</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>2.8</td>
<td>5.6</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>3.78</td>
<td>7.56</td>
</tr>
<tr>
<td>5</td>
<td>2.45</td>
<td>4.9</td>
<td>9.8</td>
</tr>
<tr>
<td>6</td>
<td>5.25</td>
<td>10.5</td>
<td>21</td>
</tr>
</tbody>
</table>
The transconductance and output conductance scale with device width. For the nFET, the transconductance \((g_{m,n})\) and output conductance \((g_{d,n})\) per device width are 0.98mS/\(\mu\)m and 107\(\mu\)mhos/\(\mu\)m respectively. Similarly for the pFET, the transconductance \((g_{m,p})\) and output conductance \((g_{d,p})\) per device width are 0.4mS/\(\mu\)m and 63\(\mu\)mhos/\(\mu\)m respectively.

The reader should be aware that the FET variables for all calculations in the following sections were taken from SPECTRE model output parameters for each device \(W_f\). The FET’s per unit width device capacitances, transconductance and output conductance presented could equally have been scaled accordingly and substituted for all calculations to obtain similar results.

4.4. Simulation

Each of the 6 rings is composed of a looped-chain of identical inverters, whose devices are sized according to Table 4-2. To examine the small-signal high frequency behaviour of these 6 inverters, simulations are performed on their netlists to determine \(f_{r_{inv}}\) and \(f_{max_{inv}}\). Then, transient simulations are performed on the rings themselves, to find the inverter’s average propagation delay from the oscillation frequency (4-1). With these simulation results, as well as theoretical calculations, a prevailing trend for designing a ring for maximum oscillation frequency may be identified.
4. Ring Oscillator

4.4.1. Unity Current Gain Frequency and Unity Power Gain Frequency

Referring to the inverter schematic of Figure 4-6, the power rail $V_{DD}$ and the input $V_{IN}$ were DC biased at 1V and 0.5V respectively. At this operating point, both of the inverter’s transistors are in saturation, and the $f_{T, inv}$ and $f_{max, inv}$ will be at their highest. While operating in a 25°C ambient environment, a two-port small-signal S-parameter SPECTRE simulation was executed on each inverter. Table 4-3 and Table 4-4 show the simulated results as well as calculations for $f_{T, inv}$ and $f_{max, inv}$ respectively.

### Table 4-3 Inverter Device $W_f$ and $f_{T, inv}$

<table>
<thead>
<tr>
<th>Inverter Netlist Designation</th>
<th>FET Finger Width $W_f$ (µm)</th>
<th>Simulated $f_{T, inv}$ (GHz)</th>
<th>Theoretical $f_{T, inv}$ (GHz)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>119</td>
<td>106</td>
<td>-11%</td>
</tr>
<tr>
<td>2</td>
<td>1.05</td>
<td>117</td>
<td>104</td>
<td>-11%</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>117</td>
<td>103</td>
<td>-12%</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>117</td>
<td>103</td>
<td>-12%</td>
</tr>
<tr>
<td>5</td>
<td>2.45</td>
<td>116</td>
<td>103</td>
<td>-11%</td>
</tr>
<tr>
<td>6</td>
<td>5.25</td>
<td>115</td>
<td>102</td>
<td>-11%</td>
</tr>
</tbody>
</table>

### Table 4-4 Inverter Device $W_f$ and $f_{max, inv}$

<table>
<thead>
<tr>
<th>Inverter Netlist Designation</th>
<th>FET Finger Width $W_f$ (µm)</th>
<th>Simulated $f_{max, inv}$ (GHz)</th>
<th>Theoretical $f_{max, inv}$ (GHz)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>938</td>
<td>673</td>
<td>-28%</td>
</tr>
<tr>
<td>2</td>
<td>1.05</td>
<td>336</td>
<td>332</td>
<td>-1%</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>243</td>
<td>248</td>
<td>2%</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>176</td>
<td>183</td>
<td>4%</td>
</tr>
<tr>
<td>5</td>
<td>2.45</td>
<td>134</td>
<td>142</td>
<td>6%</td>
</tr>
<tr>
<td>6</td>
<td>5.25</td>
<td>62</td>
<td>66</td>
<td>6%</td>
</tr>
</tbody>
</table>
4. Ring Oscillator

