FREQUENCY SYNTHESIS IN WIRELESS AND WIRELINE SYSTEMS

A Dissertation

by

DIDEM ZELIHA TÜRKER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2010

Major Subject: Electrical Engineering

FREQUENCY SYNTHESIS IN WIRELESS AND WIRELINE SYSTEMS

A Dissertation

by

DIDEM ZELIHA TÜRKER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Edgar Sánchez-Sinencio
Committee Members,	Jose Silva-Martinez
	Costas N. Georghiades
	Sıla Çetinkaya
Head of Department,	Costas N. Georghiades

December 2010

Major Subject: Electrical Engineering

ABSTRACT

Frequency Synthesis in Wireless and Wireline Systems. (December 2010)
Didem Zeliha Türker, B.S., Sabanci University
Chair of Advisory Committee: Dr. Edgar Sánchez-Sinencio

First, a frequency synthesizer for IEEE 802.15.4 / ZigBee transceiver applications that employs dynamic True Single Phase Clocking (TSPC) circuits in its frequency dividers is presented and through the analysis and measurement results of this synthesizer, the need for low power circuit techniques in frequency dividers is discussed.

Next, Differential Cascode Voltage-Switch-Logic (DCVSL) based delay cells are explored for implementing radio-frequency (RF) frequency dividers of low power frequency synthesizers. DCVSL flip-flops offer small input and clock capacitance which makes the power consumption of these circuits and their driving stages, very low. We perform a delay analysis of DCVSL circuits and propose a closed-form delay model that predicts the speed of DCVSL circuits with 8% worst case accuracy. The proposed delay model also demonstrates that DCVSL circuits suffer from a large low-to-high propagation delay (τ_{PLH}) which limits their speed and results in asymmetrical output waveforms. Our proposed enhanced DCVSL, which we call DCVSL-R, solves this delay bottleneck, reducing τ_{PLH} and achieving faster operation.

We implement two ring-oscillator-based voltage controlled oscillators (VCOs) in 0.13μm technology with DCVSL and DCVSL-R delay cells. In measurements, for the same oscillation frequency (2.4GHz) and same phase noise (-113dBc/Hz at 10MHz), DCVSL-R VCO consumes 30% less power than the DCVSL VCO. We also use the proposed DCVSL-R circuit to implement the 2.4GHz dual-modulus prescaler of a low power frequency synthesizer in 0.18μm technology. In measurements, the synthesizer exhibits -135dBc/Hz phase noise at 10MHz offset and 58μm settling time with 8.3mW power consumption, only 1.07mW of which is consumed by the dual modulus prescaler and the buffer that drives it. When compared to other dual modulus prescalers with similar division ratios and operating frequencies in literature, DCVSL-R dual modulus prescaler demonstrates the lowest power consumption.

An all digital phase locked loop (ADPLL) that operates for a wide range of frequencies to serve as a multi-protocol compatible PLL for microprocessor and serial link applications, is presented. The proposed ADPLL is truly digital and is implemented in a standard complementary metal-oxide-semiconductor (CMOS) technology without any analog/RF or non-scalable components. It addresses the challenges that come along with continuous wide range of operation such as stability and phase frequency detection for a large frequency error range. A proposed multi-bit bidirectional smart shifter serves as the digitally controlled oscillator (DCO) control and tunes the DCO frequency by turning on/off inverter units in a large row/column matrix that constitute the ring oscillator. The smart shifter block is completely digital, consisting of standard cell logic gates, and is capable of tracking the row/column unit availability of the DCO and shifting multiple bits per single update cycle. This enables fast frequency acquisition times without necessitating dual loop filter or gear shifting mechanisms.

The proposed ADPLL loop architecture does not employ costly, cumbersome DACs or binary to thermometer converters and minimizes loop filter and DCO control complexity. The wide range ADPLL is implemented in 90nm digital CMOS technology and has a 9-bit TDC, the output of which is processed by a 10-bit digital loop filter and a 5-bit smart shifter. In measurements, the synthesizer achieves 2.5GHz-7.3GHz operation while consuming 10mW/GHz power, with an active area of 0.23 mm^2.

To my mom Hülya, my dad Yalvaç, and my sister Dilsad

&

To Zeki and Maya

ACKNOWLEDGMENTS

I was very lucky to be surrounded by wonderful people who helped and supported me throughout my studies at Texas A&M University. I am grateful to my advisor Dr. Edgar Sánchez-Sinencio, for his invaluable guidance, encouragement and support throughout the years. His vision, experience, knowledge and enthusiasm in research and in life, has been and will be a constant source of inspiration for me.

I would like to thank my advisory committee Dr. Jose Silva-Martinez, Dr. Costas N. Georghiades and Dr. Sıla Çetinkaya for their valuable support, discussion and suggestions.

I am very grateful to Dr. Yusuf Leblebici, my academic advisor during my undergraduate studies, for introducing me to the exciting area of microelectronics and circuit design. His wonderful teaching and encouragement is the reason why I decided to continue my studies and dared to start working towards a Ph.D. in this field.

I would like to thank Dr. Alexander Rylyakov from IBM T.J. Watson Research Center, for his valuable guidance. His excitement for his work, openness and desire to share his wealth of knowledge with others, is such an inspiration for me and his feedback and discussions have been an invaluable source of support in my research.

I would also like to thank Dr. Sunil P. Khatri for his valuable discussions and suggestions. It was in his class, that we came up with an idea that motivated and enabled a valuable portion of my research. I am humbled by his kindness and support.

I thank Texas Instruments for their kind support throughout my Ph.D.

My years of studies at the AMSC would not have been the same without the wonderful Ms. Ella Gallagher. Her spirit has always been a constant source of joy in our office and her strict rules and planning creates an organized and successful office environment for the students and the professors in our group. As I leave AMSC, I will miss her dearly.

I would like to thank Tammy Carda, our department's senior academic advisor, for her valuable help and kindness throughout the years.

I had the opportunity to meet many wonderful colleagues at the AMSC throughout the years. I would like to thank Rangakrishnan Srinivasan, whom I worked with in several projects during my Ph.D. and who has been a wonderful friend to me for many years. I thank Sang Wook Park for the many discussions we had on our research and for his valuable support and feedback. I also thank Alberto Valdes-Garcia for the many valuable discussions we had but most importantly for being such a dear friend. I thank Sathya Venkatesh for being such a good friend whose strength, positivity and wit have always impressed me. I would also like to thank my friends and colleagues Felix O. Fernandez, Hesam Aslanzadeh, Raghavendra Kulkarni, Manisha Gambhir, Chinmaya Mishra, David Hernandez, Faisal Hussien, Erik Pankratz, Jorge Zarate, Mohammed Mohsen, Seenu Gopalraju and Adrian Colli-Menchi. They have not only impressed me with their brightness, but also have been good friends to me.

I also met wonderful friends in College Station throughout the years. I would like to thank Baris Gunersel, Paola Guerrero and Selcuk Dincal for their beautiful friendship and support. We shared unforgettable memories and College Station would not have been the same without them.

I would like to thank my friends from Turkey; Hanife, Buke, Kivilcim, Burcu, Cigdem, Mehmet, Tuba, Mukaddes, Baris and Emrah whose support throughout the years meant a lot to me.

I would like to thank my family. I am grateful to my parents Hülya and Yalvaç Türker and my sister Dilsad Karacabey, whose unconditional love and support is behind every achievement I had in my life, big or small. They always made sure that I know, that wherever I am, however far I may be from them, they are always with me, in my heart. Anything that I accomplish in my life is because of them.

I thank my dear nephew Kaan and my little nephew that I can't wait to welcome into the world, and my brother in law Tolga Karacabey, for their love and support. I thank my dear Lucy; her memory will always be in my heart and I will never forget that she was what I dreamt for throughout my childhood. I miss her everyday. I would also like to thank my late grandfather Mustafa Özcan; his hardwork, kindness and his love for life and his family has always inspired me to push myself in my career and in life. My late grandfather Ramazan Türker for teaching me that true success comes from being hardworking, honorable and honest. Their memories are always with me. I also thank my grandmother and my namesake Zeliha Türker, for always believing in me and her endless love and support. My grandmother Nebahat Özcan for showing me that kindness and love are the key behind a happy life and family. I thank all of them with all my heart.

Finally, I would like to thank my dear Zeki and my lovely Maya. Maya's unconditional love, innocence, sweetness and her wonderful heart has been a constant reminder of what the most important things in life are. I thank Zeki, with all my heart, for being my constant source of support, for his endless love, kindness and patience. Words cannot express how grateful I am that they are in my life, and how much I love them. I know that when I am with them, I am home.

TABLE OF CONTENTS

Page

ABSTRACT . . . iii

ACKNOWLEDGMENTS . . . vi

TABLE OF CONTENTS . . . ix

LIST OF FIGURES . . . xi

LIST OF TABLES . . . xviii

CHAPTER

I INTRODUCTION . . . 1

A. Motivation and Contributions . . . 1
B. Overview . . . 4

II A FULLY INTEGRATED FREQUENCY SYNTHESIZER FOR ZIGBEE APPLICATIONS . . . 6

A. IEEE 802.15.4 / ZigBee . . . 6
B. Frequency Synthesis for a ZigBee Transceiver . . . 7
C. Synthesizer Implementation . . . 10
D. Frequency Dividers . . . 18
1. Divider Basics . . . 18
a. Divide-by-2 Operation . . . 18
b. Division By an Odd Ratio . . . 20
c. Dual Modulus Division . . . 22
2. Programmable Divider . . . 24
3. Dual Modulus Prescaler Implementation . . . 27
4. Frequency Divider Buffers . . . 32
E. Measurement Results . . . 34
1. Discussion on Power Consumption . . . 37

III FREQUENCY DIVIDER CIRCUITS AND A NEW DCVSL-R DELAY CELL . . . 40

A. Frequency Divider Circuit Techniques . . . 40
B. Differential Cascode Voltage Switch Logic Circuits . . . 42
C. A Delay Model for DCVSL Circuits . . . 45
1. Analysis and Derivation . . . 45

CHAPTER Page

2. Model Accuracy 52
3. Process Variations 55
4. Discussion on DCVSL Delay Behavior 55
D. Proposed DCVSL-R Circuit 57

IV A LOW POWER FREQUENCY SYNTHESIZER WITH DCVSL-R DIVIDERS 63

A. Implementation 63
B. Measurement Results 67
C. Discussion on Divider Performance 71

V RING OSCILLATORS USING DCVSL AND DCVSL-R DELAY CELLS 75

A. Ring Oscillator Design 76
B. Measurement Results 77
C. Performance Evaluation 86

VI ALL DIGITAL PHASE LOCKED LOOPS 88

A. Background and Motivation 88
B. DPLL Basics 89
C. Noise in ADPLLs 95
D. Design Challenges 99

VII A WIDE RANGE ALL DIGITAL PLL 102

A. Previous Work 102
B. Proposed System Design 104
1. Phase Frequency Detection 108
C. Digitally Controlled Oscillator 109
D. Smart Shifter 112
1. Bidirectional Shifting 118
E. Digital Loop Filter 121
F. Other ADPLL Building Blocks 127
G. System Simulations 128
H. Measurement Results 131
1. DCO Measurements 133
2. ADPLL Measurements 141
I. Layout Techniques in the ADPLL 146
1. Standard Cell Design 147

CHAPTER Page

2. Power Supply Distribution 150
3. DCO Layout 154

VIII CONCLUSIONS 157

A. Summary 157
B. Contribution and Impact 159

REFERENCES 162

APPENDIX A 173

APPENDIX B 190

VITA 200

LIST OF FIGURES

FIGURE Page

1 Block diagram of a standard transceiver system 9

2 Block diagram of the ZigBee frequency synthesizer 12

3 Second order loop filter of the charge-pump based PLL 13

4 Block diagram of a divide-by-2 frequency divider 19

5 Divide-by-2 operation (a) state diagram (b) input and output timing 20

6 Block diagram of a divide-by-3 frequency divider 21

7 State table of a divide-by-3 frequency divider 22

8 Block diagram of a dual modulus divide-by-3/4 frequency divider . . 23

9 Block diagram of the NOR based divide-by-3/4 circuit 24

10 Block diagram of a pulse-swallow programmable divider 25

11 Block diagram of the P counter and S block 27

12 Dual modulus (15/16) prescaler block diagram 28

13 State table of the 15/16 prescaler 29

14 Circuit-level simulations of glitch in divide-by-3/4 circuit at 2.4GHz operation (a) using regular TSPC logic (b) using glitch-free TSPC logic 30

15 Post-layout simulations of 15/16 prescaler circuit at 2.48GHz operation 31

16 Schematic of the differential to single ended buffer, the bias-T circuit to set proper common mode level and the first inverter of the inverter chain buffer 32

17 Die micrograph of the frequency synthesizer 34

FIGURE Page

18 Measured output spectrum of the synthesizer demonstrating first channel of ZigBee 35

19 Measured output spectrum of the synthesizer for channel 16 of ZigBee 36

20 Phase noise spectrum of the frequency synthesizer 36

21 Pie chart of the measured power consumption distribution in the ZigBee synthesizer 38

22 Pie chart of the power consumption distribution in the ZigBee synthesizer with individual frequency divider blocks 39

23 Schematic of a CML latch 41

24 Schematic of a TSPC D-flip-flop 42

25 Schematic of a DCVSL inverter 43

26 Two-clock-phase DCVSL flip-flop 45

27 DCVSL inverter setup for transient delay analysis 46

28 Propagation delay derivation for τ_{PHL} 47

29 Propagation delay derivation for $\tau_{PLH} = t_1 + t_2$ 49

30 Approximation of t_2 50

31 Comparison of calculated vs. simulated values of τ_{PLH} and τ_{PHL} (a) for (WP/WN)=1.33 in 0.18μm technology (b) for (WP/WN)=1.57 in 0.13μm technology 53

32 Simulated voltage and current waveforms of a DCVSL inverter in 0.18μm for WP/WN=3, demonstrating mid-transition slow-down . . 56

33 Proposed DCVSL-R circuit 57

34 Inverter output waveforms in a ring oscillator setting for WP/WN=1 (a) for conventional DCVSL (b) for proposed DCVSL-R with R=380 ohms 59

FIGURE Page

35 Circuit-level simulation results for τ_{PLH}, τ_{PHL} and τ_{TOTAL} values vs. the resistance R for a DCVSL-R inverter with (WP/WN)=1.66 in 0.18μm technology 60

36 Circuit-level simulation of propagation delay vs. the resistance R for a DCVSL-R inverter with (WP/WN)=1.66 in 0.18μm technology for values of R past the point of symmetry 62

37 Block diagram of the new PLL with DCVSL-R divider 64

38 Circuit level diagram of D flip-flops used in the DCVSL-R based prescaler 66

39 Layout of the DCVSL-R based dual modulus (15/16) prescaler . . . 66

40 Output frequency spectrum of the new synthesizer with DCVSL-R dividers at 2.405GHz 68

41 Phase noise spectrum of the new synthesizer at 2.405GHz 69

42 New frequency synthesizer measured phase noise spectrum at 2.48GHz 69

43 Die micrograph of the new PLL 70

44 New frequency synthesizer measured settling time 70

45 Power consumption distribution of the synthesizer with TSPC dividers and the new synthesizer with DCVSL-R dividers 74

46 Block diagram of the three stage ring oscillators 75

47 VCO delay cells (a) conventional DCVSL for OSC1 (b) proposed DCVSL-R for OSC1-R 76

48 Measured fine and coarse tuning range of OSC1 at 1.2V supply . . . 78

49 Measured fine and coarse tuning range of OSC1-R at 1.2V supply . . 78

50 Measured output frequency spectrum of OSC1 at 2.4GHz operation . 79

51 Measured phase noise spectrum of OSC1 at 2.4GHz operation 79

FIGURE Page

52 Measured output frequency spectrum of OSC1-R at 2.4GHz operation 80

53 Measured phase noise spectrum of OSC1-R at 2.4GHz operation . . . 80

54 Output frequency spectrum of OSC1-R at 3.12GHz and 2.8mW power 82

55 Phase noise spectrum of OSC1-R at 3.12GHz and 2.8mW power . . . 82

56 Ring VCO measured power vs. frequency curves for OSC1 and OSC1-R 83

57 Layout of OSC1 (based on DCVSL) 85

58 Layout of OSC1-R (based on DCVSL-R) 85

59 Die micrograph of OSC1 and OSC1-R 86

60 Block diagram of a conventional DPLL 89

61 A conventional proportional integral digital loop filter and DCO control interface 91

62 Block diagram of the proposed all digital PLL 105

63 System level diagram of the time to digital converter 108

64 Three stage ring oscillator based DCO 3-D representation 110

65 Three stage ring oscillator based DCO put in a row-column matrix for ease in control 111

66 Block diagram of a conventional 3-bit barrel shifter 112

67 Operational flow diagram of the smart shifter 113

68 Block diagram of a 3-bit implementation of the proposed smart shifter 114

69 Generation of MC controls and rowshift signals in 3-bit smart shifter 115

70 Sample operation of the 3-bit smart shifter 117

71 Block diagram and sample operation of right-shifting using the left-shift smart shifter 119

FIGURE Page

72 Block diagram of the complete bidirectional row/column shifter as the DCO interface 120

73 Transistor level implementation of a Manchester Carry Chain 122

74 A 1-bit full adder using Manchester Carry Chain 123

75 Block diagram of the 3-bit Carry-Skip Adder used in the loop filter . 124

76 Block diagram of the loop filter 125

77 Block diagram of an adder/subtractor 126

78 Simulink time-domain simulations of the ADPLL, DCO total control word for ADPLL operation frequencies between 2GHz-7GHz . . 129

79 Simulink time-domain simulations of the ADPLL, detail of total and coarse DCO control words for 4GHz operation 130

80 Simulink time-domain simulations of the ADPLL, TDC output at 6GHz operation 130

81 Measurement instruments and setup for ADPLL 131

82 Printed circuit board of ADPLL with connecting cables 132

83 Printed circuit board of ADPLL 133

84 DCO Output frequency spectrum - minimum frequency for VDD=1V 135

85 DCO Output frequency spectrum - maximum frequency for VDD=1V 135

86 DCO Output frequency spectrum - minimum frequency for VDD=0.9V 136

87 DCO Output frequency spectrum - maximum frequency for VDD=0.9V 136

88 DCO wide span output frequency spectrum at 7.3GHz operation . . 137

89 DCO phase noise spectrum for 7.3GHz operation 138

90 DCO phase noise spectrum for 6.24GHz operation 138

91 DCO phase noise spectrum for 5.8GHz operation 139

FIGURE Page

92 DCO phase noise spectrum for 4.58GHz operation 139

93 DCO phase noise spectrum for 2.11GHz operation 140

94 ADPLL output and its period histogram at 6GHz operation with 1.9ps rms period jitter 143

95 ADPLL output and its period histogram at 3.6GHz operation with 4.2ps rms period jitter 143

96 ADPLL output phase noise spectrum for 4GHz operation 144

97 ADPLL die micrograph 145

98 Layout of the ADPLL active area implemented in 90nm digital CMOS 146

99 Demonstration of standard library cell layout 148

100 Standard library cell layout examples Height=6.75 μm (a)a 3-input AND gate (b) a current starved inverter 149

101 Layout of a full adder consisting of standard cells 150

102 Supply grid with horizontal metal 9 and vertical metal 8 layers . . . 152

103 Digital loop filter layout with power supply grid (89 μm x 94 μm) . . 152

104 Layout of the ADPLL chip with pads and decoupling capacitors . . . 153

105 Detailed view of power supply routing from pads 153

106 Layout of a single DCO row/column matrix unit 155

107 Layout of the three stage DCO ring unit 156

108 Block diagram of a PLL with a prescaling divider before the pulse-swallow divider 175

109 Bode plot of the open loop gain of the third order PLL 187

110 Bode plot of the open loop gain of the second order approximation of the PLL 187

FIGURE Page

111 Closed loop frequency response of the third order PLL 189

112 Closed loop frequency response of the second order approximation of the PLL 189

113 Step response of the closed loop ADPLL 198

114 Frequency response of the closed loop ADPLL 198

LIST OF TABLES

TABLE Page

I List of various wireless receivers and their FS power consumption . . 8

II Performance specifications for a ZigBee frequency synthesizer 10

III Charge pump based PLL building block transfer functions for second order continuous approximation linear analysis 15

IV Summary of the PLL second order loop parameters 16

V Useful relations between second order approximation loop parameters 16

VI Summary of loop parameters used in the fabricated ZigBee frequency synthesizer prototype 17

VII Summary of the pulse-swallow divider parameter values 26

VIII Measured performance of the ZigBee frequency synthesizer 37

IX Values of DCVSL delay model empirical correction factors 52

X A list of calculated and simulated values of τ_{PLH}, τ_{PHL} and model error for various transistor configurations 54

XI Measured performance summary of the frequency synthesizer 67

XII Performance comparison of the DCVSL-R prescaler with previously reported solutions 72

XIII Measured performance summary of OSC1 (based on Fig.47(a)) and OSC1-R (based on Fig.47(b)) 84

XIV Performance comparison of OSC1-R with previously reported solutions 87

XV List of DPLL loop parameters 94

XVI Summary of the DPLL important second order loop expressions . . . 95

TABLE Page

XVII Summary of the loop parameters of the implemented ADPLL 106

XVIII Truth table of 2-bit controlled MUX 115

XIX Measured DCO power supply level and tuning range 134

XX Measured DCO figure of merit for various frequencies 141

XXI ADPLL measured performance summary 145

XXII Damping factor and pole locations in a second order system 179

XXIII Closed loop bandwidth of the designed ADPLL based on its frequency response 197

CHAPTER I

INTRODUCTION

A. Motivation and Contributions

The present advancements in the information technology are driven by the developments and innovations in Integrated Circuit Design Techniques. Small laptops with high computational powers, wireless internet and information transfer facilities, cell phones and many other electronic devices that we use in daily life rely on the efficient implementations of communication circuits, receiver/transmitter radios on silicon. These receiver and transmitter circuits require the use of phase locked loops (PLL) for down/up conversion of the data carrying signal in wireless transceiver applications and for clock generation in serial link and microprocessor applications. This dissertation focuses on the design, analysis and implementation of these phase locked loop based frequency synthesizers and clock generators as well as their building blocks.

The frequency synthesizer is one of the key elements of a wireless transceiver. Several performance parameters of the synthesizer such as phase noise, frequency spurs, settling time, has considerable effect on the overall wireless system behavior. Power consumption performance of a wireless transceiver determines its battery life. Active during both transmit and receive modes, the frequency synthesizer has significant contribution to the overall power consumption of the transceiver. Particularly, the synthesizer employs several frequency dividers that operate at RF channel frequency, making the design of this block a challenge for low-power wireless transceiver applications.

This dissertation follows the style of *IEEE Journal of Solid-State Circuits.*

Along with frequency of operation and technology speed, the circuit topology is key in determining the power consumption of frequency dividers. Until recently, Current Mode Logic (CML) circuits were widely employed in the frequency dividers of synthesizers [1], [2], [3] due to their capability of high speed operation. With the migration towards sub-micron technologies, digital dynamic-circuit techniques such as True-Single-Phase Clocking (TSPC) are becoming popular [4], [5] to optimize the power consumption of high-speed frequency dividers.

In this dissertation, first, the design, implementation and measurements of a frequency synthesizer that employs TSPC based frequency dividers will be presented where the goal is to provide a low power solution for an IEEE 802.15.4 / ZigBee [6] transceiver application. It will be demonstrated that the implementation of the frequency dividers is crucial in minimizing the power consumption of the frequency synthesizer. A discussion on high speed circuit techniques to implement the RF frequency dividers of a frequency synthesizer will be presented.

Later, we focus on a logic family called Differential Cascode Voltage-Switch Logic (DCVSL) as a candidate to implement the RF dividers of a frequency synthesizer. The key benefits of DCVSL are its low input capacitance, differential nature, and low power consumption. However, DCVSL delay cells have a delay bottleneck; their low-to-high-transition propagation delay (τ_{PLH}) is inherently larger than their high-to-low-transition propagation delay (τ_{PHL}). The large τ_{PLH} presents a speed bottleneck for the DCVSL cells and results in asymmetric differential output waveforms where the rising output lags the falling output. While the discrepancy between the two differential outputs is addressed in a few earlier works [7], [8], a detailed analysis of the inherent delay problem is not presented.

We analyze the delay behavior of DCVSL inverters and propose a closed-form delay model to characterize and predict the delay behavior of DCVSL circuits and demonstrate their inherent speed bottleneck. Then, we propose a circuit solution, which we term Differential Cascode Voltage Switch Logic with Resistive-enhancement (DCVSL-R), to overcome this speed bottleneck. We explore the use of the proposed circuit in the delay cells of ring oscillators to improve the power-consumption and speed trade off in these circuits and provide a comparison of DCVSL and DCVSL-R based ring oscillators through measurements. We also implement a fully integrated frequency synthesizer using the proposed DCVSL-R in its high speed frequency dividers, for low power 2.4GHz band wireless transceiver applications and present measurement results of this low power frequency synthesizer.

Analog PLLs have been widely used in communication systems. However, as the smaller, deep sub-micron technologies enable the shrinking of digital circuits, the design of analog intense circuits become more challenging. An all digital approach to implement the PLL, which is an integral part of communication systems, would enable the benefits of technology scaling in terms of low area and low voltage and will increase the integration capability of the PLL with the rest of the digital circuitry. If the PLL is implemented in an all-digital manner, the expensive need for special mixed signal processes can also be eliminated.

In this work, an all digital PLL (ADPLL) that addresses the speed and performance demands of today's wireline and microprocessor applications is designed and fabricated. The proposed ADPLL is truly digital, using a standard bulk CMOS technology (UMC 90nm CMOS) and does not require any analog/RF or non-scalable R/L/C components. The ADPLL achieves the synthesis of a wide range of output frequencies, (2.5GHz - 7.3GHz in measurements), to serve as a multi-standard compatible PLL. It addresses the challenges that come along with wide range of operation such as stability and phase frequency detection for a large frequency error range. The proposed loop accommodates a multi-bit linear time-to-digital converter (TDC) and avoids the use of digital-to-analog converters (DACs) or binary-to-thermometer (B-T) converter circuits. A proposed all digital digitally-controlled oscillator (DCO) control block, that we refer to as the Smart Shifter, facilitates faster frequency tuning per loop cycle for the wide-range PLL while minimizing implementation complexity.