As the inverter’s device $W_f$ is increased, the simulated $f_{T_{inv}}$ remains almost constant since the device’s $g_m$, $C_{gs}$ and $C_{gd}$ scale linearly with $W_f$. This trend is also observed with the theoretical $f_{T_{inv}}$. Although calculations are around -11% off from the simulated result, they are consistent and a useful figure of merit. The $f_{\text{max}_{inv}}$ is inversely proportional to the device $W_f$. This is observed for the simulation results and calculations, except for the case when $W_f$ is 0.525μm. At this $W_f$, the simulated $f_{\text{max}_{inv}}$ is 938GHz – an increase of 2.8 times from the $f_{\text{max}_{inv}}$ for $W_f = 1.05\mu m$. From $W_f = 0.525\mu m$ to $W_f = 1.05\mu m$, an increase of $f_{\text{max}_{inv}}$ by a factor of 2 is expected, this is shown by the theory. Therefore, a modelling or simulation problem is suspected as the cause for this unbelievably high $f_{\text{max}_{inv}}$ and corresponding -28% error. Ignoring $W_f = 0.525\mu m$, $f_{\text{max}_{inv}}$ calculations are within 6% of simulation results.

4.4.2. Propagation Delay

While operating in a 25°C ambient environment and with a 1V power supply, a transient SPECTRE simulation was performed on each ring netlist to determine the average propagation delay from the oscillation frequency (4-1). Table 4-5 shows the simulated and calculated results for each ring. Note that this ring netlist contains a chain of inverters connected together by ideal wires, therefore, $R_w = 0$ and $C_w = 0$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 4-5 Ring Device $W_f$ and $t_{pd\_inv\_chain}$

<table>
<thead>
<tr>
<th>Ring Netlist Designation</th>
<th>FET Finger Width $W_f$ (µm)</th>
<th>Simulated $t_p$ (ps)</th>
<th>Theoretical $t_{pd_inv_chain}$ (ps)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>5.54</td>
<td>3.49</td>
<td>-37%</td>
</tr>
<tr>
<td>2</td>
<td>1.05</td>
<td>5.62</td>
<td>3.52</td>
<td>-37%</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>5.70</td>
<td>3.56</td>
<td>-37%</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>5.82</td>
<td>3.62</td>
<td>-38%</td>
</tr>
<tr>
<td>5</td>
<td>2.45</td>
<td>6.00</td>
<td>3.69</td>
<td>-39%</td>
</tr>
<tr>
<td>6</td>
<td>5.25</td>
<td>7.58</td>
<td>4.29</td>
<td>-43%</td>
</tr>
</tbody>
</table>

With decreasing device $W_f$, simulation results show the average propagation delay $t_p$ of the ring’s inverter to decrease as well. Therefore, the ring’s oscillation frequency is maximized for the case when $W_f = 0.525\mu$m – the smallest FET finger width. Although calculations underestimated the average delay for an inverter in a chain $t_{pd\_inv\_chain}$ by 38% - a significant percentage, the same conclusion is reached for maximizing the oscillation frequency. The reasons for this large error will be discussed later in the chapter.

4.4.3. Observations

The simulation results of Table 4-3, Table 4-4 and Table 4-5 for inverter $f_{T\_inv}$, $f_{\text{max}\_inv}$, and $t_p$ respectively, are plotted in Figure 4-18. From this comparison, it shows that as the small-signal figure of merit $f_{\text{max}\_inv}$ increases, the average propagation delay for an inverter in a chain decreases, as $f_{T\_inv}$ remains relatively constant with $W_f$. Therefore, for maximum oscillation frequency design, the ring’s device finger width $W_f$ should be as small as practically possible. However, this design practice may cause an inverter drive problem if there is any extra loading on its output.
Figure 4-18 Simulated Inverter $f_{T_{\text{inv}}}$, $f_{\text{max}_{\text{inv}}}$ and Average Delay

The theoretical calculations of Table 4-3, Table 4-4 and Table 4-5 for inverter $f_{T_{\text{inv}}}$, $f_{\text{max}_{\text{inv}}}$, and $t_{pd_{\text{inv chain}}}$ respectively, are plotted in Figure 4-19. This comparison also shows that for maximum oscillation frequency design, the ring’s, and hence the inverter’s, device finger width $W_f$ should be minimized.
4.5. Layout Synthesis

From the netlist optimization of the ring (and its corresponding inverter subcircuit), the next stage in the new RF mixed-signal design flow is layout design and synthesis as illustrated in Figure 4-20. Since wiring parasitics between the inverter stages are...
expected to impact the ring’s performance, the layouts of the rings with varying device
table widths are to be studied and compared to determine which ring exhibits the highest
oscillation frequency.

Figure 4-20 Layout Design and Synthesis Flow

A C++ class for the inverter needs to be coded since this circuit’s topology does
not exist in the GDSText class database. Figure 4-21 shows a pseudocode description of
the class Inverter. The coding used previously defined nFET and pFET classes (see
Chapter 3) from the database to instantiate the objects needed to represent the inverter’s devices, which reduced the development time for the inverter’s class. Coding is added to set the device objects’ attributes, such as xy co-ordinate location, number of gate fingers, gate length, etc., when these values become available. Also, additional coding is done for top-level wiring. For instance, the gate-fingers and drains of both devices are each routed together to form the inverter’s input ($V_{\text{IN}}$) and output ($V_{\text{OUT}}$), respectively.

```python
class Inverter:

Attributes:
- nFET
- pFET

Methods:
- Inverter(...) - initialize Inverter by creating instances of nFET and pFET
- read(...) - read-in user specifications to set the attributes of the nFET and pFET
- write(...) - access technology design rule database
  - output inverter layout description (for the nFET, pFET, and wiring information for $V_{\text{IN}}$, $V_{\text{OUT}}$, $V_{\text{DD}}$ and GND nets) in GDSText to the file “inverter.txt”
- main(...) - create inverter object
  - call read method
  - call write method
```

**Figure 4-21 Pseudocode for Inverter Class**

When the coding of the inverter class is complete, the GDSText Generator tool is compiled and executed. The tool creates an inverter object, and values for its attributes,
as well as for any of its device objects’ attributes, are then set. Afterwards, the write method of the inverter object is called and the layout information is sent out in GDSText-ASCII to the file “inverter.txt”. This is then imported into the GDSII Translator tool, and a GDSII-binary format layout is created. Figure 4-22 shows the synthesized inverter layout for a $W_f = 1.89\mu m$ with the power and signal pin labels from the schematic in Figure 4-6. Paying careful attention to minimize wiring parasitics, the poly strapping of a device’s gate-fingers is shared between the nFET and pFET.

![Inverter Layout](image)

**Figure 4-22 Inverter Layout**

Since the synthesised inverter was pre-wired in the code, an 11-stage ring is created by abutting 11 inverters together in a chain. The output of the last inverter was then
connected to the input of the first inverter with an ideal wire such that all inverters in the ring are loaded with the same wiring parasitics. The ring with a device \( W_f = 1.89 \mu m \) is shown in Figure 4-23.

![Figure 4-23 11-Stage Ring Oscillator](image)

4.6. Post-Layout Simulation

Ring layouts for the 6 different device \( W_f \) of Table 4-2 are synthesized. Then, each of these layouts, as well as their respective inverter subcircuit, is extracted with Mentor Graphics’ Calibre xRC for resistive and capacitive parasitics from the interconnect layers. Noteworthy parasitics from the extraction are \( R_w \) and \( C_w \) of the wiring connecting the inverters, and the values of these parasitics for each inverter netlist are listed in Table 4-6. The total resistive parasitic of the wiring \( R_w \) is mainly the summation of the drain connection resistance \( R_{\text{drain}} \) and the poly gate strap resistance \( R_{\text{strap}} = 16 \Omega \) - a constant for all inverter sizes.
4. Ring Oscillator

Table 4-6 Relevant Inverter Chain Wiring Parasitics

<table>
<thead>
<tr>
<th>Inverter Netlist Designation</th>
<th>FET Finger Width $W_f$ (µm)</th>
<th>Extracted $R_{\text{drain}}$ (Ω)</th>
<th>Extracted $R_w$ (Ω)</th>
<th>Extracted $C_w$ (fF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>12.00</td>
<td>28.00</td>
<td>0.90</td>
</tr>
<tr>
<td>2</td>
<td>1.05</td>
<td>5.00</td>
<td>21.00</td>
<td>0.95</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>3.50</td>
<td>19.50</td>
<td>0.96</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>3.00</td>
<td>19.00</td>
<td>0.96</td>
</tr>
<tr>
<td>5</td>
<td>2.45</td>
<td>2.00</td>
<td>18.00</td>
<td>0.96</td>
</tr>
<tr>
<td>6</td>
<td>5.25</td>
<td>1.50</td>
<td>17.50</td>
<td>0.70</td>
</tr>
</tbody>
</table>

The extracted inverter netlists of varying $W_f$ are again examined for their small-signal high-frequency behaviour. Then, the ring netlists are studied to find the inverter’s average propagation delay from the oscillation frequency (4-1). With these results, insight into the designing of a ring for maximum oscillation frequency may be reached as in Section 4.4.