B. Overview

Chapter II presents the design and analysis of a fully integrated frequency synthesizer with TSPC frequency dividers, that targets 2.4GHz IEEE 802.15.4 / ZigBee transceiver applications, with a focus on the design issues of the frequency dividers. In this chapter, we discuss frequency divider basics and present measurement results of the frequency synthesizer that was fabricated in 0.18 μm CMOS technology.

Chapter III discusses various circuit topologies and offers DCVSL circuits as a candidate to implement the RF frequency dividers of frequency synthesizers. This chapter presents a delay analysis, that characterizes the operation of and pinpoints the key speed bottleneck of, DCVSL circuits. This chapter also proposes a circuit technique, DCVSL-R, which improves the speed and power consumption performance trade off of DCVSL circuits and fixes their output asymmetry.

Chapter IV presents a low-power frequency synthesizer, the programmable dividers of which are implemented with the proposed DCVSL-R circuit. This chapter provides measurement results of the frequency synthesizer that was fabricated in 0.18 μm CMOS technology and a comparison of the presented frequency divider with similar frequency dividers that are reported in literature.

Chapter V discusses the implementation of two ring-oscillator-based voltage controlled oscillators (VCO) that utilize DCVSL and DCVSL-R delay cells that are fabricated in 0.13 μm CMOS technology. This chapter demonstrates the performance improvement of the latter, through measurement results. A comparison of the proposed DCVSL-R based ring oscillator with other state-of-the art ring oscillators in literature, is also presented.

Chapter VI provides an analysis of all digital PLLs. A discussion on the motivation of moving the PLLs into digital domain is presented, along with the loop analysis of an ADPLL, a discussion on noise in ADPLLs and a summary of design challenges.

In Chapter VII, we present a wide range ADPLL and discuss the proposed loop architecture as well as building block designs. This chapter also demonstrates system level simulations of the proposed ADPLL along with the measurement results of an ADPLL prototype that was fabricated in 90 nm digital CMOS technology.

Finally, Chapter VIII concludes this dissertation.

CHAPTER II

A FULLY INTEGRATED FREQUENCY SYNTHESIZER FOR ZIGBEE APPLICATIONS

A. IEEE 802.15.4 / ZigBee

Wireless networking has become an integral part of everyday life. In the last decade, machine to machine sensor networks and remotely controlled wireless communication systems became popular. Machine to machine systems connect and network household appliances, air conditioners, heat sensors, gas sensors or simply RFID tags for tracking purposes. The basic idea behind these applications is to eliminate the user effort and try to form a network between the machine systems for environmental control, health monitoring or security issues. Remotely controlled communication systems are similar but involve the user end, where a user can create a household network to control everything from the garage door to alarm systems. Similarly, a remote network could control the automation systems in an office building or campus such as security systems, etc.

Although there is a growing number of wireless communication standards today, none of them address such low-cost applications since they require complex circuitry and protocols with higher data rates (UWB, Bluetooth, Wi-Fi) or higher communication ranges (GPRS, GSM). Such standards address wireless communication platforms that target high performance where the transfer of voice, data, video occurs between networking nodes or involves very large distances.

IEEE 802.15.4 / ZigBee [6] is a wireless personal area network (WPAN) standard that specifically targets remote control and sensor monitoring applications. ZigBee defines a flexible networking system to accommodate up to tens of thousands of nodes/sensors in a single network to perform a vast range of remote controlling applications that arise in every day life in home or industrial environments such as automated meter reading, remote lighting systems, etc. ZigBee has low data rate (up to 250 kb/s depending on the frequency band) and short range specifications (1-100m) that enable the extreme low cost and long battery life.

IEEE 802.15.4 / ZigBee is defined over three frequency bands [6]. It has one channel in the European 868MHz band, 10 channels in the 915MHz ISM band and 16 channels in the 2.4GHz ISM band. In this work, we will focus on the 2.4GHz ISM band. In this band, ZigBee has 250kbps data rate, offset quadrature phase shift keying (OQPSK) modulation and 5MHz channel spacing [6], [9].

B. Frequency Synthesis for a ZigBee Transceiver

Since a ZigBee network could have thousands of nodes, such a large network can be feasible only through an extremely low cost wireless solution for each node, and would require an ease of implementation and maintenance of the system, requiring long battery lives measured in years. The battery life of a device is determined by its power consumption while the cost and size of it is determined by the area. With this motivation in mind, the emphasis of the design of a ZigBee transceiver (or any stand-alone building block developed for a ZigBee transceiver) is on minimizing the power consumption and minimizing complexity and area while meeting the ZigBee performance specifications.

The contribution of the frequency synthesizer to the overall power consumption of the transceiver is very significant due to the fact that the frequency synthesizer has multiple building blocks that operate at the highest RF frequency of the transceiver. Moreover, the frequency synthesizer is active during both receive and transmit modes, contributing to the overall power consumption of the device at all times.

To understand the effect of the frequency synthesizer power consumption in a wireless transceiver, Table I summarizes the total power consumption of the receiver, the power of the frequency synthesizer and its percentage in the receiver for several designs that target various different wireless standards. It is seen that the power consumption of the frequency synthesizer is a significant factor in determining the overall power of the receiver. Hence, any improvement and technique that would reduce the power consumption of the synthesizer will have a direct effect on the whole system power and the battery life of the device.

Table I. List of various wireless receivers and their FS power consumption

Receiver	**Wireless Standard**	**Receiver Power**	**FS Power**	**FS Power Percentage**
[10], [11]	Bluetooth IEEE 802.11b	69.75 mW (w/o ADC)	31.25 mW	44.8%
[12]	Ultrawideband	285 mW	200 mW	70%
[13]	Wireless LAN (IEEE 802.11a)	55.7 mW (w/o ADC)	20.5 mW	36.8 %

A frequency synthesizer is designed to be used in a fully integrated ZigBee transceiver as shown in Fig. 1. A direct conversion (zero-IF) receiver architecture provides many receiver system level benefits such as eliminating the need for image rejection [14]. From the synthesizer's point of view, a transceiver with direct-conversion receiver utilizes the same frequency synthesizer in both the transmit and receive paths which results in significant area savings. Therefore, the target transceiver architecture will be assumed a direct-conversion architecture.

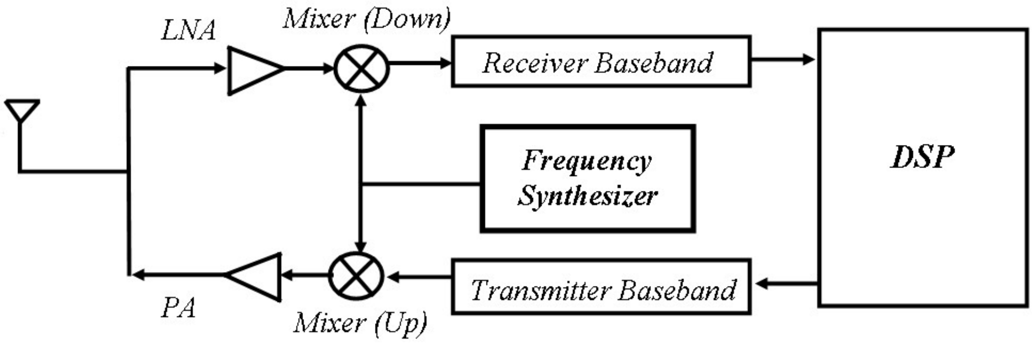


Fig. 1. Block diagram of a standard transceiver system

In the target 2.4GHz ISM band, ZigBee employs OQPSK with half-sine wave shaping. Due to the quadrature nature of the modulation, the transmit path will include in-phase (I) and quadrature (Q) up-conversion paths while the receiver will consist of I and Q down-conversion paths. Therefore, the ZigBee synthesizer should generate quadrature local oscillation (LO) signals to be compatible in a transceiver environment.

The design specifications of the frequency synthesizer should be derived from the standard specifications. For instance, the standard determines symbol rate (62.5 kilo-symbols / sec) as well as the receive to transmit turnaround time (duration of 12 symbols). This leads to the derivation of the synthesizer settling time of 192 μs. Similarly the standard defines the adjacent and alternate channel (5MHz and 10MHz away from the channel, respectively) interference test and this, along with the modulation scheme and the tolerable bit error rate, determine phase noise and spur suppression specifications.

Table II summarizes the ZigBee frequency synthesizer specifications. ZigBee standard requires 0dB adjacent channel interferer rejection while this specification for the alternate channel is 30dB [6]. This results in a tighter alternate channel spur suppression specification than the adjacent channel suppression, as seen in Table II. A detailed derivation of these specifications from the ZigBee standard is provided in [15], [16]. A detailed look at the derivation of frequency synthesizer specifications from a wireless standard, is given in [17], [11].

Table II. Performance specifications for a ZigBee frequency synthesizer

Performance Metric	**Value**
Frequency Synthesis	2.405GHz - 2.48 GHz
Channel Spacing	5MHz
Number of Channels	16
Settling Time	< 192 μs
Settling Accuracy	± 40ppm (96 kHz)
Spur Suppression	< -13dBc at 5MHz < -43dBc at 10MHz
Phase Noise	< -112dBc/Hz at 10MHz offset < -102dBc/Hz at 3.5MHz offset

C. Synthesizer Implementation

As discussed in Section A, the focus of the design of this ZigBee synthesizer is in keeping the implementation simple (low-cost) and having low-power consumption (long battery life). An integer-N based architecture is chosen due to its simplicity in implementation when compared to their fractional-N based counterparts. In an integer-N architecture the maximum reference frequency is determined by the greatest common divisor (GCD) of the channel frequencies and the channel spacing of the targeted wireless standard, as given in (2.1) [17].

$$F_{REF\_MAX} = GCD(F_O, F_{SP}) \tag{2.1}$$

where F_O is the channel center frequency and F_{SP} is the channel spacing. It is seen that channel spacing can also serve as the PLL's reference frequency. Therefore, a reference frequency of 5MHz is used in this design. The relationship between the output frequency and the reference frequency is given by:

$$FOUT = FREF \times N \tag{2.2}$$

where N is the frequency division ratio. Note that in a fully-integrated PLL solution, the reference frequency is often generated by a stable crystal oscillator [18] and is therefore constant. Then, (2.2) shows that the frequency synthesizer output tone can be controlled through changing the divider ratio.

The block diagram of the synthesizer is shown in Fig. 2. To meet the requirements of Table II, the synthesizer generates quadrature LO outputs for 16 channels, spaced with 5MHz, through the programmable frequency divider ratio N. The values of N are:

$$N = 481, 482, ..., 495, 496 \tag{2.3}$$

There are alternative solutions to generate quadrature components of the received / transmitted signal in a wireless radio such as using passive RC networks or active frequency dividers. While the use of active frequency dividers consumes additional power, it is usually preferred over passive solutions due to its minimal amplitude and

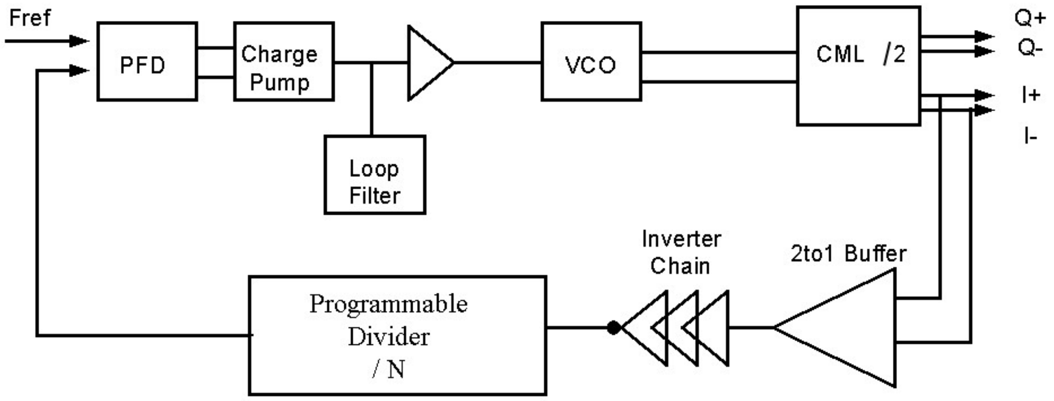


Fig. 2. Block diagram of the ZigBee frequency synthesizer

phase mismatch. Moreover, with the use of an active divide-by-2 circuit to generate IQ components of the carrier, the VCO operates at double the channel frequency and the LO output is generated by dividing the VCO output frequency by 2. This prevents injection pulling and PA load pulling problems that commonly occur in monolithic implementations of transceivers [14], [19].

The stability and frequency dependent behavior of the loop is analyzed in phase domain where the input of the system is defined as the phase difference between the reference and the divider output signals and the output of the system is defined as the phase of the PLL output signal. Note that frequency lock is a very nonlinear behavior. Therefore, for a linear analysis to apply, it is assumed that the input phase error of the PLL is small.

The frequency synthesizer is implemented as a type II, third order charge-pump based integer-N PLL in TSMC 0.18 μm CMOS technology [4]. The loop filter is shown in Fig. 3. The loop type is determined by the number of integrations [20]. In the PLL of Fig. 2, the two integrations come from the loop filter and from the VCO where frequency is converted into phase through integration.

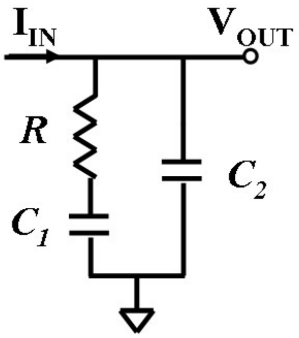


Fig. 3. Second order loop filter of the charge-pump based PLL

The transfer function of the loop filter shown in Fig. 3 is:

$$H_{LF}(s) = \frac{1}{(C_1 + C_2)} \frac{s/w_z + 1}{s(s/w_p + 1)} \tag{2.4}$$

where the zero and the pole created by the loop filter are given by (2.5) and (2.6).

$$w_z = \frac{1}{RC_1} \tag{2.5}$$

$$w_p = \frac{C_1 + C_2}{RC_1C_2} \tag{2.6}$$

The loop pole w_p occurs due to C_2 of the loop filter. This capacitor is added to the loop filter to minimize ripples on the VCO control line that arise due to the voltage drops on R. However, to maintain the stability of the system, this pole is often placed further than the loop zero and loop bandwidth. Then, the capacitor C_2

is often much smaller then C_1 such that:

$$w_p \approx \frac{1}{RC_2} \quad (C_1 >> C_2) \tag{2.7}$$

With a second order loop filter, and the integration that comes from the VCO, the PLL is a type II, third-order system. However, since w_p is placed much further than the loop bandwidth and for frequencies that are of interest, the loop behaves similar to a second order system. Then, the loop filter transfer function be approximated as follows:

$$H_{LF}(s) \approx \frac{1}{C_1}\frac{s/w_z + 1}{s} \tag{2.8}$$

which can be rewritten as:

$$H_{LF}(s) \approx R\frac{s + w_z}{s} \tag{2.9}$$

While analyzing the loop as a second order system is a valid approximation, for phase margin analysis, the placement of w_p should be considered to ensure stability. A detailed analysis on the third-order analysis of a PLL can be found in [21].

Table III summarizes the individual building block transfer functions in the phase-domain continuous approximation linear analysis of the PLL where $\Delta\phi_{in}$ is the phase difference at the PFD input, K_{VCO} is the VCO frequency gain in $radians/(second\times V)$, N is the feedback division ratio in the loop, I_{CP} is the charge pump current and w_z is the loop filter zero given in (2.5). Further information on the continuous approximation linear analysis of charge-pump based PLLs, the derivation of the below equations and the third-order loop analysis can be found in Appendix A and in [21–24].

Based on Table III, the second order approximation of the closed-loop transfer

Table III. Charge pump based PLL building block transfer functions for second order continuous approximation linear analysis

Building Block	**Transfer Function**
PFD and Charge Pump	$\frac{I_{out}}{\Delta\phi_{in}} = \frac{I_{CP}}{2\pi}$
Loop Filter	$\frac{V_{out}}{I_{in}} = R \times \frac{s + w_z}{s}$
Voltage Controlled Oscillator	$\frac{\phi_{out}}{V_{in}} = \frac{K_{VCO}}{s}$
Frequency Dividers	$\frac{\phi_{out}}{\phi_{in}} = \frac{1}{N}$

function of the PLL is given by:

$$H_{CL\_PLL}(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{(K_{LOOP} \times N)(s + w_z)}{s^2 + (K_{LOOP})\, s + K_{LOOP} w_z} \tag{2.10}$$

where

$$K_{LOOP} = \frac{K_{VCO} I_{CP} R}{2\pi N} \tag{2.11}$$

Note that N is the frequency division ratio in the feedback dividers. Any frequency division in the forward path should be separately incorporated in the loop transfer function to the forward path gain. Note that the units of K_{VCO} in this text is defined in $radians/(second \times V)$. A common mistake is to assume K_{VCO} in Hz/V and not take the 2π factor into account in the loop gain. If K_{VCO} is defined in Hz/V, then the VCO gain in the loop transfer function should be $2\pi K_{VCO}$ since it is a phase domain analysis. To avoid confusion, one should be careful to maintain consistency in the definition and units of the loop parameters.

Based on (2.10), important loop parameters such as the natural frequency (w_n), the damping factor (ξ) of the system and the closed loop 3-dB bandwidth (w_c) are determined as summarized in Table IV.

Table IV. Summary of the PLL second order loop parameters

Control Parameter	**Expression**	**Charge-pump PLL Expressions**
Natural Frequency	$w_n = \sqrt{K_{LOOP} w_z}$	$w_n = \sqrt{\frac{K_{VCO} I_{CP}}{2\pi N C_1}}$
Damping Factor	$\xi = \frac{1}{2}\sqrt{\frac{K_{LOOP}}{w_z}}$	$\xi = \frac{R}{2}\sqrt{\frac{K_{VCO} I_{CP} C_1}{2\pi N}}$
Loop Bandwidth	$w_c \approx GBW = K_{LOOP}$	$w_c = \frac{K_{VCO} I_{CP} R}{2\pi N}$

Table V. Useful relations between second order approximation loop parameters

Parameters	**Relations**
Loop Bandwidth	$w_c \approx 2\xi w_n$
Natural Frequency	$w_n = 2\xi w_z$
Loop Zero	$w_z = w_c w_n^2$

Some useful relations between the loop parameters are given in Table V. Note that the approximation of w_c comes from the fact that the closed loop 3-dB bandwidth of a feedback system is approximately equal to the gain bandwidth product(GBW)

of its open loop gain [18].

Based on the ZigBee specifications given in Table II, the loop equations in Table IV and technology-dependent factors (current gain, control voltage dynamic range, varactor gain, etc.) the building block design parameters are determined and are listed in Table VI. The details of the derivation of the parameters in Table VI can be found in [15].

Appendix A provides a detailed design procedure and loop design analysis for charge-pump based PLLs and provides an alternative loop design for a ZigBee synthesizer as a design example.

Table VI. Summary of loop parameters used in the fabricated ZigBee frequency synthesizer prototype

Loop Parameter	**Value**
w_c	$2\pi\times$ 30kHz
w_z	$2\pi\times$ 7.5kHz
ξ	≈ 1
w_p	$2\pi\times$ 120kHz
K_{VCO}, I_{CP}	$2\pi\ \times$ 135MHz/V , 20 μA
R, C_1 , C_2	61 kohms, 346 pF, 21.62pF

The synthesizer consists of three separate voltage supply domains. The phase-frequency detector (PFD) and charge pump (CP) both use thick-oxide transistors and have a 3V supply instead of the nominal 1.8V of the 0.18 μm technology, to allow for cascode transistors in the charge pump and to improve matching. This configuration also increases the dynamic range of the control voltage and allows for a low VCO gain

$(2\pi \times 135MHz/V)$ to achieve the desired frequency range. The loop filter (LF) is a fully integrated solution that features an active capacitance multiplier [25].

It is common practice to separate the digital circuit supply from the analog power supply to minimize noise coupling from the notoriously noisy digital to the sensitive analog. Therefore, the digital frequency dividers (programmable divider and the inverter chain buffer that drives it) operate under a separate supply voltage. Since the digital circuit power consumption is directly related to its supply level, we operate these digital circuitry at a lower supply of 1.3V. The LC-tank VCO, the divide-by-2 circuit that follows it and the differential-to-single-ended (2to1) buffer that drives the digital divider circuitry all operate at the nominal 1.8V supply.

The VCO operates at twice the channel frequency range (4.81GHz - 4.96GHz) and features frequency tuning through the use of PMOS inversion varactors and junction varactors for discrete coarse and continuous fine tuning, respectively. The PFD, CP, LF and VCO are designed by Mr. Rangakrishnan Srinivasan and the details of their design are provided in [15]. In this dissertation, we focus on the implementation details of the frequency dividers.

D. Frequency Dividers

1. Divider Basics

a. Divide-by-2 Operation

As seen in Fig. 2, a divide-by-2 prescaler circuit generates quadrature LO signals to be used by up/down conversion mixers of a transceiver. Note that since the VCO operates at double the channel frequencies, this divider circuit should operate at 5GHz range and is therefore critical in terms of power consumption and performance.

Fig. 4 shows the block diagram of a simple divide-by-2 circuit. Note that it con-

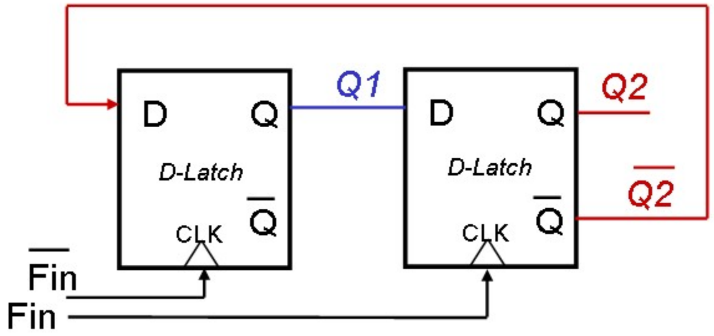
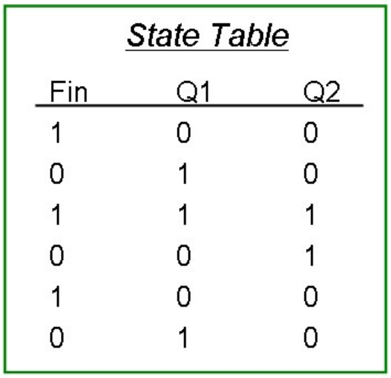


Fig. 4. Block diagram of a divide-by-2 frequency divider

sists of a D-flip-flop (two D-latches in master-slave configuration), placed in a negative unity feedback. To understand how this circuit divides its clock input's frequency, we should examine its state table and timing diagram, given in Fig. 5 (a) and (b), respectively.

In the state table, each row represents the next state of the output that occurs after the previous state. Note that the outputs Q_2 and Q_1 have a period, twice that of the clock signal and these outputs have 90 degrees of phase difference. This shows that a divide-by-2 circuit that consists of two master-slave latches, inherently generates quadrature phases at its two latch outputs.

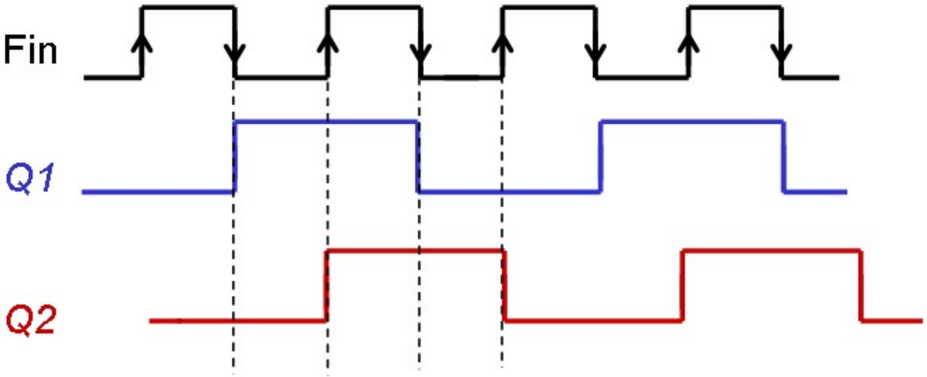
The circuit implementation of the divider depends on several design metrics such as operating frequency and clock input signal swing. Several circuit techniques will be discussed in detail, in Chapter III Section A. In the proposed ZigBee synthesizer, the divide-by-2 circuit is implemented with Current Mode Logic [2,3,26,27] due to its ability to operate at very high frequencies and for quality quadrature signal generation with very small IQ mismatch and with smaller controlled swing at the LO to improve mixer linearity.



State Table

Fin	Q1	Q2
1	0	0
0	1	0
1	1	1
0	0	1
1	0	0
0	1	0

(a)



(b)

Fig. 5. Divide-by-2 operation (a) state diagram (b) input and output timing

b. Division By an Odd Ratio

The most basic frequency division, divide-by-2 operation, was discussed and demonstrated in Fig. 4 and Fig. 5. Similarly, frequency division where the division ratio is a power of 2, can be implemented by cascading asynchronous divide-by-2 stages. However, division by an odd number is not as straightforward.

One of the most commonly used odd number dividers is a divide-by-3 circuit [14].

The block diagram of a divide-by-3 circuit is shown in Fig. 6. The operation of a divide-by-3, and most odd-ratio divisions, are based on power-of-2 ratio divisions and additional logic controls that prevent certain output states and therefore limit the total number of states, and therefore the period, of the output signals.