4.6.1. Unity Current Gain Frequency and Unity Power Gain Frequency

With the same testbench described in Section 4.4.1, a two-port small-signal S-parameter SPECTRE simulation was executed on each extracted inverter. Table 4-7 and Table 4-8 show the simulated results as well as calculated for $f_{\text{r,inv}}$ and $f_{\text{max,inv}}$ respectively. The theoretical $f_{\text{r,inv}}$ and $f_{\text{max,inv}}$ includes the effect of $C_w$, modelled as a capacitance in parallel with $C_{gs,n}$ and $C_{gs,p}$. In the calculation of $f_{\text{max,inv}}$, the total gate resistance is the summation of the inverter’s equivalent gate resistance $R_{gs,eq}$ and the poly gate strap resistance $R_{\text{strap}} = 16Ω$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 4-7 Extracted Inverter Device $W_f$ and $f_{r_{inv}}$

<table>
<thead>
<tr>
<th>Inverter Netlist Designation</th>
<th>FET Finger Width $W_f$ (µm)</th>
<th>Simulated $f_{r_{inv}}$ (GHz)</th>
<th>Theoretical $f_{r_{inv}}$ (GHz)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>81</td>
<td>77</td>
<td>-5%</td>
</tr>
<tr>
<td>2</td>
<td>1.05</td>
<td>94</td>
<td>87</td>
<td>-7%</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>98</td>
<td>91</td>
<td>-8%</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>101</td>
<td>93</td>
<td>-8%</td>
</tr>
<tr>
<td>5</td>
<td>2.45</td>
<td>103</td>
<td>95</td>
<td>-8%</td>
</tr>
<tr>
<td>6</td>
<td>5.25</td>
<td>112</td>
<td>100</td>
<td>-11%</td>
</tr>
</tbody>
</table>

Table 4-8 Extracted Inverter Device $W_f$ and $f_{max_{inv}}$

<table>
<thead>
<tr>
<th>Inverter Netlist Designation</th>
<th>FET Finger Width $W_f$ (µm)</th>
<th>Simulated $f_{max_{inv}}$ (GHz)</th>
<th>Theoretical $f_{max_{inv}}$ (GHz)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>259</td>
<td>286</td>
<td>10%</td>
</tr>
<tr>
<td>2</td>
<td>1.05</td>
<td>178</td>
<td>198</td>
<td>11%</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>150</td>
<td>165</td>
<td>10%</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>120</td>
<td>133</td>
<td>11%</td>
</tr>
<tr>
<td>5</td>
<td>2.45</td>
<td>99</td>
<td>109</td>
<td>10%</td>
</tr>
<tr>
<td>6</td>
<td>5.25</td>
<td>52</td>
<td>58</td>
<td>12%</td>
</tr>
</tbody>
</table>

With increasing device finger width $W_f$, the inverter’s $f_{r_{inv}}$ increases - but at a decreasing rate. This is attributed to the parasitic wiring capacitance $C_w$. As the device’s capacitances increase with $W_f$, $C_w$ remains relatively constant and therefore its effect is less significant. This trend is also observed with the calculated $f_{r_{inv}}$, however it consistently underestimates the unity current gain frequency of the inverter by around 8%. The $f_{max_{inv}}$ decreases with increasing $W_f$. Also, largely due to the effect of the poly gate strap resistance $R_{strap}$, the extracted $f_{max_{inv}}$ is significantly lower than the non-extracted $f_{max_{inv}}$.
4. Ring Oscillator

for each \( W_f \). The calculations overestimates \( f_{\text{max, inv}} \), but they remain consistent over device finger width at approximately 11%.

4.6.2. Propagation Delay

Using the same testbench described in Section 4.4.2, a transient SPECTRE simulation was performed on each extracted ring netlist. The average delay simulation results, as well as theoretical estimates, are listed in Table 4-9.

### Table 4-9 Extracted Ring Device \( W_f \) and \( t_{\text{pd, inv, chain}} \)

<table>
<thead>
<tr>
<th>Ring Netlist Designation</th>
<th>FET Finger Width ( W_f ) (( \mu \text{m} ))</th>
<th>Simulated ( t_p ) (ps)</th>
<th>Theoretical ( t_{\text{pd, inv, chain}} ) (ps)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.525</td>
<td>8.15</td>
<td>4.80</td>
<td>-41%</td>
</tr>
<tr>
<td>2</td>
<td>1.05</td>
<td>7.29</td>
<td>4.33</td>
<td>-41%</td>
</tr>
<tr>
<td>3</td>
<td>1.4</td>
<td>7.18</td>
<td>4.25</td>
<td>-41%</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>7.25</td>
<td>4.24</td>
<td>-42%</td>
</tr>
<tr>
<td>5</td>
<td>2.45</td>
<td>7.43</td>
<td>4.25</td>
<td>-43%</td>
</tr>
<tr>
<td>6</td>
<td>5.25</td>
<td>9.00</td>
<td>4.81</td>
<td>-47%</td>
</tr>
</tbody>
</table>

From the simulation results, as the device \( W_f \) scales from 0.525\( \mu \text{m} \) to 5.25\( \mu \text{m} \), the average propagation delay of the inverter reaches a minimum of 7.18ps at \( W_f = 1.4\mu \text{m} \). It is at this FET finger width that the ring’s design has been optimized for the wiring parasitics of the layout. As seen previously, calculations for \( t_{\text{pd, inv, chain}} \) underestimate the simulation results, and the percentage difference is on average -42%. However, calculations also show that the optimum FET finger width for maximum oscillation frequency occurs when \( W_f = 1.4\mu \text{m} \). Table 4-10 shows the percentage difference from the optimum inverter delay for the simulation and theoretical results, respectively.
Table 4-10 Percentage Difference from Optimum Extracted Inverter Delay

<table>
<thead>
<tr>
<th>Ring’s Device $W_f$ (μm)</th>
<th>Simulated $t_p$ (ps)</th>
<th>Difference from $t_p = 7.18$ps (%)</th>
<th>Theoretical $t_{pd inv _chain}$ (ps)</th>
<th>Difference from $t_p = 4.30$ps (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.525</td>
<td>8.15</td>
<td>13.5%</td>
<td>4.83</td>
<td>12.4%</td>
</tr>
<tr>
<td>1.05</td>
<td>7.29</td>
<td>1.5%</td>
<td>4.37</td>
<td>1.7%</td>
</tr>
<tr>
<td>1.4</td>
<td>7.18</td>
<td>0.0%</td>
<td>4.30</td>
<td>0.0%</td>
</tr>
<tr>
<td>1.89</td>
<td>7.25</td>
<td>0.9%</td>
<td>4.30</td>
<td>0.1%</td>
</tr>
<tr>
<td>2.45</td>
<td>7.43</td>
<td>3.5%</td>
<td>4.33</td>
<td>0.7%</td>
</tr>
<tr>
<td>5.25</td>
<td>9.00</td>
<td>25.4%</td>
<td>4.97</td>
<td>15.7%</td>
</tr>
</tbody>
</table>

The trend observed here in propagation delay with device $W_f$ is clearly different than that of the non-extracted ring, making the layout design an important factor in the ring’s RF performance. To determine the ring’s dominating time constants, the high-to-low propagation delay equation is broken down into the following RC networks,

$$
t_{pHL\_inv\_chain} = 0.69R_{on\_n}C_{OUT} + (0.69R_{on\_n} + 0.38R_W)C_W$$

$$+ (0.69R_{on\_n} + 0.69R_W + 0.38(R_{g\_n} \parallel R_{g\_p})C_{IN})$$

$$= 0.69R_{on\_n}C_{OUT} + 0.69R_{on\_n}C_W + 0.38R_WC_W$$

$$+ 0.69R_{on\_n}C_{IN} + 0.69R_WC_{IN} + 0.38(R_{g\_n} \parallel R_{g\_p})C_{IN} \quad (4-22)$$

Table 4-11 shows the values of these time constants for each of the 6 rings.
Table 4-11 Ring's Dominant Time Constants for High-to-Low Propagation Delay