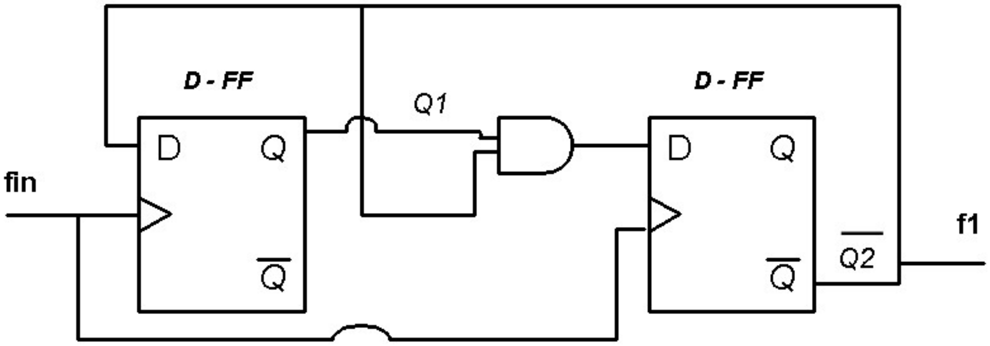


Fig. 6. Block diagram of a divide-by-3 frequency divider

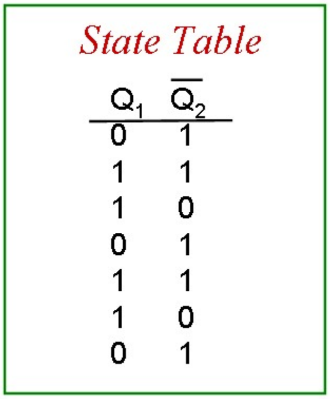
The divide-by-3 circuit example of Fig. 6 consists of two D-flip-flops. Note that in Fig. 4, we demonstrated the most simple division through a single flip-flop which consists of two latches. However, often, the latches are not shown for simplicity and only the D-flop-flops are shown in block diagrams. Therefore, in the following divider block diagrams, we will only show the flip-flops, since the internal master-slave latches are implied by the definition of a flip-flop.

The additional AND gate in the divide-by-3 circuit results in the below relationship between the two outputs $Q1$ and $\overline{Q2}$ and prevents the output state 00.

$$Q1(n) = \overline{Q2}(n-1) \tag{2.12}$$

$$\overline{Q2}(n) = \overline{\overline{Q2}(n-1)\ \text{ AND }\ Q1(n-1)}$$
$$\overline{Q2}(n) = Q2(n-1)\ \text{ OR }\ \overline{Q1}(n-1) \tag{2.13}$$

The resulting state table for the outputs is shown in Fig. 7. Note that the states of *Fin* are not shown for simplicity, but each state of $Q1$ and $\overline{Q2}$ are triggered by a transition of *Fin*. Therefore, each row (each state of $Q1$ and $\overline{Q2}$) implies one clock period of*Fin*. It is seen that the outputs have three possible states, therefore three times the period of the input clock *Fin*.



State Table

Q_1	$\overline{Q}_2$
0	1
1	1
1	0
0	1
1	1
1	0
0	1

Fig. 7. State table of a divide-by-3 frequency divider

c. Dual Modulus Division

Dual modulus division, often noted as divide-by-M/(M+1) is very commonly used in frequency synthesizers. A commonly used dual modulus divider, that implements the core of larger ratio dual modulus dividers, is the divide-by-3/4 circuit that is shown in Fig. 8. When the modulus control (MC) is low, the output of the OR gate becomes $\overline{Q2}$ and therefore the circuit reduces to a divide-by-3. When MC is high, the input of the second flip-flop is equal to the output of the first flip-flop. Then, the divider acts as the synchronous cascade of two divide-by-2 circuits, therefore becomes a divide-by-4 circuit.

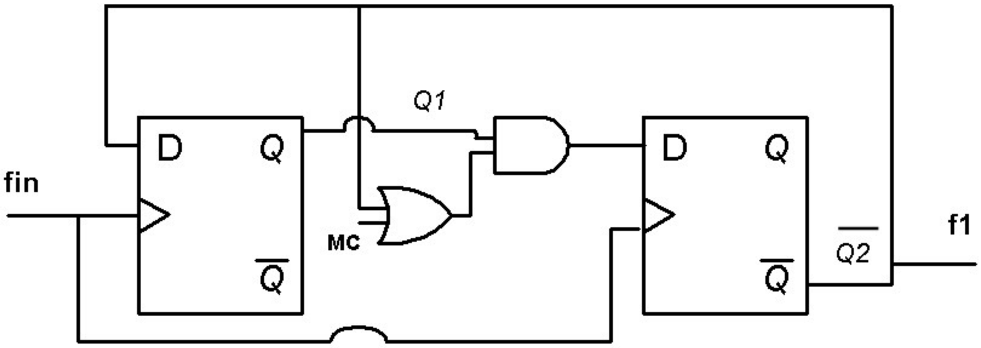


Fig. 8. Block diagram of a dual modulus divide-by-3/4 frequency divider

Due to the reduced number of inverter stages, NAND and NOR circuits are preferred over AND, OR circuits in implementation. To convert the divide-by-3/4 block of Fig. 8 into a NOR-based implementation, we apply De Morgan's law as follows:

$$\begin{aligned}
Q2(n) &= \Big(MC(n-1) \text{ OR } \overline{Q2}(n-1)\Big) \text{ AND } Q1(n-1) \\
\overline{Q2(n)} &= \overline{\Big(MC(n-1) \text{ OR } \overline{Q2}(n-1)\Big)} \text{ OR } \overline{Q1}(n-1) \\
Q_2(n) &= \Big(MC(n-1) \text{ NOR } \overline{Q2}\Big) \text{ NOR } \overline{Q1}(n-1) \qquad (2.14)
\end{aligned}$$

the NOR-based implementation of the divide-by-3/4 circuit is shown in Fig. 9. Another core dual modulus divider is a divide-by-2/3 circuit which follows a similar logic with a 3/4 divider. The derivation of a divide-by-2/3 circuit is left to the reader.

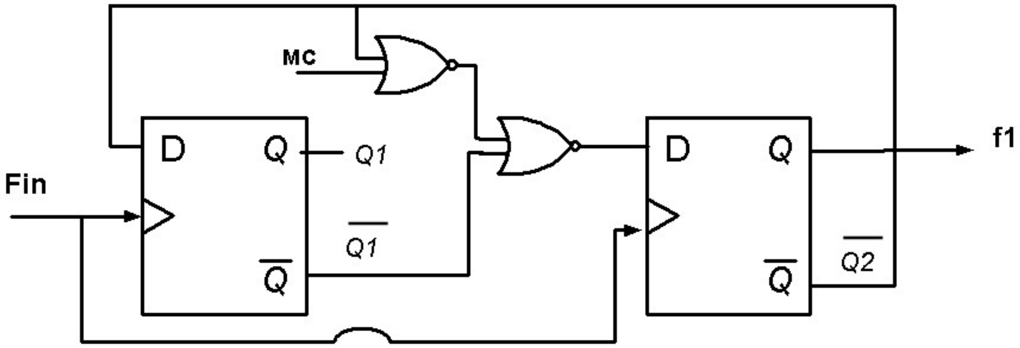


Fig. 9. Block diagram of the NOR based divide-by-3/4 circuit

2. Programmable Divider

The programmable dividers in the feedback path of the loop should generate the division ratios given by (4.1). Pulse-swallow dividers [14] are commonly used in wireless frequency synthesizers to control the output channel frequency of the PLL. The block diagram of a pulse-swallow divider is shown in Fig. 10.

The input clock of the pulse-swallow divider (in this design, it is the output of the divide-by-2 IQ generation circuit in the forward path) is a dual modulus prescaler (DMP). The DMP runs at the highest frequency in the pulse-swallow divider, and is therefore the most power-critical block. Depending on the value of its control signal MC_IN, the DMP divides its input frequency by M (when MC_IN is logic 0)or by $M+1$ (when MC_IN is logic 1).

The output of the DMP controls the program and swallow counters. The program counter can count to a maximum of P cycles where the value of P is constant. The value of S on the other hand, is variable and determined by the channel selection control bits. The operation is as follows. Let's assume that initially the DMP control

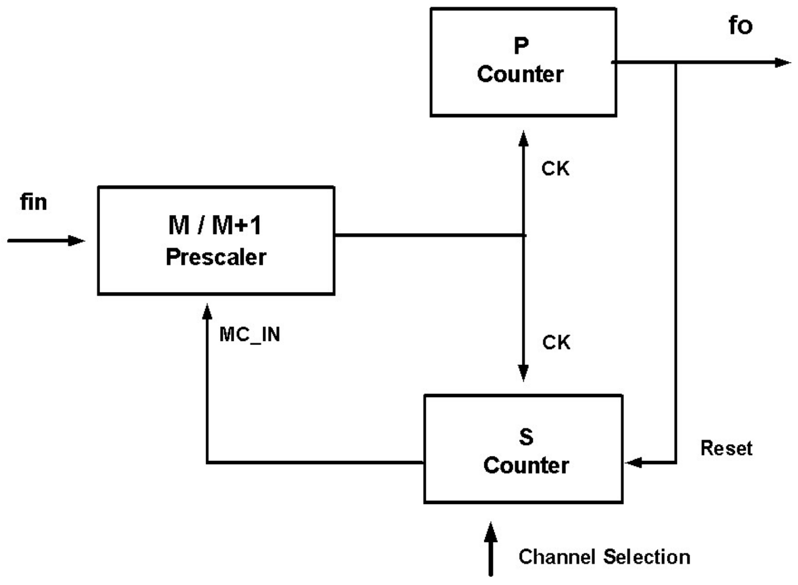


Fig. 10. Block diagram of a pulse-swallow programmable divider

MC_IN is set high and DMP is in divide-by-M mode. In this case, program counter counts and when it reaches S cycles the S counter resets MC_IN. The DMP starts dividing-by-(M+1). The P counter continues counting until it reaches its maximum count of P. Then, the S counter sets MC_IN to a logic high again and a new division cycle begins. Note that the output Fo of the pulse-swallow divider goes through one period for every N cycles of the input Fin. Based on this discussion, the total division ratio of the pulse-swallow divider is given by:

$$N = (M+1) \times S + M \times (P - S)$$

$$N = M \times P + S \tag{2.15}$$

Table VII summarizes the values of M,P and S used in this implementation, to achieve the values of N given in (4.1). Note that P is a power of 2. Therefore, the counter will wrap around and start counting from 0 automatically when it reaches its maximum count. This simplifies the circuitry since a non-power-of-2 value of P would require additional count stop and reset circuitry.

Table VII. Summary of the pulse-swallow divider parameter values

Parameter	*Value*
P	32
S	1,2,3,...,16
M/(M+1)	15/16

As seen from Table VII, S takes 16 values. Therefore, 4 channel control bits are used in the design. While it is called a counter, in implementation, the function of S described above, can be implemented with digital logic circuitry. The implementation of program counter and the function of S in this synthesizer is shown in Fig. 11. CK is the output of the dual modulus prescaler as shown in Fig. 10. Note that the Set and Reset control signals have one more clock delay due to the additional D-flip-flops. Therefore, these D-flip-flop inputs are high when P counter output is equal to P-1 and to S-1. Then, the channel select bit word is:

$$Ch_3Ch_2Ch_1Ch_0 = S - 1 \tag{2.16}$$

Based on the values of M shown in Table VII, the P and S counters operate at frequencies less than 200MHz. Therefore, the circuit-level implementation of the logic gates shown in Fig. 11 are done by conventional static CMOS logic [28].

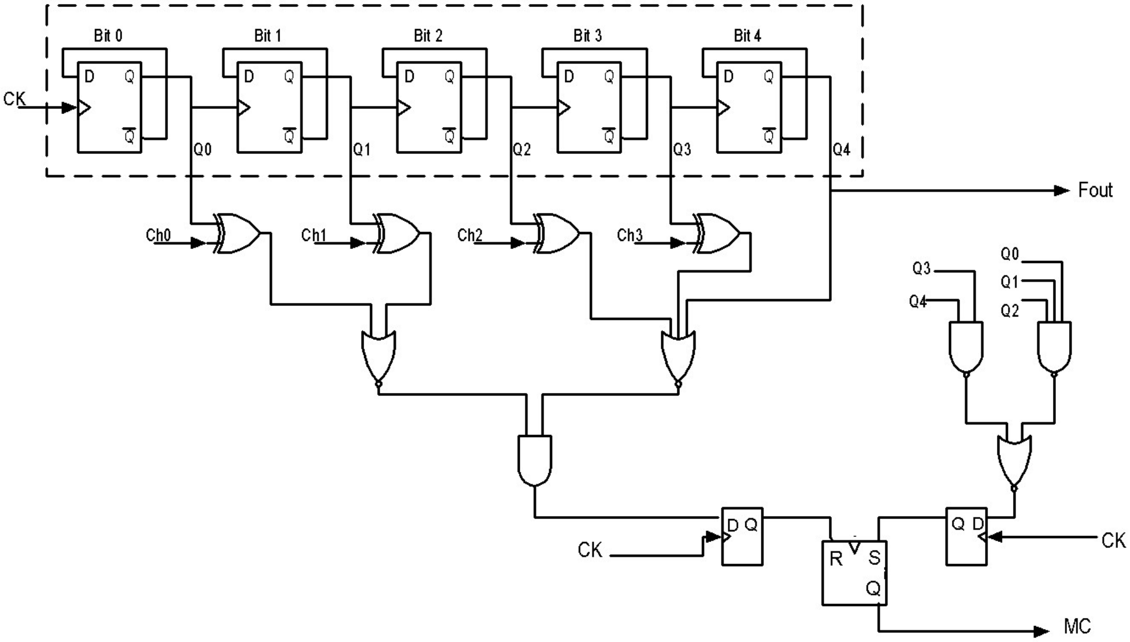


Fig. 11. Block diagram of the P counter and S block

3. Dual Modulus Prescaler Implementation

The 15/16 dual modulus prescaler (DMP) is implemented with a divide-by-3/4 core which is followed by asynchronous divide-by-2 stages. Fig. 12 displays the block diagram of the 15/16 prescaler where FIN is the input clock signal to be divided in frequency, $FOUT$ is the output clock signal and MC_IN is the input modulus control that is generated by the P and S counters.

The prescaler divides FIN by 15 when MC_IN is low and by 16 otherwise. The prescaler consists of a divide-by-3/4 core marked with a circle in the figure, as well as asynchronous divide-by-2 stages. Note that the physical connections between the divide-by-3/4 stage and the following /2 stages in the prescaler are not drawn for simplicity but are marked with signal names such that the output of 3/4 stage, $F1$,

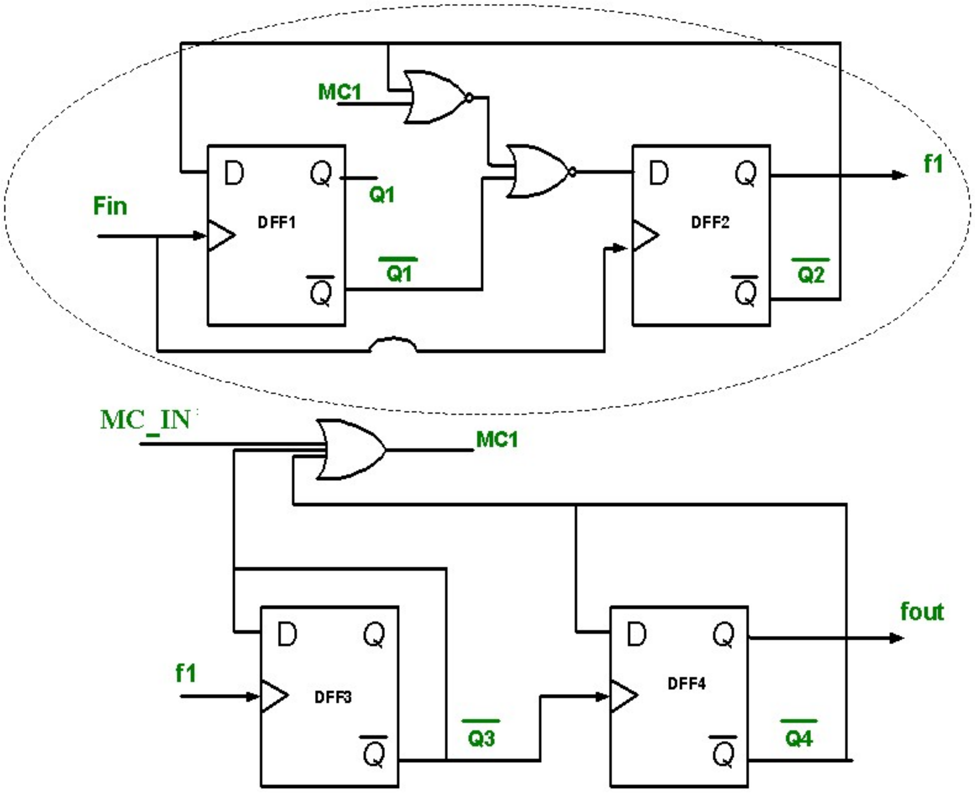
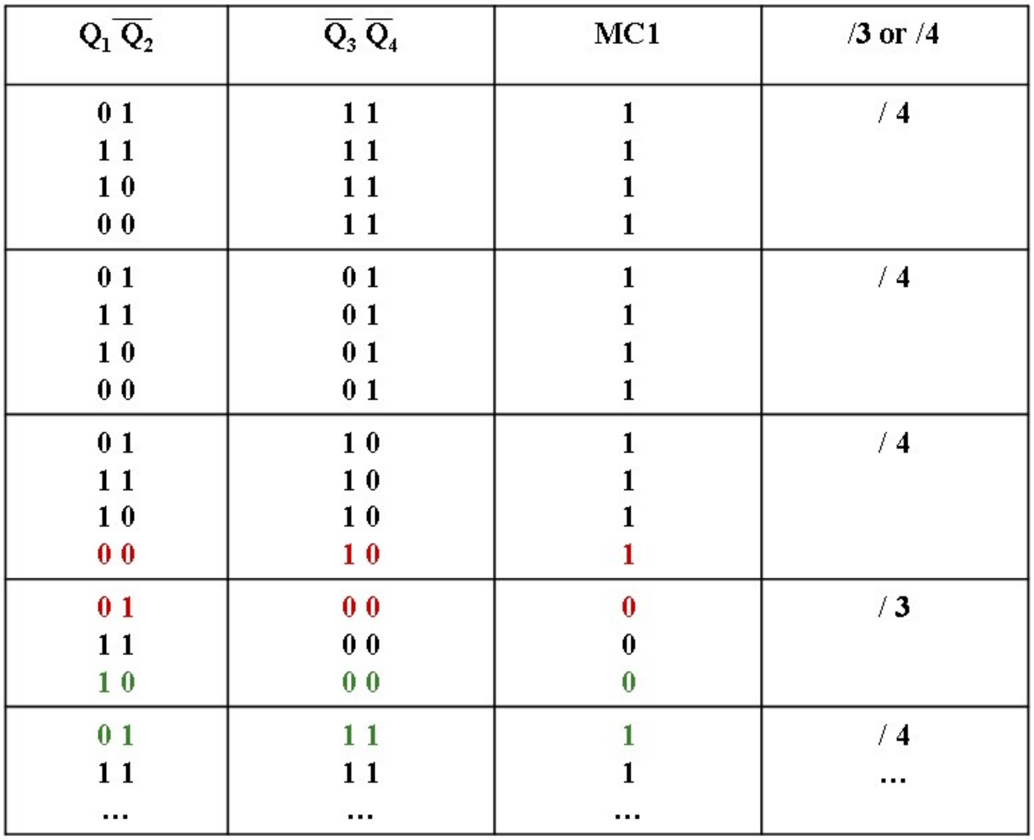


Fig. 12. Dual modulus (15/16) prescaler block diagram

is the clock input of the third flip-flop stage and the output of the OR stage, $MC1$, acts as the modulus control of the 3/4 stage. The state table that demonstrates the /15 operation is shown in Fig. 13.

Note that the division-by-15 is performed by swallowing one of the 16 possible output states. In this case the swallowed state is 0000. In this implementation, the first two flip-flops, DFF1 and DFF2, are rising-edge triggered while the last two, DFF3 and DFF4, are falling-edge triggered. The reason behind preferring falling edge triggered flip-flops for the last two stages is as follows. In the state table Fig. 13, it is seen that when /3 mode begins (marked red), Q_1 and $\overline{Q_2}$ are three states away from the swallowed 00 state. This gives enough time to the feedback control to settle. If DFF3



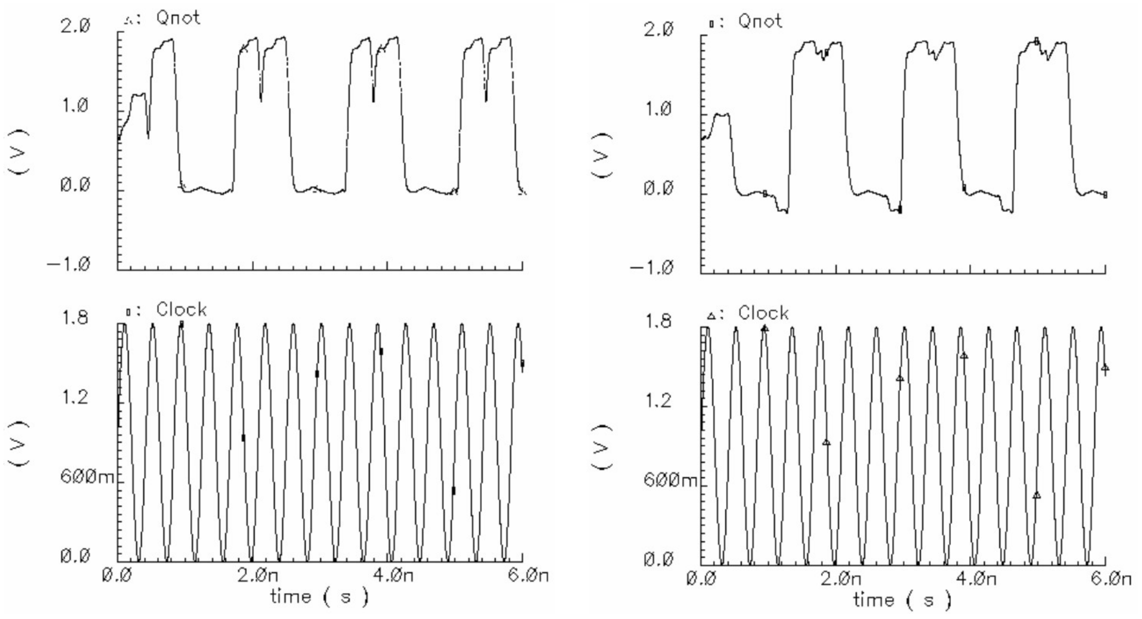
$Q_1 \overline{Q_2}$	$\overline{Q_3} Q_4$	MC1	/3 or /4
0 1 1 1 1 0 0 0	1 1 1 1 1 1 1 1	1 1 1 1	/ 4
0 1 1 1 1 0 0 0	0 1 0 1 0 1 0 1	1 1 1 1	/ 4
0 1 1 1 1 0 0 0	1 0 1 0 1 0 1 0	1 1 1 1	/ 4
0 1 1 1 1 0	0 0 0 0 0 0	0 0 0	/ 3
0 1 1 1 ...	1 1 1 1 ...	1 1 ...	/ 4 ...

Fig. 13. State table of the 15/16 prescaler

and DFF4 were rising edge triggered, when the /3 mode starts, Q_1 and $\overline{Q_2}$ would have to swallow the next state immediately, which would tighten the feedback timing requirement significantly [14]. The states that signal the return to the /4 mode are marked with green.

The circuit-level implementation of the flip-flops and logic gates of the DMP are done with dynamic True Single Phase Clocking (TSPC) [28], [29]. This circuit technique is preferred over CML that was used in the initial /2 prescaler because of its lower power consumption. The details of frequency divider circuit techniques and their trade offs are discussed in Chapter III Section A.

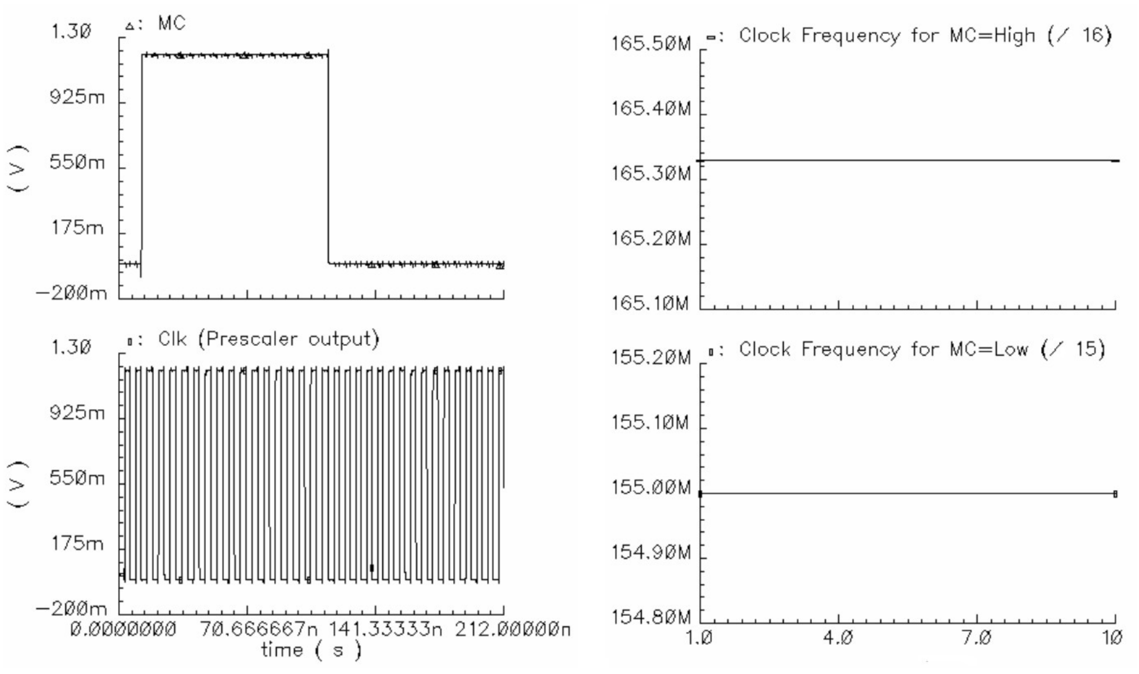
TSPC circuits suffer from glitch problems particularly at high frequency operation. Therefore, the divide-by3/4 core of the prescaler shown in Fig. 12 is implemented with a glitch-free TSPC technique proposed in [30]. Note that DFF3 and DFF4 operate at lower frequency in the prescaler, and are therefore implemented with regular TSPC flip-flops. Fig. 14 (a) show the 2.4GHz input clock and output waveforms of a divide-by-3/4 circuit implemented with TSPC logic, in 0.18 μm technology while Fig. 14 (b) demonstrates the waveforms of the same circuit implemented with glitch-free circuit.