<table>
<thead>
<tr>
<th>Ring’s Device $W_f$ ($\mu$m)</th>
<th>0.525</th>
<th>1.05</th>
<th>1.4</th>
<th>1.89</th>
<th>2.45</th>
<th>5.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.69R_{on,n}C_{OUT}$ (ps)</td>
<td>1.57</td>
<td>1.40</td>
<td>1.38</td>
<td>1.37</td>
<td>1.35</td>
<td>1.33</td>
</tr>
<tr>
<td>$0.69R_{on,n}C_w$ (ps)</td>
<td>0.68</td>
<td>0.35</td>
<td>0.27</td>
<td>0.20</td>
<td>0.15</td>
<td>0.05</td>
</tr>
<tr>
<td>$0.38R_wC_w$ (ps)</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>$0.69R_{on,n}C_{IN}$ (ps)</td>
<td>2.16</td>
<td>2.14</td>
<td>2.16</td>
<td>2.17</td>
<td>2.17</td>
<td>2.18</td>
</tr>
<tr>
<td>$0.69R_wC_{IN}$ (ps)</td>
<td>0.06</td>
<td>0.08</td>
<td>0.10</td>
<td>0.14</td>
<td>0.17</td>
<td>0.35</td>
</tr>
<tr>
<td>$0.38(R_{g,n}/R_{g,p})C_{IN}$ (ps)</td>
<td>0.01</td>
<td>0.03</td>
<td>0.05</td>
<td>0.10</td>
<td>0.16</td>
<td>0.76</td>
</tr>
<tr>
<td>$t_{pHL_inv_chain}$</td>
<td>4.48</td>
<td>4.02</td>
<td>3.97</td>
<td>3.98</td>
<td>4.02</td>
<td>4.67</td>
</tr>
</tbody>
</table>

This comparison also shows that although the wiring capacitance $C_w$ is relatively constant with device $W_f$, it forms an increasing delay network with the FET’s average on-resistance $R_{on,n}$ (and $R_{on,p}$) for smaller devices.

4.6.3. Observations

The extracted simulation results of Table 4-7, Table 4-8 and Table 4-9 for inverter $f_{r\_inv}$, $f_{max\_inv}$, and $t_p$ respectively, are plotted in Figure 4-24. From this comparison, and the non-extracted observations from Figure 4-18, the identification of the parasitics influencing the average propagation delay of the ring can be made. As the device $W_f$ decreases, $f_{r\_inv}$ degrades due to the nearly fixed $C_w$ - the wiring capacitive parasitic between the inverter stages in the loop. With smaller devices, the wiring parasitic $C_w$ and the FET’s average on-resistance form a more dominating time constant which loads the inverter’s output and increases the ring’s delay. As seen in the non-extracted simulation results, the $t_p$ increased and $f_{max\_inv}$ decreased with increasing $W_f$ because of the device
parasitics. The same trend is also observed here, but occurs at the point when \(f_{\text{max, inv}}\) decreases below the inverter’s \(f_{r_{\text{inv}}}\).

![Figure 4-24 Simulated Extracted Inverter \(f_{r_{\text{inv}}}, f_{\text{max, inv}}\) and Average Delay](image)

The theoretical calculations of Table 4-7, Table 4-8 and Table 4-9 for inverter \(f_{r_{\text{inv}}}, f_{\text{max, inv}},\) and \(t_{p_{\text{d, inv, chain}}}\) respectively, are plotted in Figure 4-25. By examining these results, the same conclusions can be reached for minimizing ring delay. Where at small

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
FET finger widths, the capacitive wiring parasitic adversely affects delay, and at larger finger widths, the device parasitics adversely affect delay.

Figure 4-25 Calculated Extracted Inverter $f_{T_{inv}}, f_{max_{inv}}$, and Average Delay

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
4. Ring Oscillator

4.7. Discussion

Although the theory of the average propagation delay for an inverter in a ring was able to predict the trends observed in the simulations, it continually underestimated delay for the varying FET finger widths; by around 38% for the non-extracted ring and 42% for the extracted ring.

For a non-extracted ring, the inverter’s high-to-low propagation delay is given by

$$t_{\text{pH.Inv.chain}} = 0.69R_{\text{on,n}}C_{\text{OUT}} + (0.69R_{\text{on,n}} + 0.38(R_{g,n} \parallel R_{g,p}))C_{\text{IN}}$$

(4-23)

The error in $t_{\text{pH.Inv.chain}}$ could come from underestimating the nFET’s average on-resistance $R_{\text{on,n}}$ (and likewise, the pFET’s on-resistance for $t_{\text{pLH.Inv.chain}}$), the device capacitances $C_{\text{IN}}$ and $C_{\text{OUT}}$, or a combination of the two, since the error is consistent as $W_f$ is varied.

Consider Figure 4-26 b) which shows the input $V_{\text{IN}}$ of the inverter II in Figure 4-26 a) going from low-to-high in a transient simulation of a ring with $W_f = 1.4\mu\text{m}$. Points A and B, when $V_{\text{IN}} = 0.5\text{V}$ and $V_{\text{Y}} = 0.5\text{V}$, define the time difference in which the high-to-low propagation delay is measured, and in this particular ring, it is 5.44ps. It can be seen that during this period, the nFET of II turns on, and sinks current $I_{\text{DS,n}}$ to discharge the capacitances $C_{\text{IN}}$ and $C_{\text{OUT}}$, lowering the potential of $V_{\text{Y}}$ from point C to B.
The lumped capacitances $C_{IN}$ and $C_{OUT}$ represent the input capacitance $II$ and the output capacitance of $II$ respectively, and their capacitance is dependent on their device’s mode of operation. To determine whether the change in device capacitance is negatively affecting the delay estimate $t_{pHL_{inv,chain}}$, Table 4-12 shows $C_{IN}$ and $C_{OUT}$ as they change over the time interval defined by points A and B.
Table 4-12 Device Capacitances $C_{IN}$ and $C_{OUT}$ Over the H-to-L Delay Time Interval

<table>
<thead>
<tr>
<th>Time (ps)</th>
<th>$I2$’s $C_{IN}$ (fF)</th>
<th>$II$’s $C_{OUT}$ (fF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.211</td>
<td>6.13</td>
<td>5.10</td>
</tr>
<tr>
<td>9.212</td>
<td>6.14</td>
<td>4.07</td>
</tr>
<tr>
<td>9.213</td>
<td>6.18</td>
<td>3.95</td>
</tr>
<tr>
<td>9.214</td>
<td>6.23</td>
<td>3.93</td>
</tr>
<tr>
<td>9.215</td>
<td>6.44</td>
<td>3.85</td>
</tr>
<tr>
<td>9.216</td>
<td>7.30</td>
<td>3.72</td>
</tr>
<tr>
<td>Average</td>
<td>6.40</td>
<td>4.10</td>
</tr>
</tbody>
</table>

As presented in section 4.2.3, the values of these capacitances were determined when $V_{IN} = 0.5\text{V}$ and $V_T = 0.5\text{V}$, when all pertaining devices are operating in saturation. This method resulted in a $C_{IN}$ and $C_{OUT}$ of 7.53fF and 3.74fF respectively. Hence, $C_{IN}$ was underestimated by 15\% and $C_{OUT}$ was overestimated by 10\% when comparing with the average values of $C_{IN}$ and $C_{OUT}$ listed in Table 4-12. Clearly, this error cannot be the sole cause for the underestimation in propagation delay.

The next possible source of error in $t_{PHL_{inv\_chain}}$ is the average on-resistance estimation. From section 4.2.3, $R_{on,n}$ was determined from the reciprocal of the slope of the line determined by points B and C, as shown by the nFET’s $I_{DS}$-$V_{DS}$ curve in Figure 4-27 for a voltage step at $V_{IN}$. In reality, $V_{IN}$ has finite rise (and fall) time, and if the response of $V_T$ and $I_{DS,n}$ of Figure 4-26 is plotted in an $I_{DS}$-$V_{DS}$ curve for $II$’s nFET, the shape is quite different as Figure 4-27 illustrates. The line B-to-C on the actual transient response of inverter’s nFET does not appear to be a good estimation of the average on-resistance.
This discussion leads to the conclusion that the source of error in the theory of the average propagation delay for an inverter in a ring is attributed to underestimating the FET's average on-resistance. However, since this error is nearly consistent with different device finger widths, the delay theory is still useful in determining the optimum ring design for maximum oscillation frequency.