(a) (b)

Fig. 14. Circuit-level simulations of glitch in divide-by-3/4 circuit at 2.4GHz operation (a) using regular TSPC logic (b) using glitch-free TSPC logic

Post-layout simulations of the 15/16 prescaler are shown in Fig. 15 for channel 16 (2.48GHz operation). The modulus control signal generated by the P and S counters to control the division ratio of the 15/16 prescaler is shown at the top while Prescaler output signal which forms the clock signal for the P and S counters, is shown at the bottom of Fig. 15 (a). Note that the prescaler will divide its input frequency (2.48 GHz) by 16 when its control is set high, the frequency measurement of the prescaler output for the duration when modulus control is high, is given at the top of Fig. 15 (b). When the modulus control is reset to low, division by 15 is performed. The frequency measurement of the prescaler output for the duration when modulus control is low, is given at the bottom of Fig. 15 (b).



(*a*) (*b*)

Fig. 15. Post-layout simulations of 15/16 prescaler circuit at 2.48GHz operation. (a) Prescaler modulus control signal generated by the P and S counters (top) and prescaler output signal (bottom). (b) Frequency of the prescaler output signal for modulus control high (top) and low (bottom).

4. Frequency Divider Buffers

The frequency synthesizer employs a divide-by-2 circuit implemented with CML circuitry, as shown in Fig. 2. Often, the programmable divider, which also runs at high frequency (2.4GHz band channel frequencies in this design), requires its own buffer since it provides clock input capacitance that is significant at the high operating frequency.

In this design, the CML divider is a differential circuit, to provide differential quadrature LO signal to the up and down conversion mixers. However, the first block of the programmable divider, the 15/16 prescaler is implemented with TSPC circuitry which is single ended. The single-ended nature of the TSPC circuit minimizes routing of the clock that runs at critical speeds and diminishes the effect of crosstalk and interconnect capacitance. However, the TSPC circuitry requires a differential-to-single-ended (2to1) conversion buffer between the CML divider and the RF prescaler.

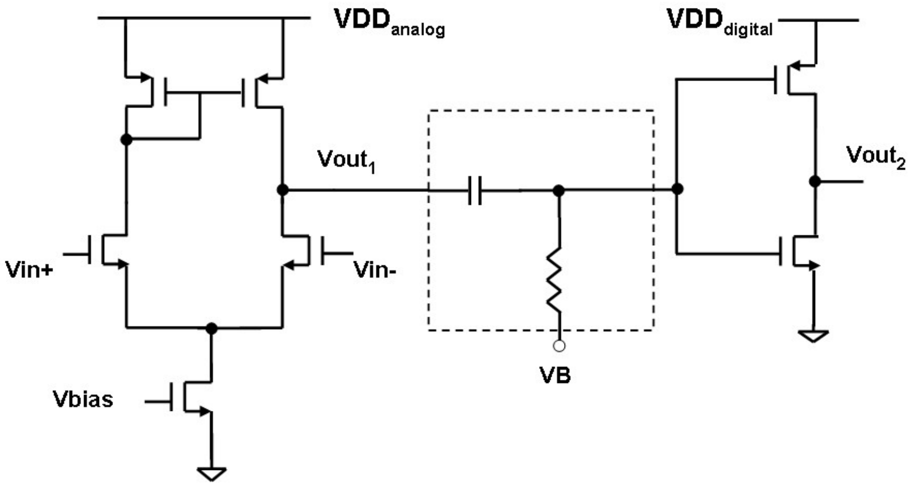


Fig. 16. Schematic of the differential to single ended buffer, the bias-T circuit to set proper common mode level and the first inverter of the inverter chain buffer

Note that at RF speeds, the buffer will deliver loss rather than gain in converting the differential input swing into a single-ended output swing. Also, as discussed in Section C, the synthesizer has separate voltage domains for its digital and analog circuitry. The digital supply domain is at a different voltage level (1.3V) than the analog supply domain (VCO, CML divider and the 2to1 buffer) which operates at the nominal supply level of this technology (1.8V).

To compensate for the loss of the 2to1 buffer and the small swing, and also to switch from the analog supply domain to the digital, a chain of four inverters are employed between the 2to1 buffer and the programmable dividers. These inverter buffers boost the signal swing and convert the signal levels to the digital domain supply levels.

Fig. 16 demonstrates the 2to1 buffer and the first inverter of the inverter chain. The supply level change between the two is performed by the insertion of a bias-T circuit. The value of VB, which sets the DC common mode level of the digital domain input signal, can be set to $VDD_{DIGITAL}/2$ which is a standard inverter switching threshold for digital inverters.

E. Measurement Results

The frequency synthesizer was implemented in TSMC 0.18 μm CMOS technology and fabricated. It was packaged in a 64-pin TQFP style packaging and mounted on an FR-4 PC board. Fig. 17 shows the die micrograph.

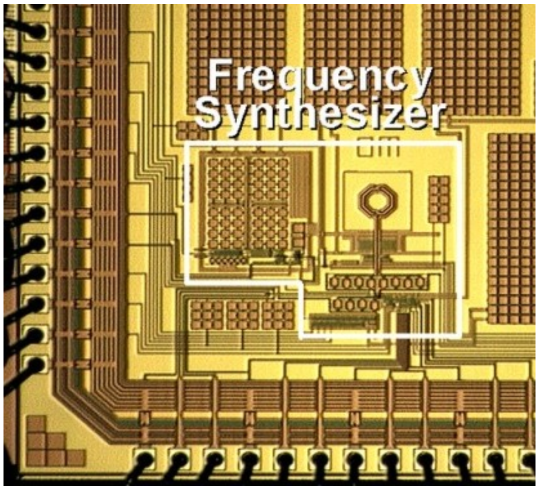


Fig. 17. Die micrograph of the frequency synthesizer

Fig. 18 and Fig. 19 show the synthesizer output frequency spectrum for the first and last channels of ZigBee, respectively. As discussed in Section A, the alternate channel (10MHz offset from the channel) spur rejection requirement was tougher than the adjacent channel (5MHz offset) rejection specification in ZigBee. It is seen that the worst case alternate channel spur suppression, observed at the last channel, is 50dB which comfortably meets the design specification.

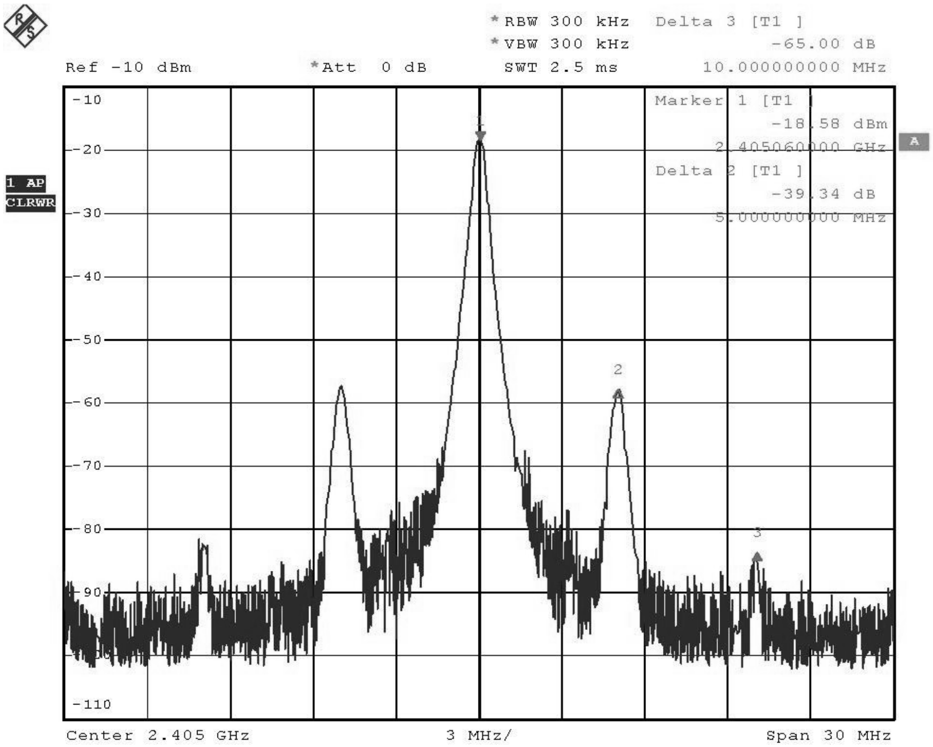


Fig. 18. Measured output spectrum of the synthesizer demonstrating first channel of ZigBee

Fig. 20 demonstrates the phase noise spectrum of the synthesizer. The measurements showed that the frequency synthesizer met the specifications given in Table II with a power consumption of 15mW. Table VIII summarizes the measured performance of the synthesizer. The implemented synthesizer that was discussed in this chapter and the measurement results were partially published in [4] and [15].

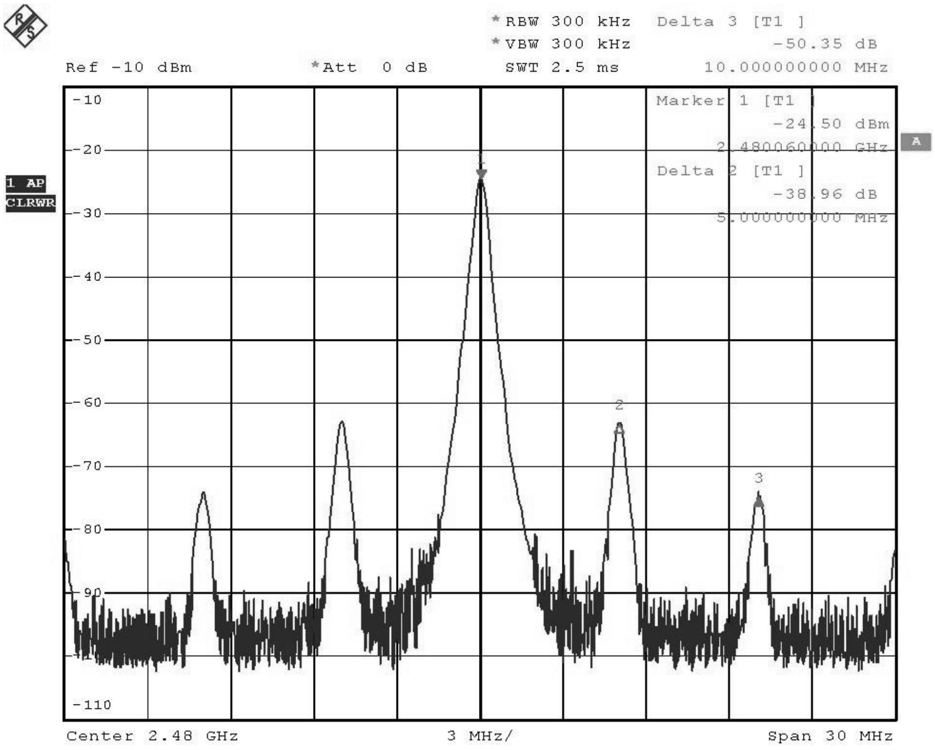


Fig. 19. Measured output spectrum of the synthesizer forchannel 16 of ZigBee [4]

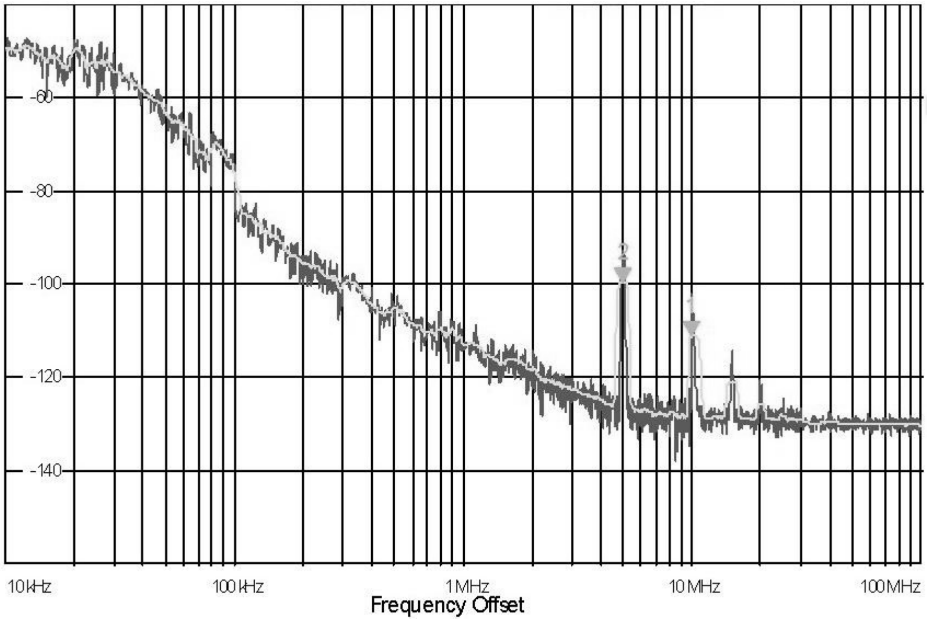


Fig. 20. Phase noise spectrum of the frequency synthesizer [4]

Table VIII. Measured performance of the ZigBee frequency synthesizer [4]

Performance Metric	**Measured Value**
Frequency Synthesis	2.405GHz - 2.48 GHz
Reference Frequency	5MHz
Number of Channels	16
Settling Time	55 μs
Spur Suppression	-40dBc at 5MHz -50dBc at 10MHz
Phase Noise	< -130dBc/Hz at 10MHz offset < -122dBc/Hz at 3.5MHz offset
Power Consumption	15 mW
Area	0.63 mm^2
Technology	0.18 μm CMOS

1. Discussion on Power Consumption

The measured total synthesizer power consumption is 15mW. In measurements, we can obtain the power consumption of the individual supply domains to understand the power distribution within the synthesizer. Fig. 21 shows this distribution. Note that the power supply of the CML /2 circuit and the 2to1 buffer were separate from the VCO in measurements although both were 1.8V. This was done to enable the characterization of the individual blocks.

The CML /2 circuit and the 2to1 buffer have their individual bias currents and therefore is easy to determine the individual power consumption from the values of the bias current setup during measurements. The digital circuits of the inverter

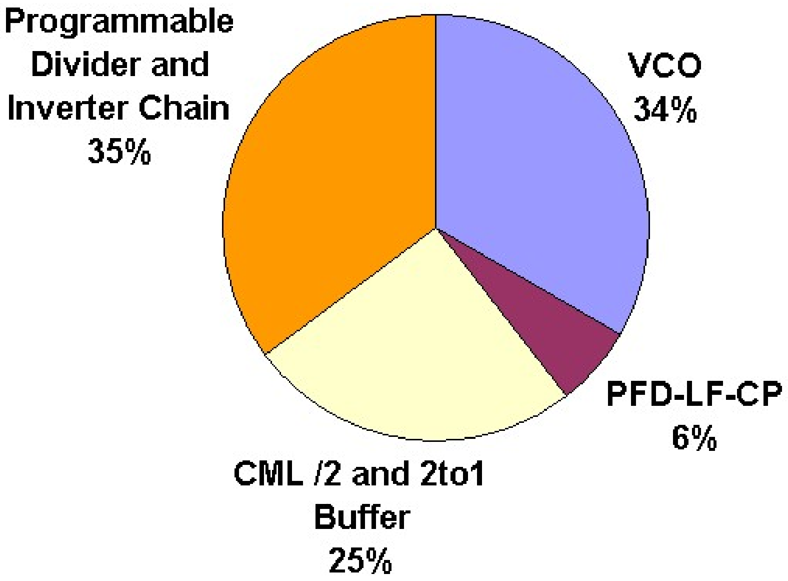


Fig. 21. Pie chart of the measured power consumption distribution in the ZigBee synthesizer

chain and the programmable dividers on the other hand, consume dynamic switching power. In measurements, it was noted that both the speed and the power consumption of the circuits were as expected from the post-layout simulation characterizations. Therefore, we can extrapolate the individual power consumption of the digital circuits based on the relative distribution of power from simulations. Fig. 22 demonstrates the power consumption pie chart that details the distribution of the total measured power consumption to individual building blocks based on simulation data.

Fig. 22 shows that the VCO that runs at double the channel frequency (4.96GHz band) consumes 34% of the total synthesizer power while the CML /2 circuit which operates at the same frequency as the VCO takes 12%. Note that 47 % of the total power is consumed in the programmable divider and its buffers in the feedback path of the synthesizer.

Since the programmable dividers consume a significant portion of the total power,

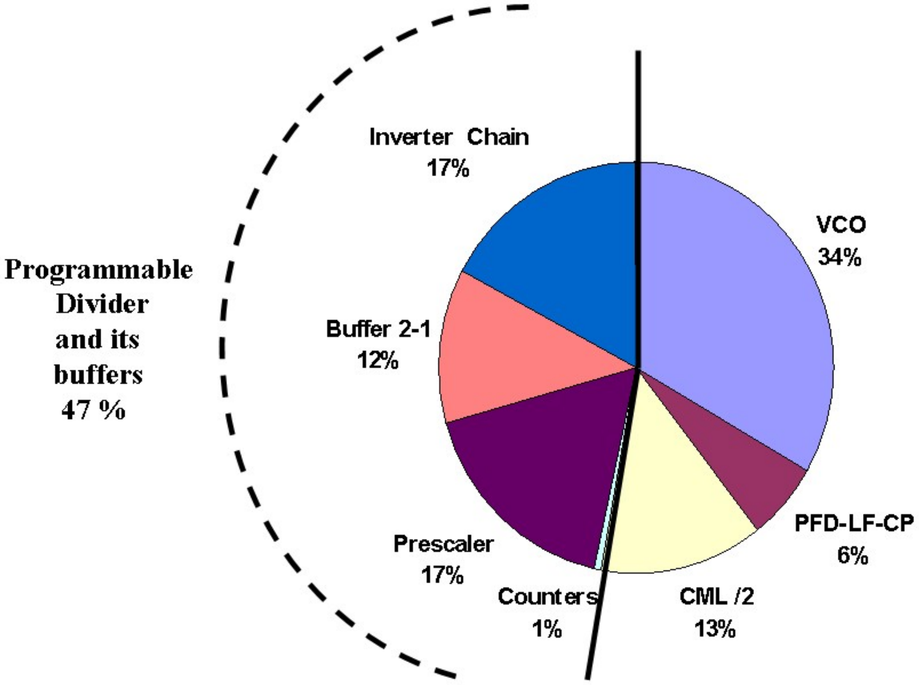


Fig. 22. Pie chart of the power consumption distribution in the ZigBee synthesizer with individual frequency divider blocks

a power reduction in these circuits will not only reduce the synthesizer power, but will also significantly affect the power consumption of the transceiver system that employs this synthesizer since the synthesizer contributes to both transmit and receive mode powers. With 15mW power consumption, the synthesizer takes 66% of the total ZigBee receiver power and 61% of the total transmitter power(based on the post-layout simulation results of the other transceiver building blocks designed by the ZigBee team members [16]).

It is seen in Fig. 22 that the RF buffers that drive the prescaler consume more power in total then the prescaler itself. Then, to reduce the power consumption of the frequency synthesizer, we should implement a low-power 15/16 prescaler whose input clock capacitance is small and therefore easier to drive at high frequency.

CHAPTER III

FREQUENCY DIVIDER CIRCUITS AND A NEW DCVSL-R DELAY CELL

In Chapter II, it was shown that the power consumption of the RF frequency dividers are a significant contributor on the power consumption of a frequency synthesizer. It was also concluded that not only the power consumption of the frequency dividers should be minimized by investigating low power circuit techniques, but their input clock capacitance, which effects the power consumption of the buffer that drives the dividers at high frequency, should also be small.

In this chapter, circuit techniques to implement the high frequency dividers of a frequency synthesizer will be discussed, Differential Cascode Voltage Switch Logic (DCVSL) circuits will be explored as a candidate to implement the RF dividers, a delay model to characterize the speed performance of DCVSL circuits will be proposed and a new delay cell called DCVSL-R, that has a better speed and power consumption performance, will be presented.

A. Frequency Divider Circuit Techniques

A commonly used circuit technique in the high frequency dividers of wireless radio synthesizers is CML [2], [3]. A CML latch is shown in Fig. 23. CML circuits enable high-speed operation with small signal swing. Their constant DC bias current minimizes switching noise, and their differential nature makes them immune to common-mode noise. However, CML, though high speed, consumes considerable power due to it's DC bias current and has limited headroom due to stacked transistors. Load resistance and bias current values determine the output swing and DC common mode level, putting a lower limit on the bias current value. Moreover, a CML D-flip-flop requires two CML latches of Fig. 23, using fourteen transistors and four resistors for a single flip-flop, resulting in much more area than traditional flops.

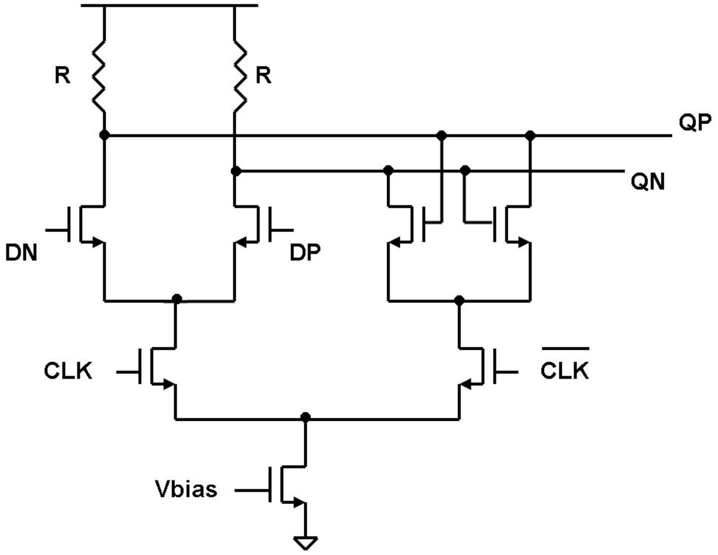


Fig. 23. Schematic of a CML latch

As an alternative to CML, TSPC circuits implement the frequency dividers of wireless-radio frequency synthesizers [4, 5, 31]. Fig. 24 shows a rising-edge triggered TSPC D-flip-flop. They consume no static power and use fewer transistors. However, they have stacked transistors that present large bias-dependent capacitive loading. Due to these large internal parasitics and the hard-switching nature of the transistors, they have high switching current peaks, leading to noise.

In a PLL, frequency dividers are driven either by a buffer or directly by the VCO, and VCO architectures are often differential. Single-ended frequency dividers such as TSPC, result in an asymmetrical loading at the VCO output, which leads to mismatch at the LO signals of a transceiver. To minimize the mismatch, dummy circuits can be used to provide symmetric loading for the VCO [5]. Such dummy circuits will

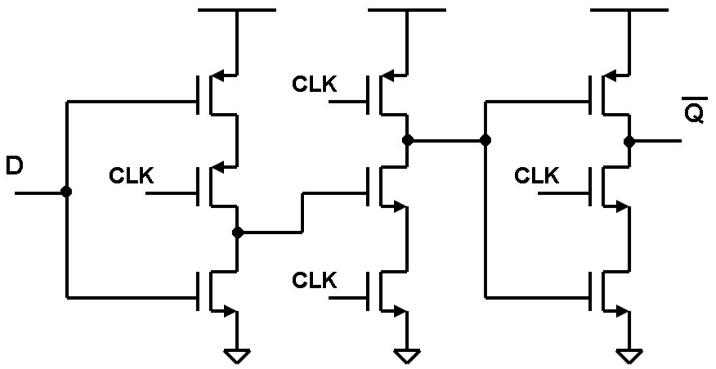


Fig. 24. Schematic of a TSPC D-flip-flop

not only generate additional parasitics at the RF nodes but also if left disconnected from VDD to save power, they will not completely remove mismatch. Differential-to-single-ended conversion buffers may also be employed; however, at high frequency these buffers consume large power. [4] uses such a buffer followed by an inverter chain and while the TSPC significantly reduces the dual-modulus-prescaler (DMP) power, the buffers consume as much power as the DMP. [5] uses a modified version called E-TSPC to avoid stacked transistors. This reduces buffer power but E-TSPC has charge sharing issues and static power dissipation.

Based on the above discussion, we can conclude that the optimum divider topology should have low power consumption, provide a symmetric (differential) loading for the VCO, have small clock input capacitance and should be able to operate at the channel frequency with low switching noise. We next discuss how DCVSL implementation solves these problems.

B. Differential Cascode Voltage Switch Logic Circuits

The differential cascode voltage-switch-logic (DCVSL) family, first introduced in 1984, has small input gate capacitance (compared to full CMOS logic styles) and can implement complex logic functions with low transistor count [32]. A simple DCVSL inverter is shown in Fig. 25.

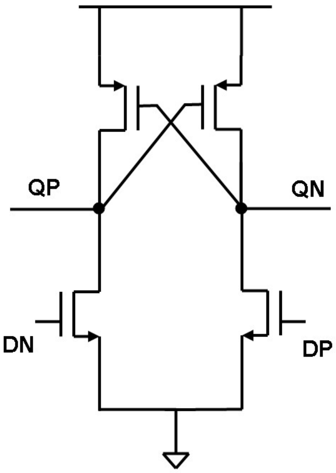


Fig. 25. Schematic of a DCVSL inverter

One drawback of this circuit technique occurs while the PMOS load transistors are in latching mode. For a brief period, both PMOS and NMOS transistors in at least one of the differential branches are on at the same time, leading to crowbar current for a short time. However, this transition period also smoothens the instantaneous current switching of these logic gates and generates less switching supply noise compared to hard-switching, static, full-CMOS logic.

Several static and dynamic versions of DCVSL have been proposed in subsequent years such as a differential split level (DSL) scheme where the speed is enhanced by limiting the output swing to half the supply voltage [33] with a trade-off of increased complexity and the need for the generation of an additional reference voltage. Most of the proposed DCVSL variations are based on modifications to the PMOS load, which is a regenerative latch in the initially proposed static version.

A dynamic precharged version where the regenerative PMOS loading is replaced by precharge transistors and inverters was also proposed in [32]. Another dynamic scheme that keeps the cross-coupled PMOS loads without the inverters was shown in [34]. Several additional modified DCVSL family versions are proposed and the existing structures are compared in [35] and [36]. The majority of the literature in this area focuses on the implementation of complex digital logic functions using the DCVSL family, without an emphasis on high speed. Therefore, all the above mentioned sources present modifications that involve the addition of several transistors to the DCVSL structure, increasing the overall complexity.

Several DCVSL based flip-flops are discussed and compared in [29] with an emphasis on speed improvement, which is of crucial interest in the frequency divider application. This reference provides a very good comparison between the different DCVSL latch schemes. The conclusion is that for high speed latches, a simple non-precharge dynamic latch proves to be the most efficient. The D-flip-flop (DFF) of Fig. 26 shows the best candidate for high speed applications due to its simplicity and low transistor count. By avoiding precharge schemes, additional PMOS clock transistors are eliminated.

Among the various circuit families discussed in this and the previous section, we found that the non-precharge, two-phase-clocked DCVSL D-flip-flop of Fig. 26 is best suited for the frequency dividers of a synthesizer. Due to its small number of transistors, this flip-flop is fast. The whole flip-flop has only two clock transistors and no stacking, resulting in a very small clock input capacitance. Such small capacitance is crucial to minimize the clock driver buffers' power consumption. The flop has a crowbar current drawn during input transitions, yet the average power consumption is still much less than that of CML circuits. DCVSL circuits have lower power-supply glitches, as their switching capacitance is lower than that of TSPC. The

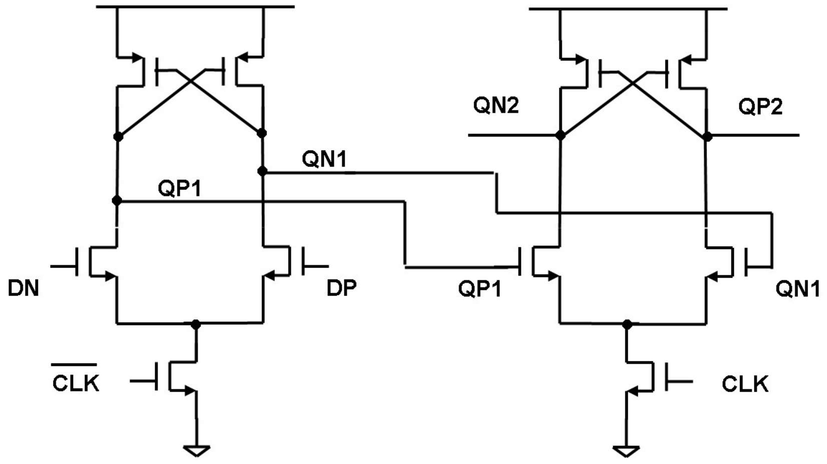


Fig. 26. Two-clock-phase DCVSL flip-flop

pseudo-differential clocking of the DCVSL flip-flop in Fig. 26 offers symmetric loading for the VCO, preventing mismatch problems at PLL outputs. However, the DCVSL structure has an inherent delay bottleneck that limits its operation speed and results in asymmetrical outputs, as will be discussed in the next section.

C. A Delay Model for DCVSL Circuits

1. Analysis and Derivation

Digital circuits' speed is characterized by their propagation delays, i.e. the low-to-high switching propagation delay τ_{PLH} (the delay from the input falling from logic high to low to the output rising from logic low to high) and the high-to-low switching propagation delay τ_{PHL} [28]. To understand the transient behavior of DCVSL circuits, we analyze the propagation delay of a simple DCVSL inverter. Fig. 27 shows the DCVSL inverter with a load capacitance C_L and with switching complementary inputs.

Delay behavior of standard CMOS inverters were analyzed in [37–39]. To develop a delay model for the DCVSL inverter, we revisit the simple yet intuitive Sakurai-

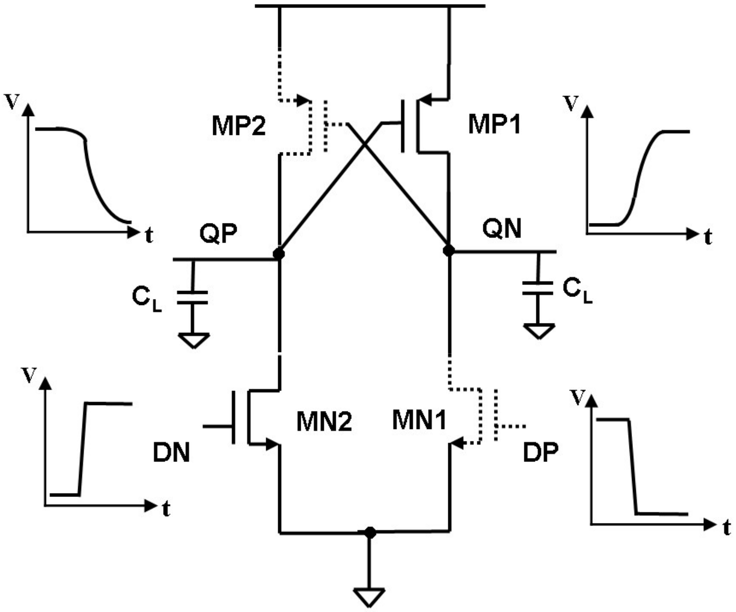


Fig. 27. DCVSL inverter setup for transient delay analysis

Newton delay model of [37] that was developed for a conventional static CMOS inverter. The transistor current-voltage equations of the alpha-power model of [37] are shown in (3.1), where α is a unitless technology-dependent parameter for a given transistor length and is derived from simulations as described in [37]. V_{DSO} and I_{DO} are the drain saturation voltage and drain current, respectively, of the transistor when $V_{GS} = V_{DS} = VDD$; and V_{TH} is the threshold voltage.

$$\begin{aligned} I_D &= I_{DO}\left(\frac{V_{GS}-V_{TH}}{VDD-V_{TH}}\right)^{\alpha}, \quad (V_{DS} \geq V'_{DS0}) \\ I_D &= V_{DS}\frac{I_{DO}}{V_{DSO}}\left(\frac{V_{GS}-V_{TH}}{VDD-V_{TH}}\right)^{\frac{\alpha}{2}}, \quad (V_{DS} < V'_{DS0}) \\ V'_{DSO} &= V_{DSO}\left(\frac{V_{GS}-V_{TH}}{VDD-V_{TH}}\right)^{\frac{\alpha}{2}} \end{aligned} \tag{3.1}$$

The motivation behind this analysis is to derive a closed-form model to understand the behavior of DCVSL circuits. Therefore, in our delay model derivation, we follow similar assumptions as [37] to simplify the delay equations. One such assumption is that the inverter input- and output-waveform slew-rates are similar. For the target applications of the DCVSL cells in this work (delay cells of ring oscillators and frequency dividers) we can safely assume that the DCVSL cells are driven by other DCVSL cells with similar delays.

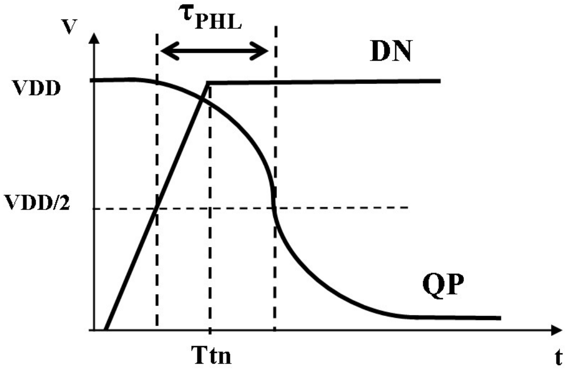


Fig. 28. Propagation delay derivation for τ_{PHL}

Fig. 28 shows the inverter input and output waveforms and the propagation delay, for the case of τ_{PHL}. The input waveform is approximated with a linear ramp where Ttn is the rising-input-waveform transition time (likewise, falling-input transition time will be referred to as Ttp for the case of τ_{PLH}). For the inverter under analysis, the NMOS driver transistor generates the rising input signal, and the PMOS load generates the falling input. Then, Ttn and Ttp can be approximated as [37]:

$$\begin{aligned} T_{tn} &= C_L \frac{VDD}{I_{DOP}} \left(\frac{0.9}{0.8} + \frac{V_{DSOP}}{0.8 \times VDD} \ln \left(\frac{10 \times VDD}{e \times VDD} \right) \right) \\ T_{tp} &= C_L \frac{VDD}{I_{DON}} \left(\frac{0.9}{0.8} + \frac{V_{DSON}}{0.8 \times VDD} \ln \left(\frac{10 \times VDD}{e \times VDD} \right) \right) \end{aligned} \tag{3.2}$$

where C_L is the load capacitance; I_{DOP}, V_{DSOP} and I_{DON}, V_{DSON} are the drain currents; and saturation voltages of the PMOS and NMOS transistors of the driving stage, respectively.

We also assume that the input waveform reaches its final value before the output reaches VDD/2, i.e. the point where propagation delay is measured. Moreover, to derive τ_{PHL}, (when DN is the rising input and QP is the falling output as shown in Fig. 28), we ignore the current conducted by MP2 before this transistor turns off completely. Therefore, we assume that QP is pulled down solely by MN2 (later, we will add a correction factor to the delay expression to compensate for this assumption). Then, the derivation of τ_{PHL} of a DCVSL inverter is similar to that of a standard CMOS inverter, and we can use the expression derived in [37]:

$$\tau_{PHL} \quad = \quad \tau_{05\_HL} - \frac{T_{tn}}{2} \tag{3.3}$$

where

$$\tau_{05\_HL} \quad = \quad T_{tn}\left(\frac{v_{TN} + \alpha_N}{1 + \alpha_N} + C_L \frac{VDD}{2I_{DON}}\right) \tag{3.4}$$

and

$$v_{TN} = \frac{V_{THN}}{VDD}, v_{TP} = \frac{V_{THP}}{VDD} \tag{3.5}$$

are the ratios of the threshold voltages of NMOS and PMOS transistors to the supply voltage.

To derive τ_{PLH} of a DCVSL inverter, Fig. 29 shows the case where the QN output is rising. Note that MP1, the transistor that pulls QN up, is triggered by QP, not DP. In other words, the input signal for the rising output QN, is QP. However, propagation delay τ_{PLH} is defined as the delay between the time when the rising output (in this case QN) and the falling input (DP) of the inverter reaches VDD/2. Then, as shown

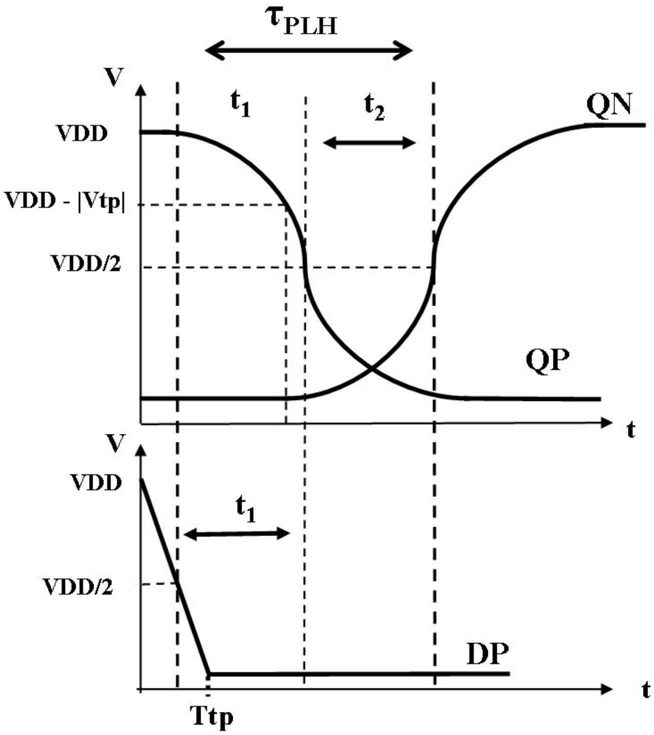


Fig. 29. Propagation delay derivation for $\tau_{PLH} = t_1 + t_2$

in Fig 29, we can represent τ_{PLH} as the summation of two delay components, t_1 and t_2.

$$\tau_{PLH} = t_1 + t_2 \tag{3.6}$$

where t_1 is determined by the speed of the NMOS pull-down transistor MN2 and is given by (3.7).

$$t_1 \quad = \quad \tau_{05\_LH} - \frac{T_{tp}}{2} \tag{3.7}$$

To find t_2, we approximate QP as a linear ramp, just as we do with the input signals DN and DP when deriving (3.2), since we assumed that the input and output

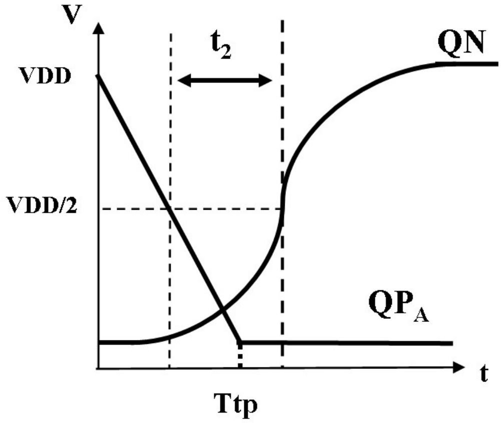


Fig. 30. Approximation of t_2

signals have similar slew-rates. Then, we obtain t_2 just like we found τ_{PHL}, as shown in Fig. 30 where QP_A is the linearly approximated QP:

$$t_2 = T_{tp}\left(\frac{v_{TP}+\alpha_P}{1+\alpha_P} + C_L\frac{VDD}{2I_{DOP}}\right) - \frac{T_{tp}}{2} \tag{3.8}$$

As mentioned earlier, the expressions for τ_{PHL} and τ_{PLH} (given in (3.3) to (3.8)), are derived ignoring the brief current conduction of NMOS transistor (MN1) for τ_{PLH} and that of PMOS loads (MP2) for τ_{PHL}. This assumption results in optimistic delay expressions. In reality, for τ_{PHL}, the PMOS load transistor conducts crowbar current during the output transition, reducing the output-node discharge current to be less than I_{DN}.

This reduction creates an error factor in the delay model, that is related to the “internal configuration ratio” (WP / WN assuming same length). The internal configuration ratio of an inverter affects the delay, particularly given deep-sub-micron-technology field effects such as velocity saturation [40].

Therefore, we propose the following DCVSL equations:

$$\begin{aligned}
\tau_{PHL} &= K_{HL} \times \left[T_{tn}\left(\frac{v_{TN}+\alpha_N}{1+\alpha_N} + C_L\frac{VDD}{2I_{DON}}\right) - \frac{T_{tn}}{2}\right] \\
\tau_{PLH} &= K_{LH} \times \left[T_{tn}\left(\frac{v_{TN}+\alpha_N}{1+\alpha_N} + C_L\frac{VDD}{2I_{DON}}\right) - \frac{T_{tp}}{2}\right. \\
&\qquad \left. +T_{tp}\left(\frac{v_{TP}+\alpha_P}{1+\alpha_P} + C_L\frac{VDD}{2I_{DOP}}\right) - \frac{T_{tp}}{2}\right]
\end{aligned} \tag{3.9}$$

where

$$\begin{aligned}
K_{HL} &= \left(\gamma_N + \frac{\zeta_N}{(WP/WN)}\right)^{-1} \\
K_{LH} &= \left(\gamma_P + \frac{\zeta_P}{(WP/WN)}\right)^{-1}
\end{aligned} \tag{3.10}$$

Note that γ_P, ζ_P, and γ_N, ζ_N are empirical correction factors obtainable from simulations, and should be constant across transistor sizes and loading conditions for a given technology. Note the τ_{PLH} correction factor, K_{LH}, is proportional to (WP / WN), because τ_{PLH} strongly depends on the NMOS transistor, for the PMOS pull-up transistor is controlled by the falling output, as explained earlier.

The voltage dependence of the load capacitance should be considered when calculating C_L. For a DCVSL inverter under test (IUT), such as the one shown in Fig. 27, load capacitance includes the input capacitance of the following fan-out stages, interconnect capacitance of the routing and capacitance due to the PMOS load transistor of the IUT itself. Note that the transition of interest is from VDD to VDD/2 and from 0 to VDD/2 for falling and rising outputs, respectively. We demonstrated that τ_{PLH} is inherently larger than τ_{PHL} (the rising output waits for the falling output to begin it's transition). For the falling output QP, since QN will wait for QP, MP1 will be in saturation while QP falls to VDD/2. For the rising output QN, we can assume that QP will fall enough for MP2 to have V_{DSOP} before QN begins rising, due to the inherent delay asymmetry of DCVSL. Then, MP2 will be in saturation during the transition of QN from 0 to VDD/2. Therefore, we safely assume that the PMOS transistors of the IUT (MP2 for QN and MP1 for QP) contribute saturation gate capacitance to the output. Similar analysis can be performed to the gate capacitance of the following fan-out stages to determine their operating region and capacitance.

2. Model Accuracy

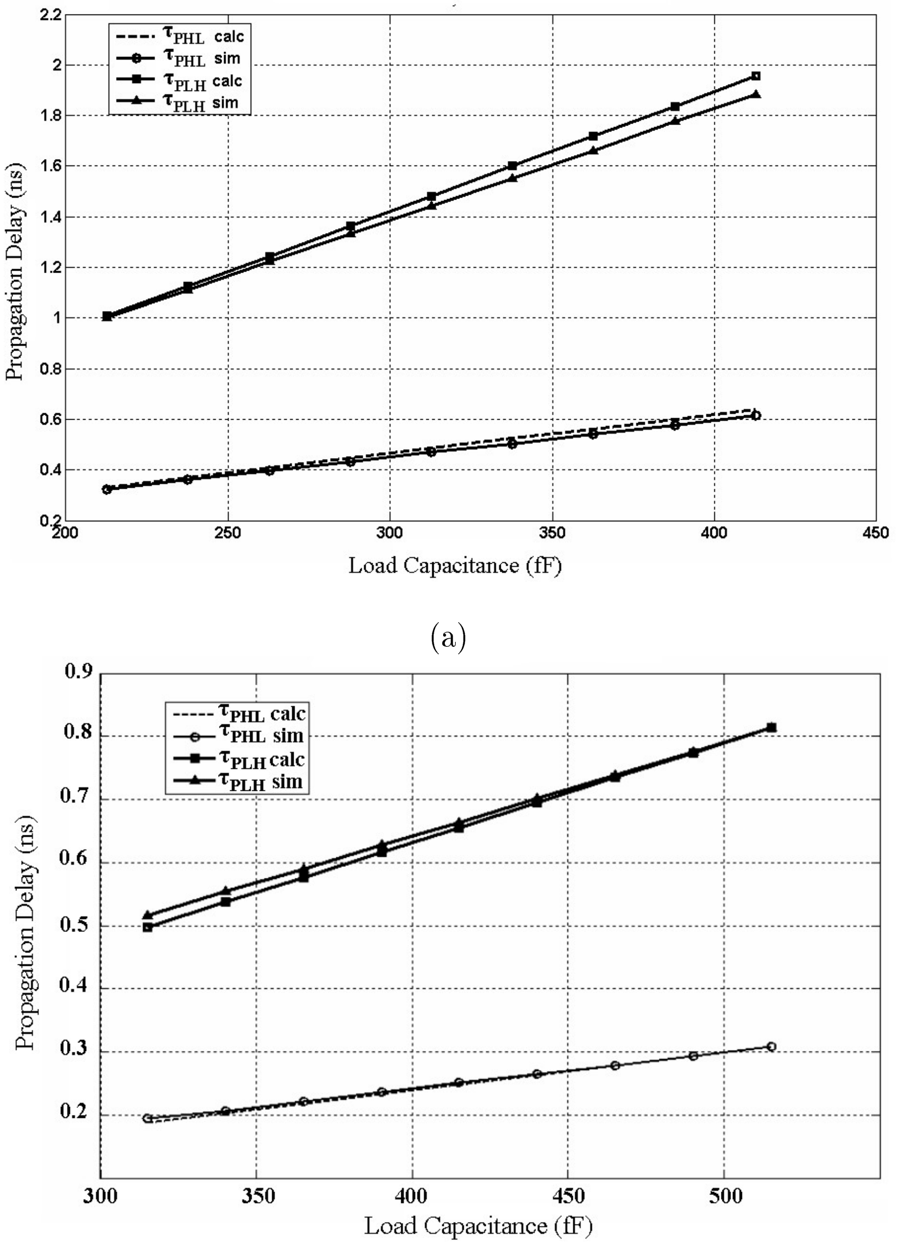
To test the accuracy of the proposed delay models of (3.9), we compare the calculated delay values to the results of schematic simulations, for 0.18μm and 0.13μm CMOS technologies. Table IX lists the values of α_N, α_P,γ_N, γ_P and ζ_N, ζ_P that we used.

Table IX. Values of DCVSL delay model empirical correction factors

Technology	α_N	γ_N	ζ_N	α_P	γ_P	ζ_P
TSMC 0.18 μm	1.1	0.26	0.403	1.4	0.36	0.245
UMC 0.13 μm	1.3	0.3	0.44	1.5	0.39	0.28

To simulate realistic input and output waveforms for the target applications, we place DCVSL inverters in a three-stage ring oscillator setting with capacitive loading at each stage and vary the load capacitors as well as transistor sizes. Fig. 31 (a) and Fig. 31 (b) compare the calculated values of τ_{PLH} and τ_{PHL} from (3.9) to their circuit-level simulated values for 0.18μm and 0.13μm technologies, respectively.

Table X list the simulated and calculated values of the propagation delays for various transistor ratios. The model error – defined as the ratio of the difference between the calculated and simulated delays over the simulated delay – is within ± 4 % for τ_{PHL} and within ± 8 % for τ_{PLH}, quite good for a closed-form model that avoids complex expressions and provides insight to the designer.



(a)

(b)

Fig. 31. Comparison of calculated vs. simulated values of τ_{PLH} and τ_{PHL} (a) for (WP/WN)=1.33 in 0.18μm technology (b) for (WP/WN)=1.57 in 0.13μm technology

Table X. A list of calculated and simulated values of τ_{PLH}, τ_{PHL} and model error for various transistor configurations

TSMC 0.18 μm technology						
$\frac{WP}{WN}$	τ_{PLH} (ps) calculated	τ_{PLH} (ps) simulated	τ_{PLH} error	τ_{PHL} (ps) calculated	τ_{PHL} (ps) simulated	τ_{PHL} error
1	1528	1420	7.6 %	410.6	398.7	2.9 %
1.33	1719	1660	3.5 %	562.5	543	3.6 %
1.66	1499	1586	-5.4 %	578.4	584.2	-1 %
0.8	1381	1291	7 %	322.2	381.1	1.3 %
0.66	1263	1208	4.5 %	264.8	267.7	-1.1 %
0.5	1439	1454	-1 %	258.2	261.1	-1.1 %
1.19	1495	1417	5.5 %	451.1	436.6	3.3 %
UMC 0.13 μm technology						
$\frac{WP}{WN}$	τ_{PLH} (ps) calculated	τ_{PLH} (ps) simulated	τ_{PLH} error	τ_{PHL} (ps) calculated	τ_{PHL} (ps) simulated	τ_{PHL} error
1.57	576.7	589.7	-2.2 %	218.1	220.7	1.1 %
1.37	550.7	545.1	1 %	192.2	192.1	0.1 %
1.12	622.5	591.9	5.1 %	192	187.4	2.4 %
1	534.1	509.2	4.9 %	154.9	153.4	1 %
0.8	797	742	7.3 %	204.3	198.5	2.9 %
0.61	745	726	2.6 %	169.6	170.7	0.6 %
0.5	816	829	-1.5 %	170.5	172.8	-1.3 %

3. Process Variations

The proposed model is also tested over the process corners provided in the technology model. Note that the values of I_{DO}, V_{DSO} and V_{TH} change over process corners and for each process corner, the new values should be used. However, the empirical correction factors are kept constant over the corners and the values given in Table IX are used, to test their sensitivity to process variations.

It is observed that while individual values of the model errors in Table X vary slightly, all of the model errors for the reported designs in Table X are still less than 8% worst case accuracy. This shows that we can use the same values for the correction factors over process corners. The effect of process variations on the values of empirical correction factors, is minimal and the proposed model provides the reported accuracy over process variations.

4. Discussion on DCVSL Delay Behavior

The delay analysis shows that the rising output of a DCVSL cell is inherently lagging the falling output since the PMOS that pulls the rising output up, has to wait for the falling output. The delay expressions of (3.9) show that τ_{PLH} has an extra delay component when compared to τ_{PHL} and therefore is larger. Note that increasing the size of PMOS loads to decrease t_2 of τ_{PLH}, increases the load capacitance and the overall delay of the inverter. Also, increasing the size of PMOS loads to have similar current driving capability as NMOS transistors, results in a mid-transition slow-down in the falling output.

To demonstrate this mid-transition slow-down, we set WP/WN=3 in 0.18μm technology and simulate DCVSL inverters in a ring oscillator configuration and obtain the voltage and current waveforms of Fig. 32. For the QP waveform, the slow-down occurs when MP2 and MN2 are ON simultaneously and when they have similar drain currents that compete against each other. Note that the inherent lead / lag asymmetrical shape of DCVSL output waveforms QP and QN extends the duration when PMOS and NMOS are both ON, causing this slow-down to effect τ_{PHL} considerably.