**Figure 4-27 FET $I_{DS}$ vs $V_{DS}$ for Voltage Step and Oscillation Input**

This diagram illustrates the comparison of $I_{DS}$ vs $V_{DS}$ for voltage step and actual oscillation inputs, highlighting the difference in performance and the effect of underestimating the FET's on-resistance.
4. Ring Oscillator

4.8. Summary

This chapter contains a demonstration of the new methodology with the design and optimization of an 11-stage ring oscillator circuit in a 90nm SOI CMOS technology. For a CMOS inverter with a 2-finger nFET device and a 4-finger pFET device, there are two main factors which influence its propagation delay in a ring oscillator as the device finger width is varied. The first factor is the capacitive wiring parasitic, which loads the inverter’s output and occurs when device finger widths are too small. The second factor is the inverter’s input parasitics, which loads the output of the proceeding inverter and occurs when device finger widths are too large. For this CMOS process, an optimum device finger width of 1.4μm for minimizing the inverter’s delay in a ring is found with the new EDA layout synthesis tool and is verified analytically. Part of the equation for the inverter’s delay is formulated from the stepped input I_{DS}-V_{DS} response of the FET, which underestimates the average on-resistance because the transient ring response is quite different due to the finite rise and fall times effect on I_{DS}-V_{DS}. Finally, small-signal high-frequency figures of merit such as inverter unity current gain frequency and inverter unity power gain frequency are developed and provided some insight into the large signal propagation delay performance of the inverter. When the unity current gain frequency degrades with small device finger widths, the inverter’s delay increased. When the unity power gain frequency decreases below the unity current gain frequency, the inverter’s delay increased.
5. Summary and Conclusions

5.1. Summary

The first phase in this new RF mixed-signal design methodology was the development of the IBM EDA synthesis tool’s class architecture database for automatic layout generation of a CMOS inverter in a 90nm SOI technology. With object-oriented programming, the designer defined and designed the inverter’s C++ hierarchal layout data structure (C++ class). Furthermore, since the synthesis tool accesses a technology’s design rules at run-time, the generated inverter layout will never become obsolete as CMOS production moves from the 90nm to the 65nm process.

The second phase in this new RF mixed-signal design methodology was the optimization of an 11-stage ring oscillator with the EDA synthesis tool. Automatic ring layout generation allowed the designer to examine different layout variations, which led to the discovery of an optimum device finger width for minimizing the ring’s propagation delay in the presence of wiring parasitics. Related theories such as inverter unity current gain frequency, inverter unity power gain frequency and inverter delay in a ring were developed for verification of the new EDA synthesis tool. Since part of the delay’s equation was derived from the stepped input $I_{DS}$-$V_{DS}$ response of the FET, it was discovered that the average on-resistance was underestimated because the transient ring response was quite different due to the finite rise and fall times effect on $I_{DS}$-$V_{DS}$.
5. Summary and Conclusions

5.2. Conclusions

The design and optimization of the 11-stage ring oscillator in 90nm SOI CMOS showed a number of advantages of this new RF mixed-signal design methodology. First, the ring’s design must be layout-oriented for nanometre technologies because of the wiring parasitics. This was verified by the developed theories for inverter delay in a ring, inverter unity current gain frequency and unity power gain frequency. Second, object oriented C++ programming resulted in a time-efficient and cost-effective layout design stage. Third, layout synthesis with the EDA tool was a very time-efficient means of optimizing the ring. Lastly, familiar steps in the design flow will lead to easier designer acceptance of this proposed methodology.

5.3. Future Work

In the area of circuit analysis, a closer examination of transistor average on-resistance should be conducted. As discussed in section 4.7, the stepped input $I_{DS}-V_{DS}$ response of the FET underestimates the average on-resistance because the transient $I_{DS}-V_{DS}$ response is quite different. If this transient response can be modelled, it would lead to a more accurate delay, and hence oscillation frequency, prediction for a ring composed of CMOS inverters.

In the area of circuit design, further demonstrations of the new methodology and development of the new EDA synthesis tool should be done. This can be achieved with the design and optimization of other RF mixed-signal circuits such as high-speed sample-
and-hold circuits, analog-to-digital converters, digital-to-analog converters and phase-locked loop circuits.

Finally, in the area of EDA, integrating the new layout synthesis tool with a simulator and extractor should be developed. This can lead to a complete push-button automation of RF mixed-signal IC design.
6. References


6. References


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


6. References