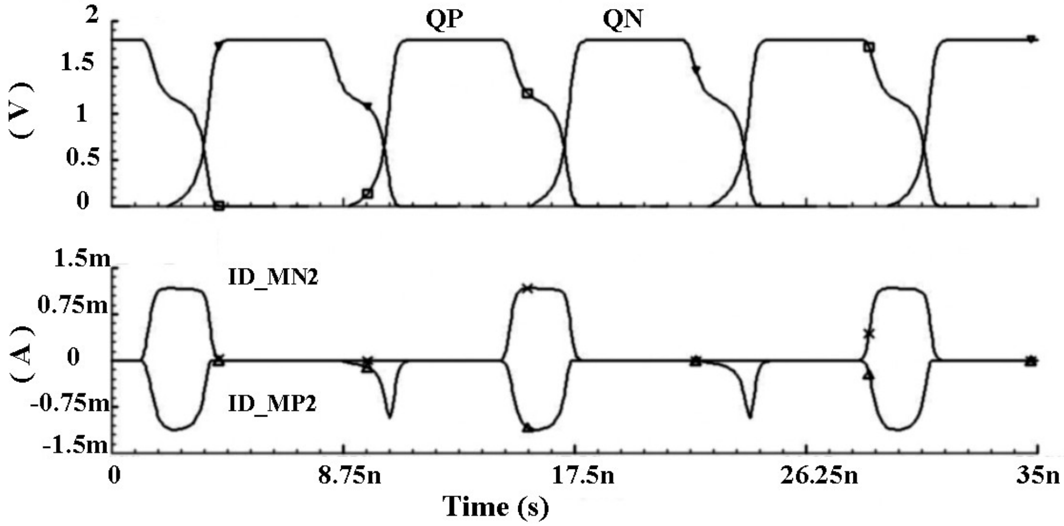


Fig. 32. Simulated voltage and current waveforms of a DCVSL inverter in 0.18μm for WP/WN=3, demonstrating mid-transition slow-down

To avoid this slow-down, ensure that the PMOS device is sized such that its current drive is less than that of the NMOS transistor. This shows that the delay bottleneck of DCVSL circuits that stem from a large τ_{PLH} can not be corrected by increasing the size of PMOS transistors and another solution is needed to improve the total propagation delay.

D. Proposed DCVSL-R Circuit

DCVSL circuits have a larger τ_{PLH} than τ_{PHL} and based on the discussion from Section III, we conclude that increasing the PMOS transistor sizing does not necessarily help this problem. The inherent delay problem of DCVSL structures is addressed in [7], [8] without going into a detailed analysis. The authors of [7] propose two types of enhanced precharge DCVSL (EDCVSL) structures that operate at 100MHz. The first structure prevents the crowbar current flow that was mentioned earlier. The second structure is proposed as a solution to prevent the asymmetry between the falling and rising outputs of the circuit. To solve the delay asymmetry problem, the authors of [8] add a PMOS pull-up network to the DCVSL scheme. However, all of the proposed circuits require several additional transistors, eliminating the benefit of low transistor count of DCVSL circuits and increasing internal parasitics, which are a primary concern in RF applications.

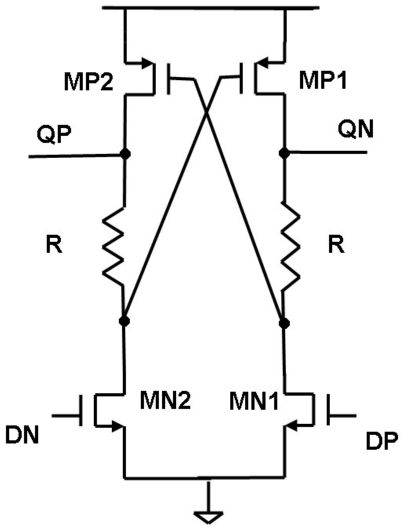


Fig. 33. Proposed DCVSL-R circuit

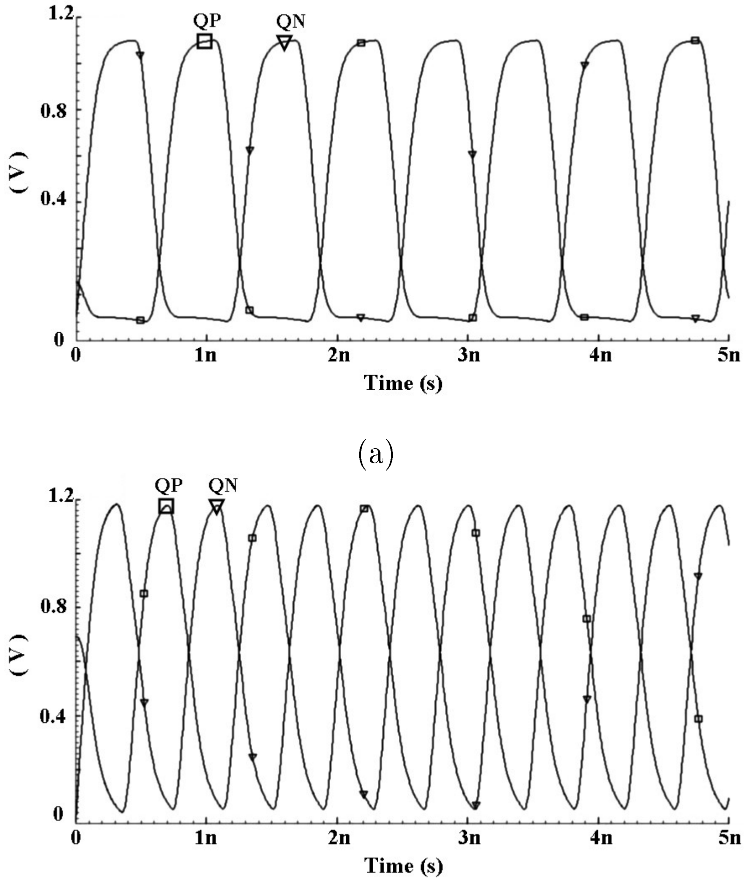
Fig. 33 shows our proposed solution, DCVSL with resistive enhancement (which we call DCVSL-R), to solve the inherent extra delay component of τ_{PLH} in DCVSL circuits. The resistors increase the gate overdrive of the PMOS load transistors. If we consider the switching conditions of Fig. 27, when MN2 turns on and starts conducting current, the gate voltage of MP1 is :

$$V_{G\_MP1}(t) = V_{QP}(t) - I_{D\_MN2}(t) \times R \tag{3.11}$$

Note that in the delay derivations for DCVSL circuits, we assumed that the transistors operate in saturation region until the output reaches VDD/2. However, in the DCVSL-R circuit, the drain node of the NMOS transistors (also the gate of the PMOS transistors) drop quickly as shown in (3.11), and push the transistors into linear region. Therefore, the delay analysis of DCVSL-R involves more complex expressions than the closed-form ones derived for DCVSL.

However, based on Section III, an intuitive analysis can explain how the DCVSL-R circuit improves the propagation delay of DCVSL circuits. The extra delay element t_1 of (3.6) in τ_{PLH} is due to MP1 waiting for QP to drop. By adding the resistors, we put an additional load to the drain of the NMOS transistors and increase the voltage drop at the gates of PMOS to turn on the PMOS transistors faster and minimize this waiting time. Therefore, based on the value of the resistor, we can achieve $\tau_{PLH} = \tau_{PHL}$ which results in symmetrical output waveforms. More importantly, due to the reduced τ_{PLH}, the total delay of the DCVSL inverter will be reduced.

To demonstrate, we simulate DCVSL and DCVSL-R cells in a ring oscillator setting and plot the outputs for both, in Fig. 34. For ease of comparison, transistor sizes of both cells are the same and only resistors are added to the DCVSL-R cell. The waveforms of Fig. 34 show how rising output lags falling output in DCVSL case and that this problem is eliminated in the DCVSL-R case.



(a)

(b)

Fig. 34. Inverter output waveforms in a ring oscillator setting for WP/WN=1 (a) for conventional DCVSL (b) for proposed DCVSL-R with R=380 ohms

Note that by adding additional resistance to the drain of NMOS transistors, τ_{PHL} is degraded due to a larger time constant. [37] provides an analysis on the effects of drain resistance in the delay degradation. However, as long as we satisfy

$$R_{MN} > R \tag{3.12}$$

where R_{MN} is the resistance of the NMOS transistor in linear region and R is the added extra resistor, the degradation of τ_{PHL} due to R will be insignificant when compared to the improvement we obtain in τ_{PLH}.

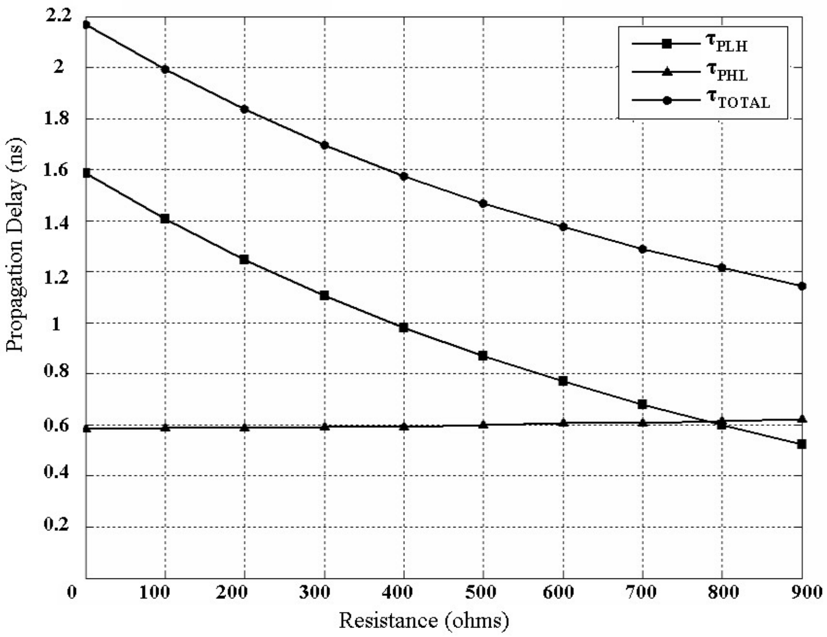


Fig. 35. Circuit-level simulation results for τ_{PLH}, τ_{PHL} and τ_{TOTAL} values vs. the resistance R for a DCVSL-R inverter with (WP/WN)=1.66 in 0.18μm technology

Fig. 35 shows circuit-level simulation results of the values of τ_{PLH}, τ_{PHL} and τ_{TOTAL} with respect to the value of R, for DCVSL-R inverters where

$$\tau_{TOTAL} = \tau_{PLH} + \tau_{PHL} \tag{3.13}$$

The values of these delays when R=0 represent the delay performance of DCVSL inverter. Note that in Fig. 35 (a), as R increases (and is kept at a reasonable value based on (3.12), the improvement in τ_{PLH} is much more significant than the degradation of τ_{PHL}, and the total effective propagation delay improves considerably. The key observation is that the total propagation delay of the DCVSL-R circuit, (which determines frequency of operation when used in an oscillator) is significantly reduced (46% reduction for R=800ohms that achieves symmetric τ_{PLH} and τ_{PHL}), compared to the DCVSL circuit.

Note that if R is increased further, than the recommended range (3.12), the degradation in τ_{PHL} will start becoming more visible and the improvement in τ_{TOTAL} will slow down. The delay asymmetry will occur again, resulting in τ_{PHL} to be larger than τ_{PLH}. Fig. 36 demonstrates this delay behavior when R is increased further than the recommended range and point of symmetry. It should be noted that while the total propagation delay decreases, despite a much smaller slope, the output waveform symmetry is significant and values of R that would generate similar τ_{PLH} and τ_{PHL}, hence, symmetric output waveforms, should be preferred especially in clocking circuits.

Similar to CML circuits, speed performance of the DCVSL-R circuit might be affected by resistor value variations. Mismatch between the resistors in the differential branches of the circuit - which would result in asymmetric outputs where one output is faster than the other - is minimized by symmetric layout techniques and the use of dummy resistors. However, similar to CML circuits, speed performance of the DCVSL-R circuit might be affected by process variations on the absolute value of the resistors.

While in ring oscillator based VCOs, frequency tuning controls can take care of such variations, in frequency dividers the designer should leave enough margin in the maximum operating frequency based on process variation expectations of a design technology. Relative mismatch between the resistors in the differential branches of the circuit can however, be effectively minimized by symmetric layout techniques and the use of dummy resistors.

Note that the DCVSL-R circuit does not speed up by limiting the output signal

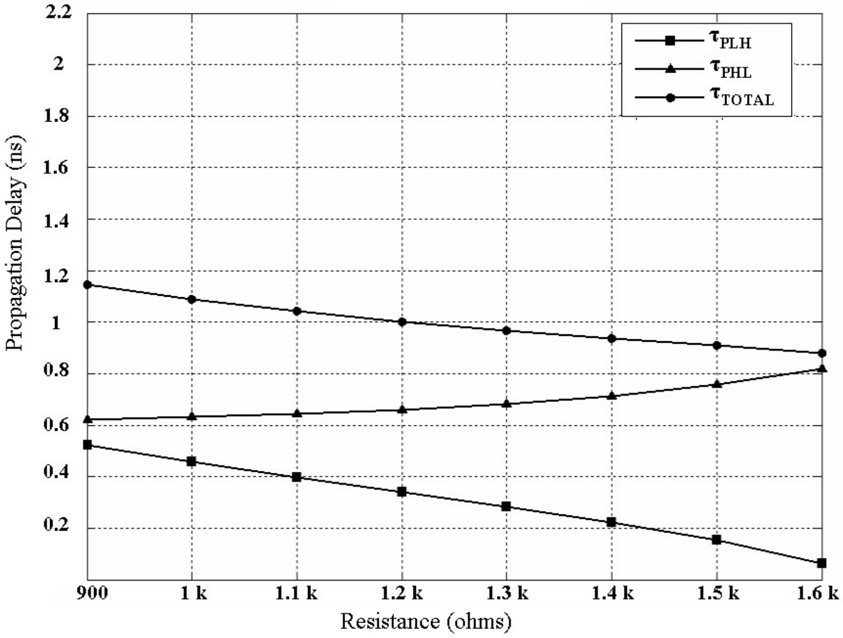


Fig. 36. Circuit-level simulation of propagation delay vs. the resistance R for a DCVS-L-R inverter with (WP/WN)=1.66 in 0.18μm technology for values of R past the point of symmetry

swing. Rather, the speedup is achieved by eliminating an inherent additional delay of DCVSL circuits. Therefore, it maintains the rail-to-rail switching, making it very suitable for low voltage applications.

CHAPTER IV

A LOW POWER FREQUENCY SYNTHESIZER WITH DCVSL-R DIVIDERS

In Chapter II, we implemented an integer-N phase-locked loop (PLL) based frequency synthesizer for ZigBee wireless transceiver applications at the 2.4GHz operating-frequency band that consumed 15mW total power. In Chapter III we discussed various circuit techniques to implement the RF frequency dividers of a synthesizer and concluded that the proposed DCVSL-R circuit provides high speed with low power consumption and with small input clock capacitance.

In this chapter, we present a new frequency synthesizer, based on the one that was implemented in Chapter II but utilizes DCVSL-R cells in its frequency dividers. Therefore, we will focus on the proposed speed-enhanced DCVSL-R circuits in the high-frequency programmable divider of the PLL, optimizing the power consumption. We will also show that the DCVSL-R based dual-modulus prescaler (DMP) and the buffer that drives it, have the lowest combined power consumption among the reported similar divider implementations at the same operating frequency. To the authors' knowledge, this work is the first to demonstrate DCVSL circuits in gigahertz range frequency dividers.

A. Implementation

The frequency synthesizer is implemented in TSMC 0.18μm CMOS technology and is based on the ZigBee synthesizer of Chapter II that was also reported in [4]. The synthesizer of Chapter II employed a TSPC prescaler in its programmable dividers and the TSPC dual-modulus prescaler consumed 2.6mW in 0.18μm technology. However, the large capacitance of TSPC circuits' input-clock path resulted in an additional 2.6mW of buffer power. To solve this problem and improve the total power consumption, the proposed new PLL employs a DCVSL-R based dual-modulus prescaler.

Fig. 37 shows the PLL block diagram. The center frequencies of 16 ZigBee channels in the targeted band are in the range from 2.405GHz to 2.48GHz and are spaced by 5MHz, which is the reference frequency of this PLL. The divide-by-4 circuit before the PFD is employed to minimize the effect of coupling from external reference signal to the sensitive nodes of the PLL and to reduce resulting spurs. Then, the strong external reference signal is at 20MHz, and the desired reference frequency of 5MHz is generated by the internal divide-by-4 circuit. Therefore, any coupling from the strong input clock pin and routing to the PLL control node within the microchip and on the PC board will be pushed to appear at 20MHz offset, where spur suppression will be better than it would be at 5MHz offset.

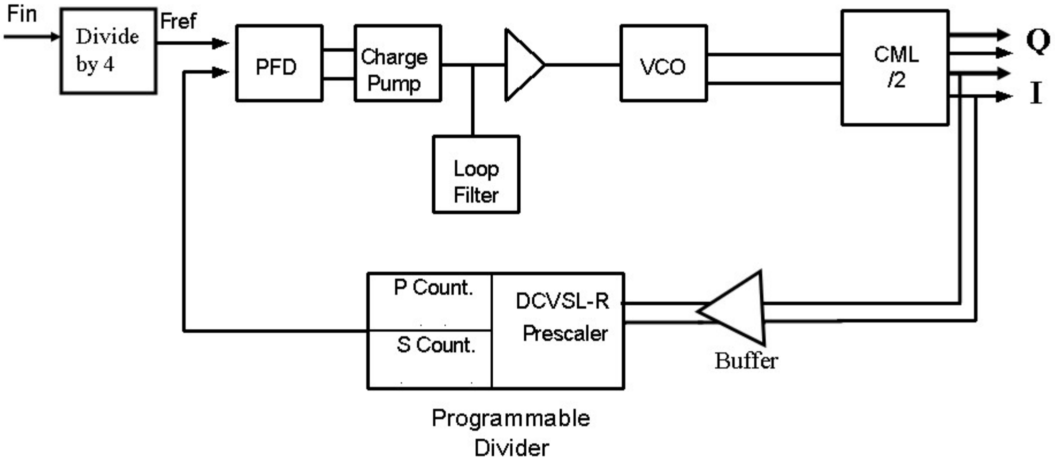


Fig. 37. Block diagram of the new PLL with DCVSL-R divider

The LC-tank VCO operates at twice the channel frequency range (4.81GHz - 4.96GHz). A divide-by-2 circuit generates quadrature LO signals to be used by up/down conversion mixers of a transceiver. This divide by 2 circuit is implemented with CML instead of DCVSL-R circuit for quadrature signal generation with very small IQ mismatch and with smaller controlled swing rather than the rail-to-rail swing of DCVSL-R circuit, to provide smaller swing at the LO to improve mixer linearity.

As discussed in Chapter II, on system level the pulse-swallow divider consists of a 5-bit programmable (P) counter, 4-bit channel selections (S counter), and a 15/16 dual-modulus prescaler. The overall programmable division ratio of the pulse-swallow divider is given by N:

$$N = 481, 482, ..., 495, 496 \tag{4.1}$$

The prescaler speed limitation arises during /15 operation, which employs the divide-by-3 mode of the /3 or /4 circuit. The critical delay path in the /3 circuit and the timing condition that the circuit should satisfy is given by:

$$TD_{DFF2\_Slave} + 2 \times TD_{NOR2} \leq \frac{T_{CLK}}{2} \tag{4.2}$$

where TD_{DFF2_Slave} is the delay of the slave latch of the second flip-flop, TD_{NOR2} is the delay of the two input NOR gate and TD_{CLK} is the input clock period of the prescaler. The delay values TD include not only the propagation delay of those circuits but also the corresponding setup and hold times. Note that FIN is the highest frequency in the divider and therefore half of it's period sets a very strict time limitation on the divide-by-3 circuit.

In this synthesizer, the flip-flops and gates shown in Fig. 12 are implemented with DCVSL-R structure. Fig. 38 shows the D flip-flop implementation based on DCVSL-R and use high-resistivity poly resistors. The layout of the prescaler is shown in Fig. 39 where the whole 15/16 prescaler takes $71\mu m \times 24\mu m$ area. Note that there are two additional dummy resistors, one for each side, for matching purposes. Also note that despite the addition of resistors, the total area of the prescaler is very small due to the reduced capacitance and stacking, small transistor sizing is used. The TSPC prescaler designed in Chapter II in the same technology takes $128.5\mu m \times 18.5\mu m$ area. Since the operating frequency falls down to a few hundred MHz frequency range at the output of the prescaler, the P and S counters are implemented with standard complementary CMOS logic.

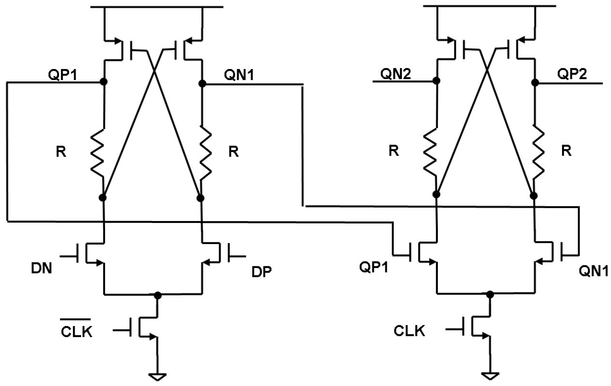


Fig. 38. Circuit level diagram of D flip-flops used in the DCVSL-R based prescaler

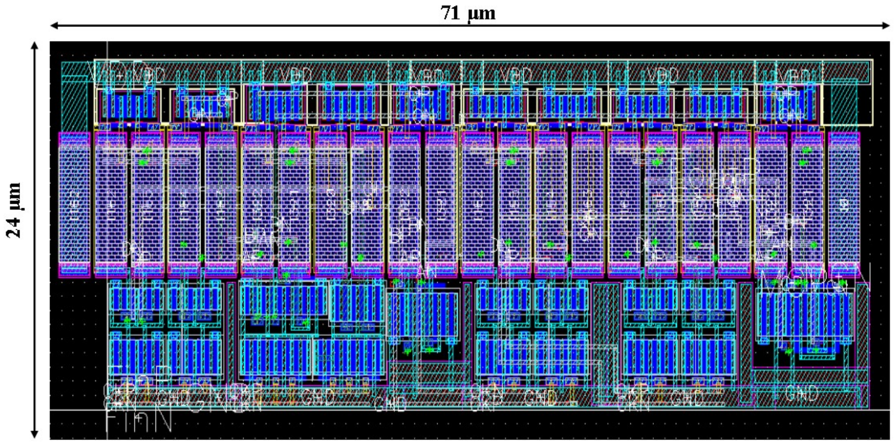


Fig. 39. Layout of the DCVSL-R based dual modulus (15/16) prescaler

B. Measurement Results

The frequency synthesizer is fabricated in TSMC 0.18μm CMOS, mounted on an FR-4 PCB, and measured. An on-chip open-drain buffer measures the PLL output. Table XI summarizes the PLL measurement results.

Table XI. Measured performance summary of the frequency synthesizer

Frequency Synthesis	2.405GHz - 2.48GHz
VCO Frequency	4.4GHz - 5.22GHz
Technology	0.18μm CMOS
Spur Suppression	-48 dBc at 5MHz offset -55 dBc at 10MHz offset
Phase Noise	-135 dBc/Hz at 10MHz offset -127 dBc/Hz at 3.5MHz offset
Settling Time	58μs
Power Consumption	8.3mW
Area	0.56 mm^2

The PLL output frequency spectrum is shown for the first channel, 2.405GHz operation, in Fig. 40. Spur suppression at this channel at 10MHz offset frequency is -55dBc/Hz. Fig. 41 illustrates the phase noise performance of the closed loop PLL for 2.405GHz while Fig. 42 shows the phase noise plot at 2.48GHz. Note that the phase noise is -135dBc/Hz at 10MHz offset frequency and it is -127dBc/Hz at 3.5MHz offset.

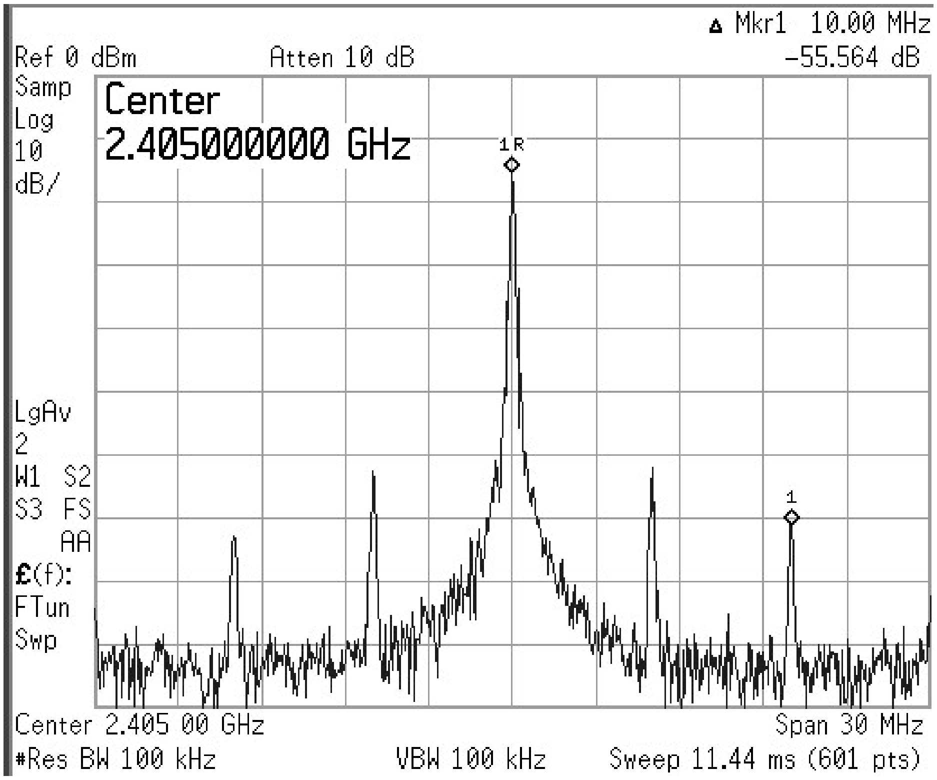


Fig. 40. Output frequency spectrum of the new synthesizer with DCVSL-R dividers at 2.405GHz

Fig. 43 displays the die micrograph, where the PLL occupies an area of $0.8mm$ by $0.7mm$. The settling time is shown in Fig. 44 where the settling time is 58μs and the overshoot is % 28.5. The synthesizer consumes 8.3mW total power. Note that operating the VCO at double the channel frequency increases the power consumption of the PLL. This is due to the generation of quadrature LO signals for ZigBee which employs OQPSK modulation.

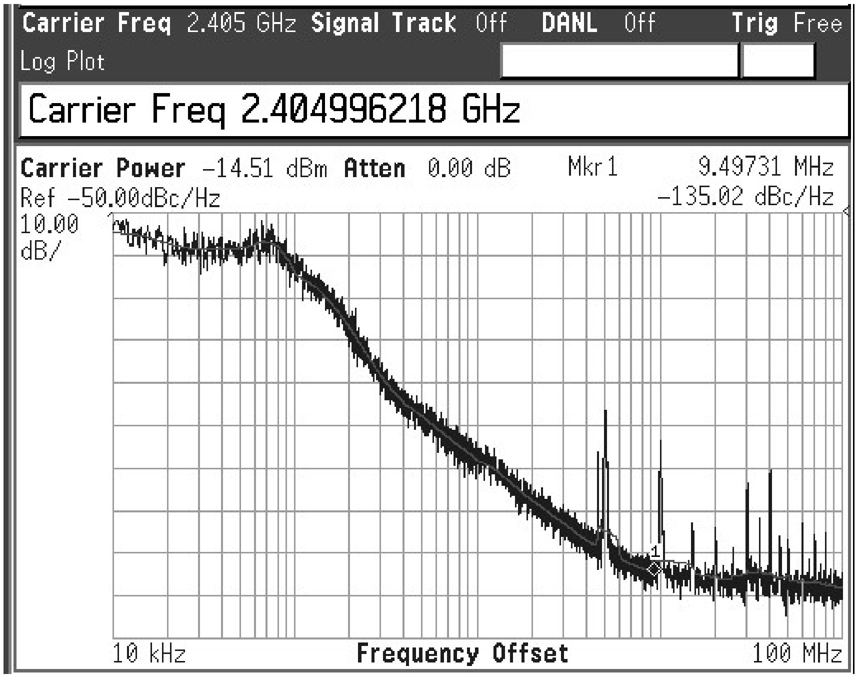

Carrier Freq 2.405 GHz Signal Track Off DANL Off Trig Free
Log Plot
Carrier Freq 2.404996218 GHz
Carrier Power -14.51 dBm Atten 0.00 dB Mkr1 9.49731 MHz
Ref -50.00dBc/Hz -135.02 dBc/Hz
10.00 dB/

Fig. 41. Phase noise spectrum of the new synthesizer at 2.405GHz

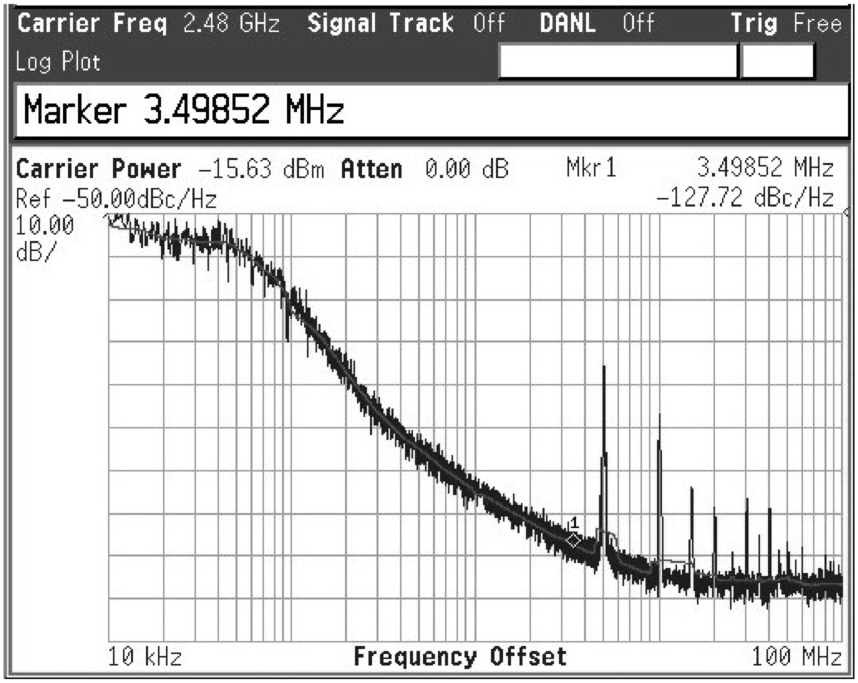

Carrier Freq 2.48 GHz Signal Track Off DANL Off Trig Free
Log Plot
Marker 3.49852 MHz
Carrier Power -15.63 dBm Atten 0.00 dB Mkr1 3.49852 MHz
Ref -50.00dBc/Hz -127.72 dBc/Hz
10.00 dB/

Fig. 42. New frequency synthesizer measured phase noise spectrum at 2.48GHz

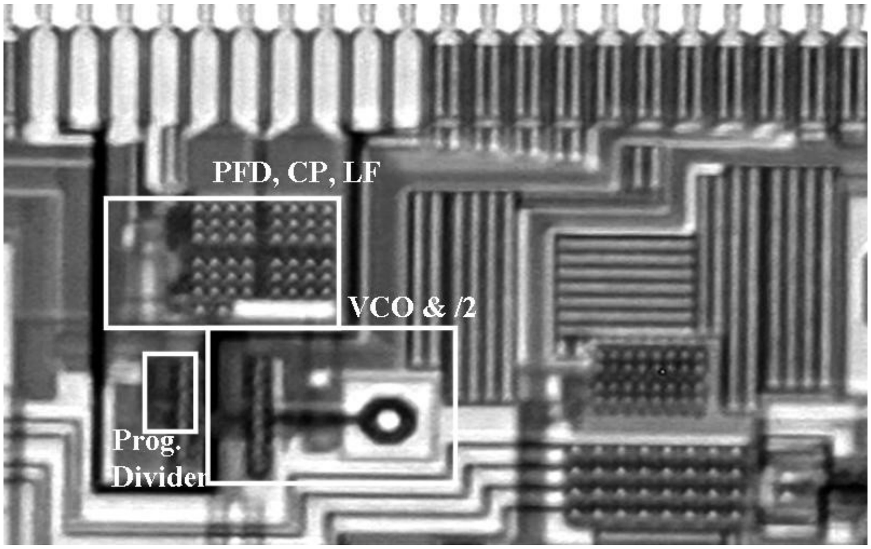
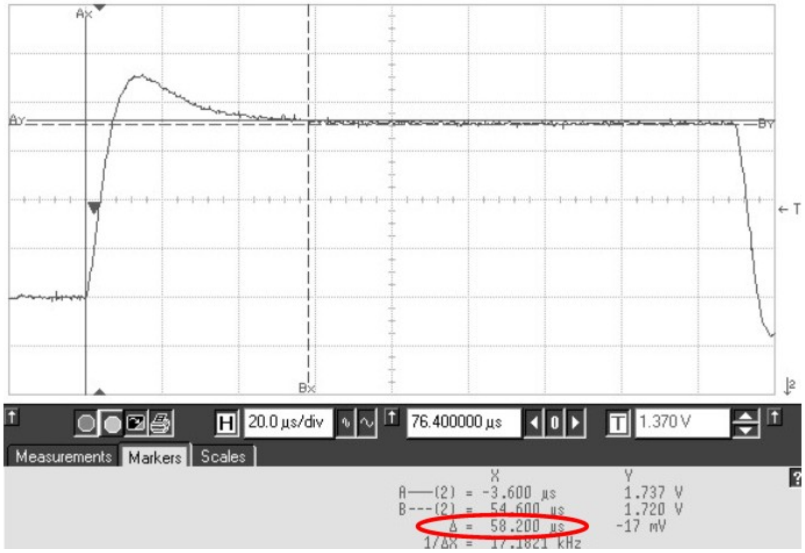


Fig. 43. Die micrograph of the new PLL



```
H 20.0 µs/div    76.400000 µs    0    T 1.370 V

             X               Y
A——(2) = -3.600 µs        1.737 V
B---(2) =  54.600 µs      1.720 V
     Δ =  58.200 µs      -17 mV
  1/ΔX =  17.1821 kHz
```

Fig. 44. New frequency synthesizer measured settling time

C. Discussion on Divider Performance

Power consumption of frequency dividers are determined by their division ratio, input frequency and the technology they are implemented in. While there are figures of merit [41], [42] that are proposed in literature that relate these parameters to have a common base of comparison, it is not trivial to do a fair comparison of various divider techniques when all of these parameters are different. This is because the effect of each parameter in the overall performance is not always linear as often predicted by figures of merit.

For instance, for an m stage divider, the power consumption will be dominated by the first x stages, the value of x depends on the input frequency and technology node. Then, after the first x stages, additional division stages will not increase the overall power consumption significantly. Therefore, an assumption of linear relation between the power consumption and the division ratio will not always give an accurate understanding on the performance of the divider.

Another issue to consider is if the divider is a fixed ratio or a multi-modulus divider. Frequency dividers whose divide ratio is a power of 2 could employ n cascaded /2 stages to divide by 2^n. In such a case, the timing constraint on the divider would come from each /2 stage that should operate fast enough at a negative feedback condition, at it's input clock speed. However, in a dual modulus prescaler, the division also involves a feedback that contains the modulus signal and logic gates that enforces certain output states to be skipped. This results in critical timing paths as the one shown in (4.2). Therefore, for a fair performance comparison, dual modulus prescaler circuits should be compared to other dual modulus prescalers rather than fixed division ratio circuits.

Based on the above discussion, when comparing the performance of various di-

Table XII. Performance comparison of the DCVSL-R prescaler with previously reported solutions

	[3]	[4]	[5]	[31]	**This Work**
Input Frequency	2.5 GHz	2.48GHz	2.5GHz	2.45GHz	2.48 GHz
Division Ratio	22 / 23	15 / 16	8 / 9	16 / 17	15 / 16
Circuit Implementation	SCL	TSPC	E-TSPC	TSPC with powerdown	DCVSL-R
Technology	0.24μm	0.18 μm	0.25 μm	0.18 μm	0.18 μm
Input Buffer Power	No buffers	2.6mW	1.1mW	Not specified	0.27mW
Prescaler Power	19mW	2.6mW	3.025mW	1.33mW	**0.8mW**
Buffer+Prescaler Total Power	19mW	5.2mW	4.125mW	Not Specified *	**1.07mW**

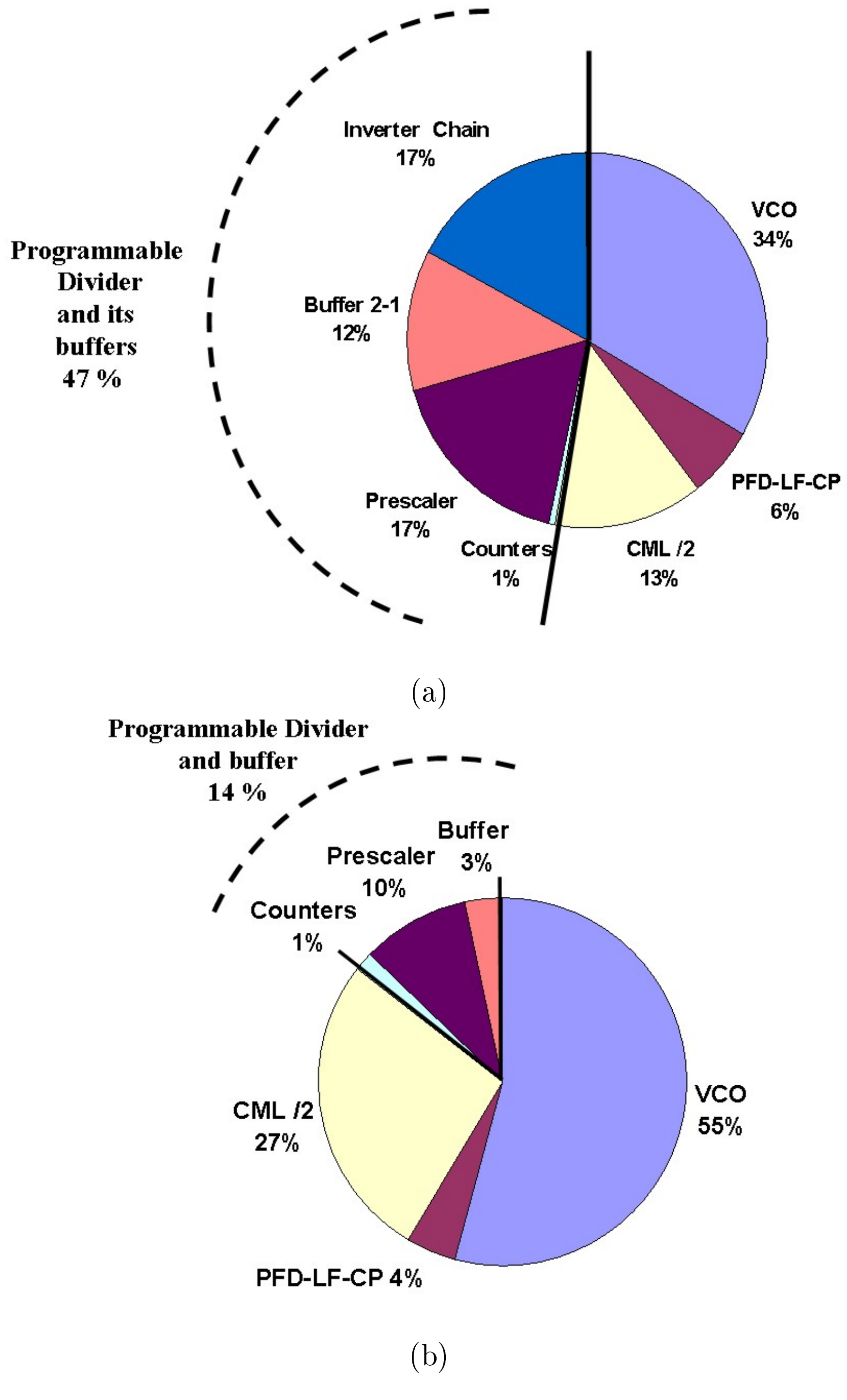
* The divider power is specified as 1.33mW but it is not specified if this includes the power consumption of the inverter chain buffer.

viders a safe approach is to compare them at similar operating conditions. Table XII shows a comparison of prescalers from literature that are used in frequency synthesizers and employ various circuit techniques. Since the power consumption is directly related to the operating frequency, all of these works feature a pulse-swallow divider with a prescaler input frequency of 2.5GHz, a popular operating frequency for wireless transceiver frequency synthesizers. They are also implemented in similar technology nodes and are using similar division ratios.

The proposed DCVSL-R based dual-modulus prescaler of the PLL consumes 0.8mW, while the buffer that drives it only consumes 0.27mW. The power consumption of the prescaler alone is not a sufficient metric, its driving-buffer power should also be taken into account as an indicator of the clock input capacitance of the prescaler and the prescaler's overall impact on the synthesizer power consumption. Note that this work has the lowest power consumption, 0.8mW, in its dual-modulus prescaler which demonstrates a 40% reduction from the other works in literature. It also demonstrates the lowest total power consumption for the prescaler and it's driving input buffer. Since DCVSL-R circuits provide a symmetrical differential non-stacked clock input loading to it's driving RF stage, no dummy dividers or differential to single ended converters are employed and the quality of the differential quadrature LO signals are maintained.

In Chapter II, it was shown that the power consumption of the programmable divider and its buffers constituted 47% of the 15mW total synthesizer power. The power consumption distribution of the old synthesizer with the TSPC divider is shown again for comparison in Fig. 45 (a) while the power distribution of the new synthesizer that features the DCVSL-R based dual-modulus prescaler is given in Fig. 45 (b). Note that in measurements, the total power consumption of the prescaler and the counters are measured since they are connected to a single supply domain. However since the measured performance is almost the same as the post-layout performance, we can deduct the individual power consumptions of the building blocks from the total measured power of the different supply domains.

It is seen that in the new design, the programmable divider and its buffer constitute only 14% of the 8.3mW total power. This verifies that the total frequency synthesizer power consumption can be significantly reduced by employing a frequency divider that employs a low power circuit techniques which also has small clock input capacitance.



(a)

(b)

Fig. 45. Power consumption distribution of the synthesizer with TSPC dividers and the new synthesizer with DCVSL-R dividers

CHAPTER V

RING OSCILLATORS USING DCVSL AND DCVSL-R DELAY CELLS

DCVSL inverter based delay cells, also called Lee-Kim delay cells [43] are often employed in ring oscillators. These cells provide a simple solution with easy frequency tuning, but are susceptible to supply variation as opposed to the more complex Maneatis delay cells [44] that offer better power supply rejection. However, the ring oscillator supply-noise-based PLL jitter can be minimized through supply noise cancellation schemes as in [45] and by employing on-chip voltage regulators. Other important performance metrics of ring oscillators include phase noise, power consumption and frequency of operation.

Frequency of operation is determined by the total delay of the unit cells of the oscillator which is closely related to power consumption. To optimize this speed and power trade off, we propose the DCVSL-R circuits to replace the conventional DCVSL delay cells of ring oscillators. As discussed in Section IV, DCVSL-R circuits provide less delay by improving the inherently slow τ_{PLH} of their DCVSL counterpart.

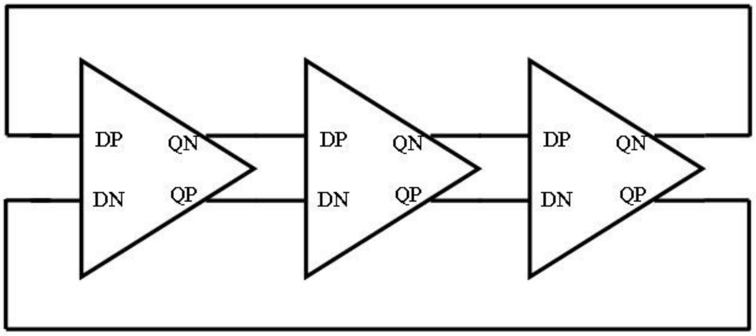
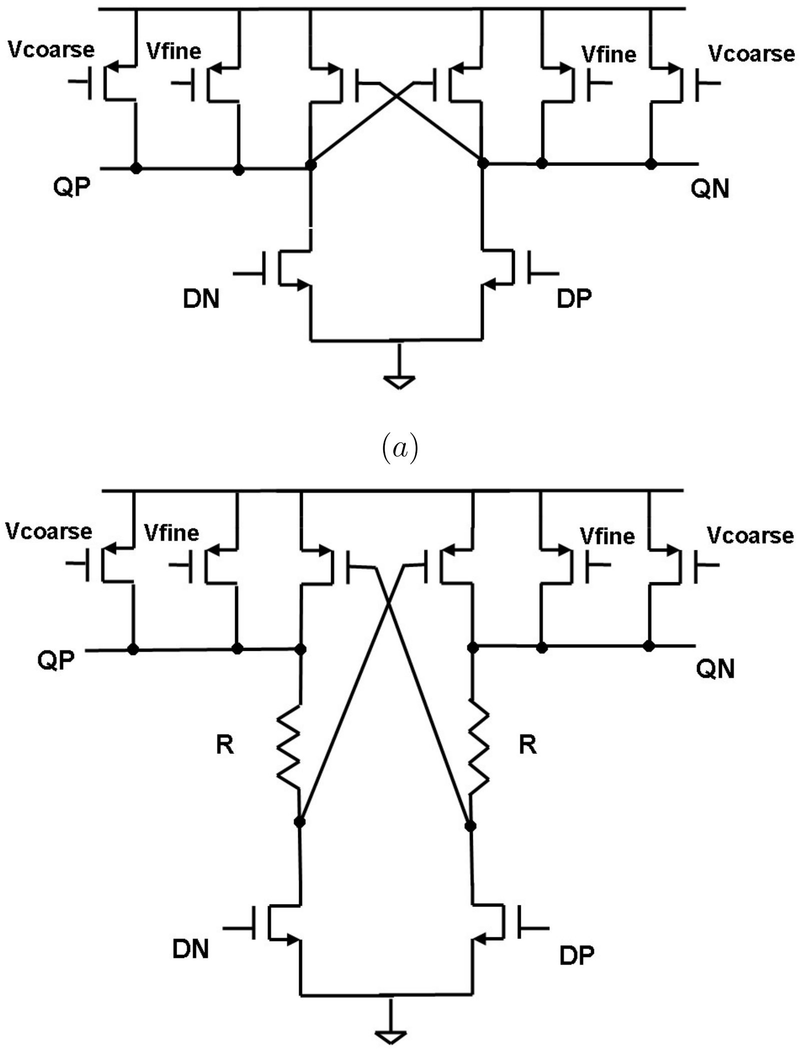


Fig. 46. Block diagram of the three stage ring oscillators

A. Ring Oscillator Design

To compare the two techniques, DCVSL and DCVSL-R, we implemented two ring-oscillator-based VCOs in 0.13μm CMOS process. Both are three-stage ring oscillators, as shown in Fig. 46. While OSC1 uses the standard DCVSL inverter based delay



(*a*)

(*b*)

Fig. 47. VCO delay cells (a) conventional DCVSL for OSC1 (b) proposed DCVSL-R for OSC1-R

cell of Fig. 47 (a), OSC1-R uses the proposed DCVSL-R based delay cell shown in Fig. 47 (b). To see the direct effect of the resistors in the speed, power and noise performance of the ring oscillators, we kept the transistor sizing of both oscillators the same and only added resistors to OSC1-R. We used high-resistivity poly resistors that implement 420 ohms with $1.5\mu m \times 5.9\mu m$ area in layout.

OSC1 is designed to target 2.4GHz operation, with coarse (VCOARSE) and fine tuning (VFINE) controls. Since the transistor sizes are the same, when operated at the same supply voltage, OSC1-R should give a higher operating frequency due to the improved delay performance. In terms of phase noise, since R is a cascode element on top of the input transistors, we expect the noise contribution of R to the phase noise to be negligible.

B. Measurement Results

The two oscillators are fabricated in UMC 0.13μm CMOS technology. The dies are packaged in a surface mount QFN type package and mounted on an FR-4 printed-circuit-board (PCB) for measurements. Oscillator outputs are connected to on-chip open drain buffers to drive an on-board RF balun that converts the differential outputs to a single node and drives the 50 ohms impedance of the spectrum analyzer.

In measurements, it is seen that OSC1-R oscillates at a higher frequency range (3.14GHz - 3.89GHz) than OSC1 (2.16GHz - 2.77GHz) at VDD=1.2V supply voltage. The frequency range is the tuning range of the oscillators, obtained through coarse and fine tuning controls. Fig. 48 and Fig. 49 demonstrate the tuning range of OSC1 and OSC1-R, respectively, for a supply voltage of 1.2V.

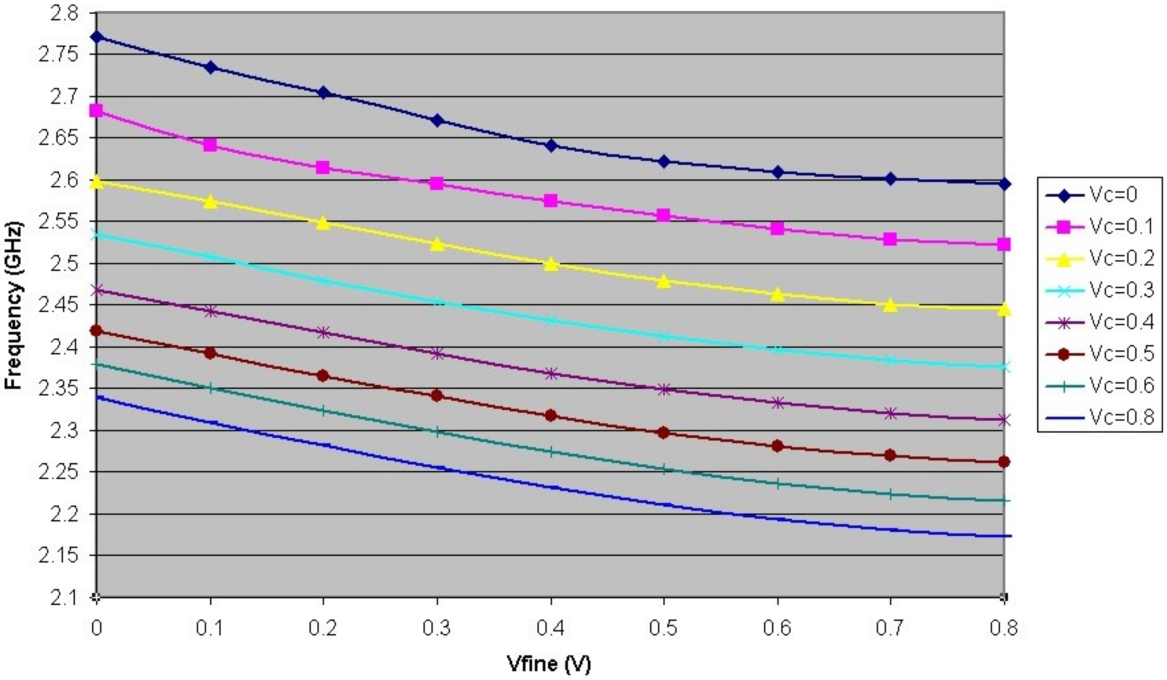


Fig. 48. Measured fine and coarse tuning range of OSC1 at 1.2V supply

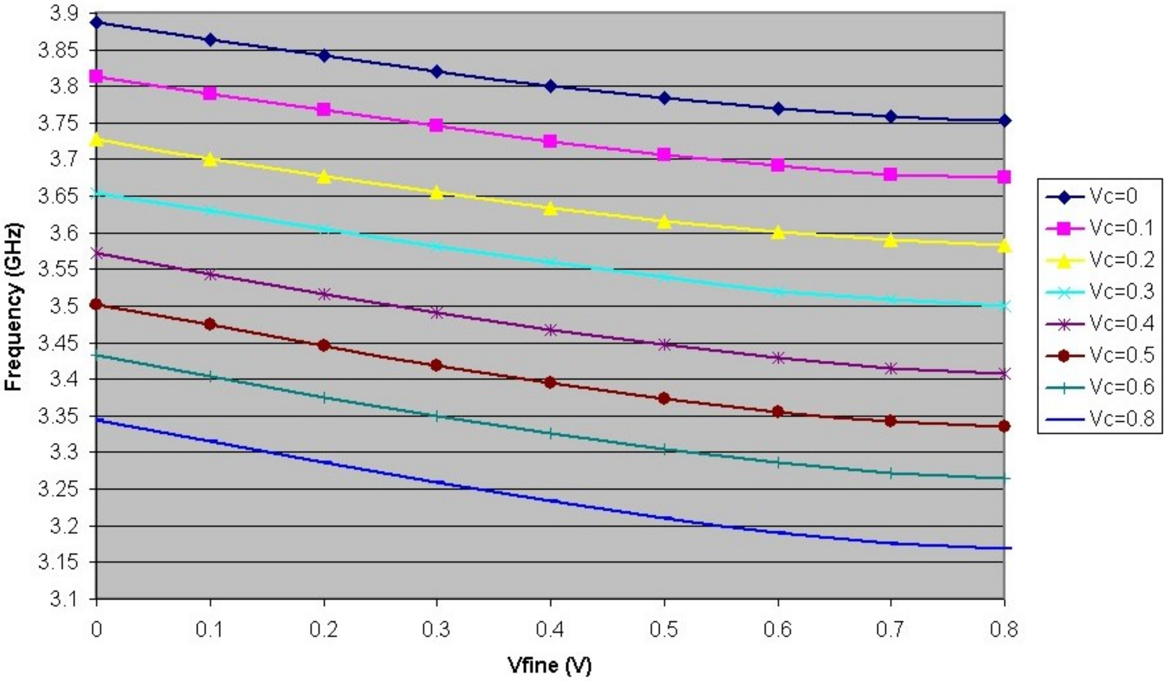


Fig. 49. Measured fine and coarse tuning range of OSC1-R at 1.2V supply

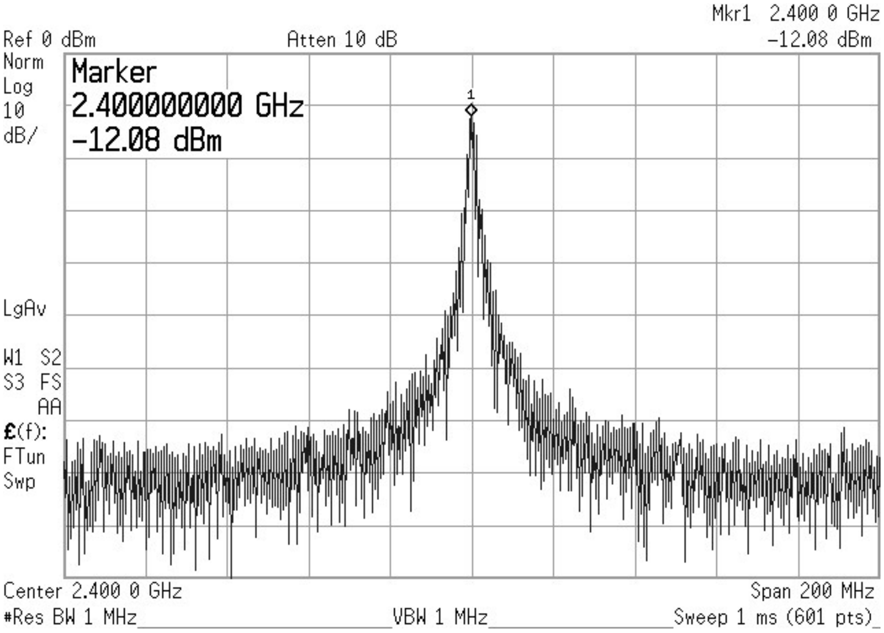


Fig. 50. Measured output frequency spectrum of OSC1 at 2.4GHz operation

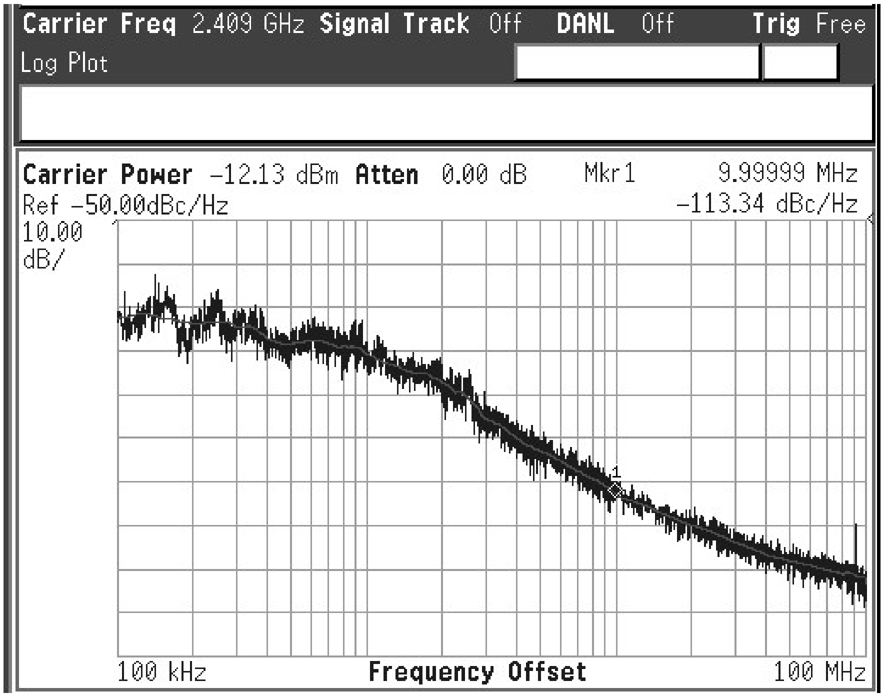


Fig. 51. Measured phase noise spectrum of OSC1 at 2.4GHz operation

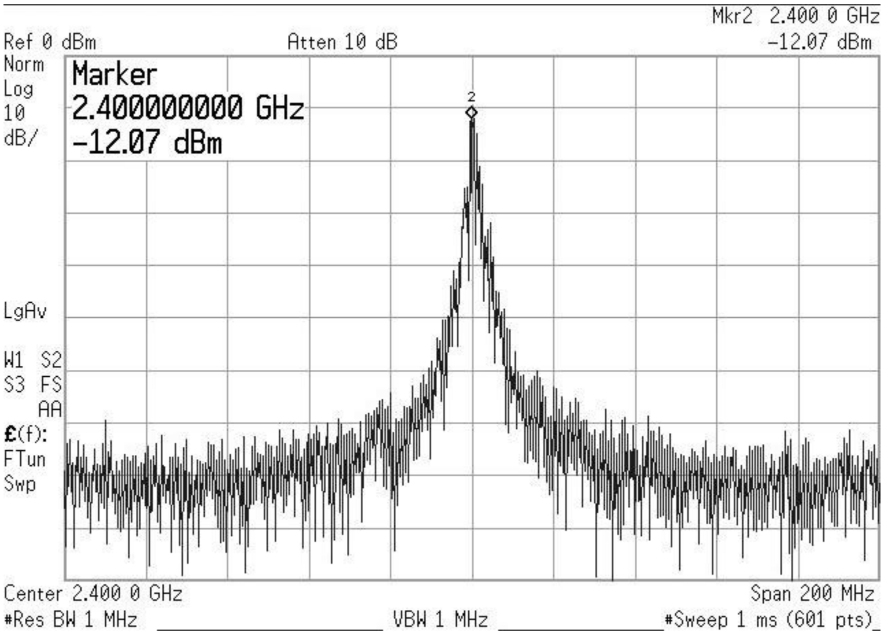


Fig. 52. Measured output frequency spectrum of OSC1-R at 2.4GHz operation

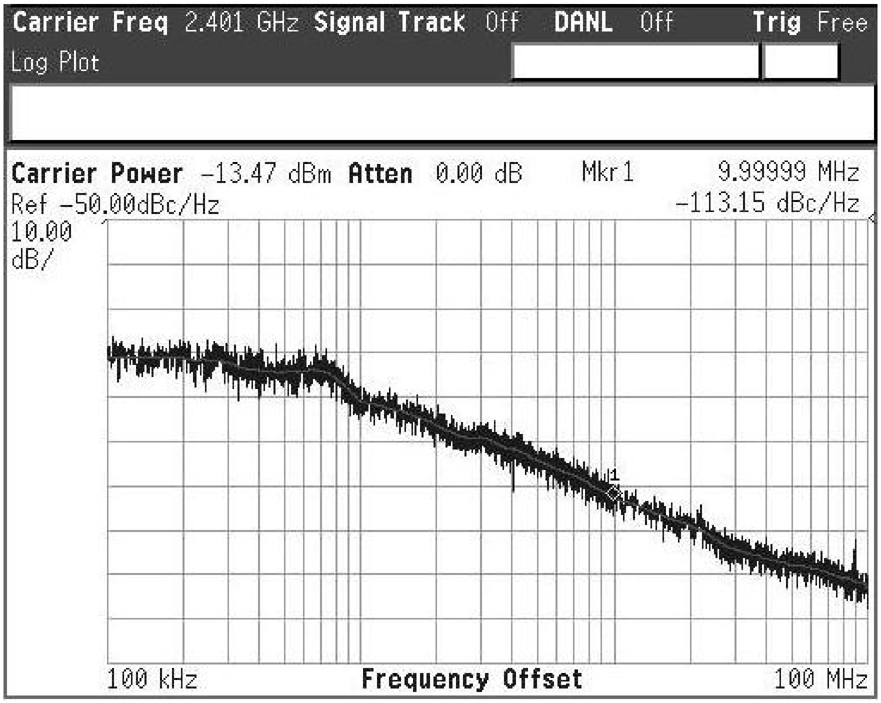
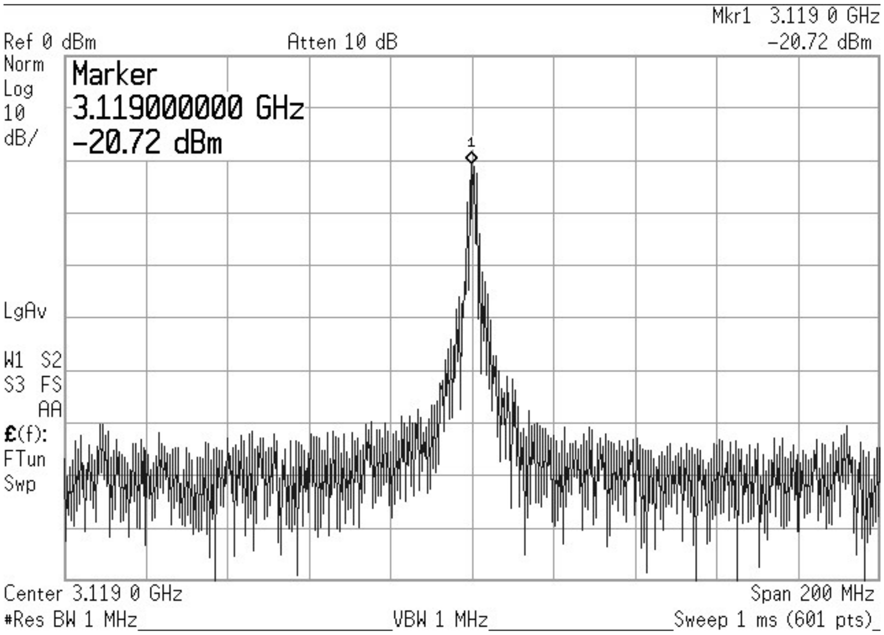


Fig. 53. Measured phase noise spectrum of OSC1-R at 2.4GHz operation

To compare the performance of both oscillators, first, we compare the performance at the same operating frequency of 2.4GHz. Note that to pull OSC1-R to this frequency, in addition to the tuning controls, we also decrease its supply voltage. Therefore, OSC1-R supply voltage is set to VDD=1.05V.

Fig. 50 and Fig. 51 show the output frequency spectrum and phase noise of OSC1 at 2.4GHz operation, respectively. It is seen that OSC1 has -113dBc/Hz phase noise at 10MHz offset at this frequency. It consumes 2.8mW of power. Fig.52 and Fig. 53 show the output frequency spectrum and phase noise of OSC1-R at 2.4GHz operation, respectively. It is seen that OSC1-R also has -113dBc/Hz at 10MHz offset at this frequency. However, OSC1-R consumes only 2mW of power.

After demonstrating that at the same operating frequency OSC1-R achieves the same phase noise with OSC1 for much lower power consumption, next test is to compare both oscillators at the same power consumption. For this, OSC1 is kept at 2.8mW power (2.4GHz frequency) and OSC1-R is pushed to 2.8mW power as well. It is seen that for this power consumption OSC1-R oscillates at 3.12GHz, delivering -112.8dBc/Hz phase noise at 10MHz offset. Fig. 54 and Fig. 55 demonstrate the frequency spectrum and phase noise spectrum of OSC1-R, respectively, at 3.12GHz operation consuming 2.8mW power.



Mkr1 3.119 0 GHz

Ref 0 dBm Atten 10 dB -20.72 dBm

Norm

Log

10

dB/

Marker

3.119000000 GHz

-20.72 dBm

LgAv

W1 S2

S3 FS

AA

£(f):

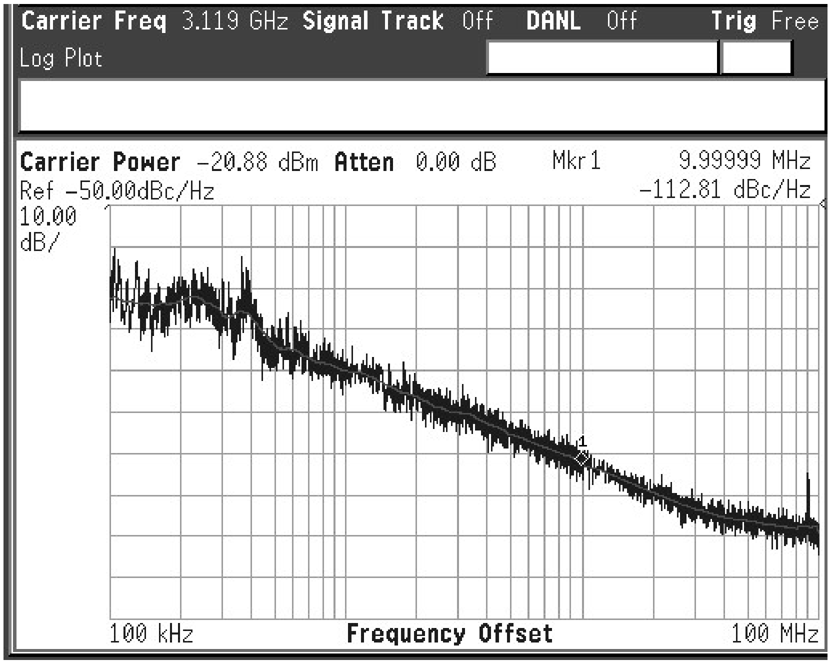
FTun

Swp

Center 3.119 0 GHz Span 200 MHz

#Res BW 1 MHz VBW 1 MHz Sweep 1 ms (601 pts)

Fig. 54. Output frequency spectrum of OSC1-R at 3.12GHz and 2.8mW power



Carrier Freq 3.119 GHz Signal Track Off DANL Off Trig Free

Log Plot

Carrier Power -20.88 dBm Atten 0.00 dB Mkr1 9.99999 MHz

Ref -50.00dBc/Hz -112.81 dBc/Hz

10.00

dB/

Fig. 55. Phase noise spectrum of OSC1-R at 3.12GHz and 2.8mW power

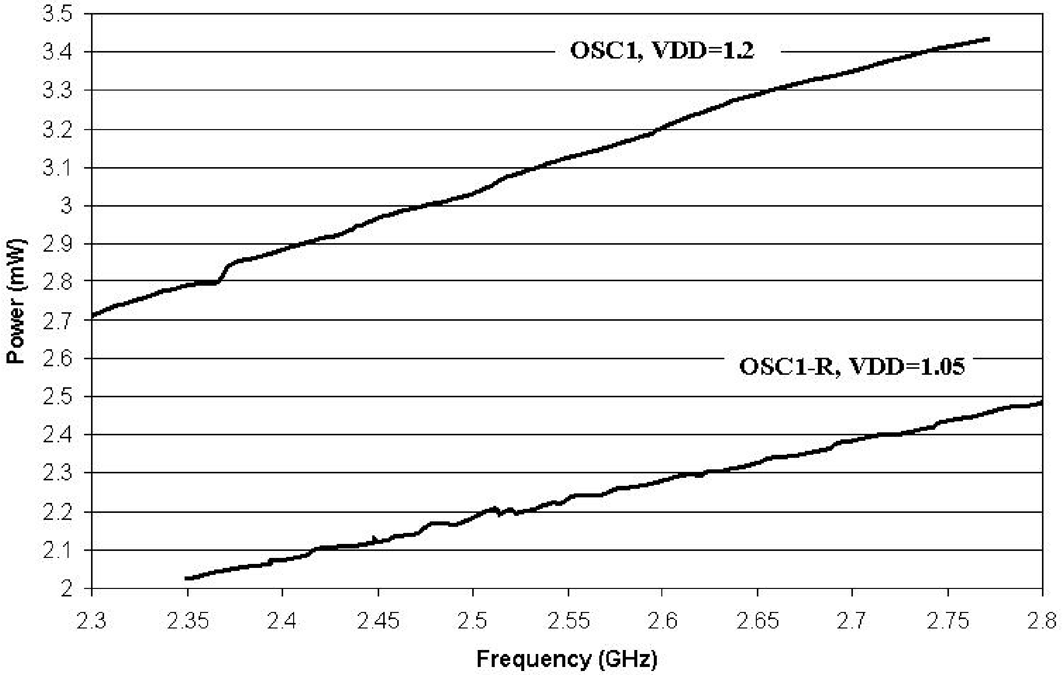


Fig. 56. Ring VCO measured power vs. frequency curves for OSC1 and OSC1-R

The power versus frequency plot shown in Fig. 56 is based on the measurement results of the two oscillators. The improvement in the speed / power trade off in the DCVSL-R oscillator, as seen in this plot, is significant. Table XIII summarizes the measured performance of both oscillators. Power consumption and areas are listed for core oscillators only, since open drain buffers are added for testing purposes. Note that the difference in the areas of the two oscillators show the area added by the resistors (including dummy resistors for matching).

The measurement results discussed in this section can be summarized as follows:

- At the same supply voltage (VDD=1.2V), OSC1-R oscillates at a 40% higher frequency range.
- At the same operating frequency (2.4GHz), OSC1-R consumes 30% less power

than OSC1 with the same phase noise performance as OSC1.

- For the same power consumption (2.8mW), OSC1-R oscillates 30% faster (3.12GHz) then OSC1 (2.4GHz).
- Speed vs. power trade off improves without sacrificing noise. The cost is added area. OSC1-R has 30% more area than OSC1.

Table XIII. Measured performance summary of OSC1 (based on Fig.47(a)) and OSC1-R (based on Fig.47(b))

Performance	OSC1	OSC1-R
Frequency Range	2.16GHz - 2.77GHz (VDD = 1.2V)	3.14GHz - 3.89GHz (VDD = 1.2V) 2.34GHz - 3.12GHz (VDD = 1.05V)
Power Consumption (2.4GHz operation)	2.8 mW	2 mW
Phase Noise (2.4GHz operation)	-113dBc/Hz at 10MHz offset	-113dBc/Hz at 10MHz offset
Area (mm^2)	54.2μm x 21μm	70.4μm x 21.3μm

Fig. 57 and Fig. 58 show the layout of the OSC1 and OSC1-R cores, respectively. As noted above, OSC1-R consumes more area than OSC1 due to the added resistors. However, the absolute values of the total area are quite small for both oscillators. Therefore, the addition in the area is not significant since the overall area consumption is very small. This will be discussed in the next section in the comparison of the proposed oscillator to other state of the art ring oscillators in literature. Fig. 59 shows the die micrograph for both oscillators.

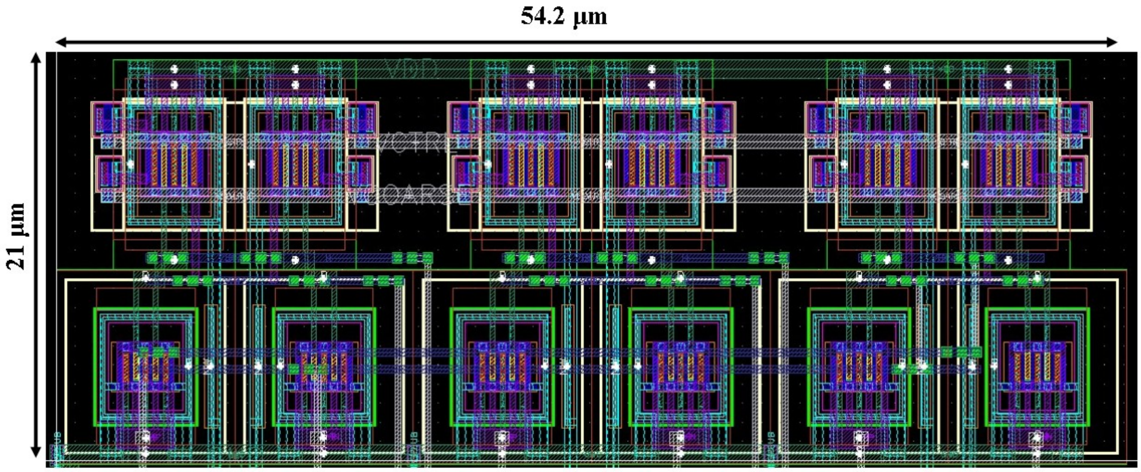


Fig. 57. Layout of OSC1 (based on DCVSL)

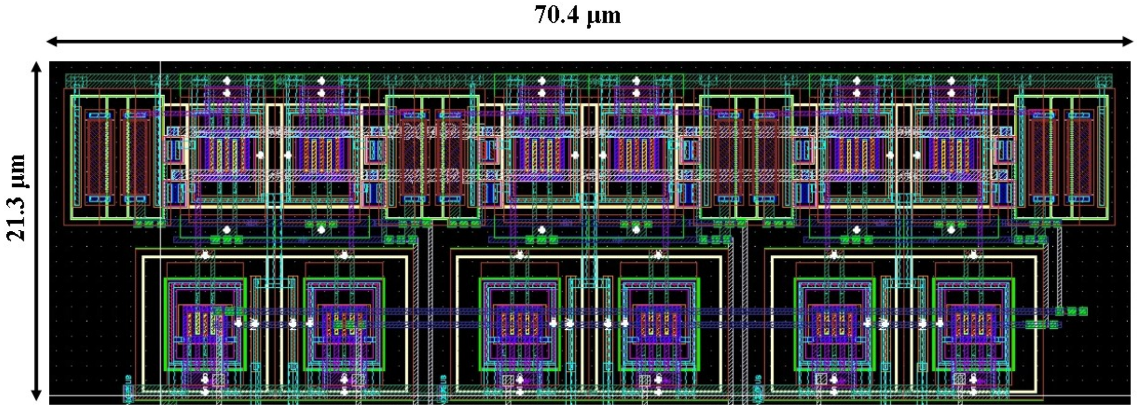


Fig. 58. Layout of OSC1-R (based on DCVSL-R)

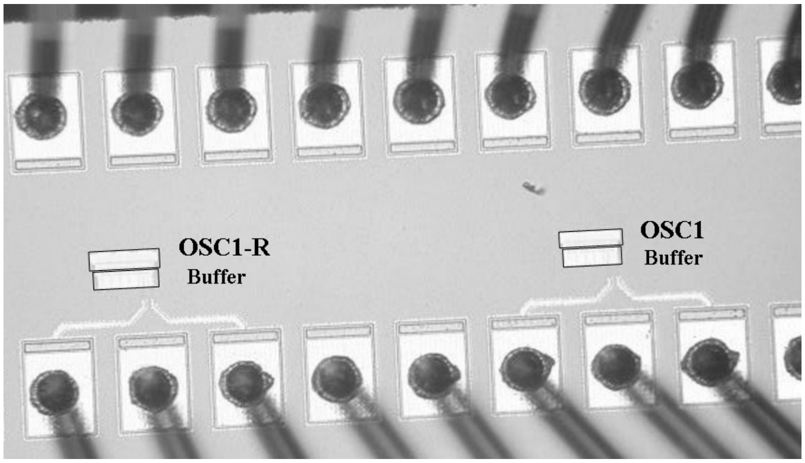


Fig. 59. Die micrograph of OSC1 and OSC1-R

C. Performance Evaluation

A Figure Of Merit (FOM) for oscillators [46] is shown in (5.1) where f_0 is the oscillation frequency and PN is the phase noise in dBc/Hz at an offset frequency of Δf. FOM is a useful performance metric that takes the power, speed and noise performances of the oscillator into account.

$$FOM\,(dBc/Hz) = PN + 10\log\left(P(mW) \times \frac{\Delta f^2}{f_0^2}\right) \tag{5.1}$$

It is demonstrated in [47] that for ring oscillators, the theoretical minimum achievable FOM is -165.2dBc/Hz ($7.33 \times kT$, where k is Boltzmann constant and T is temperature).

Table XIV provides a comparison of the proposed OSC1-R with state of the art ring-oscillator-based VCOs operating at similar frequencies. It is seen that this work demonstrates a competitive FOM of -157.6dBc/Hz when compared to the state of the art oscillators.

Note that FOM does not take area into consideration. While [49] reports an

Table XIV. Performance comparison of OSC1-R with previously reported solutions

	[48]	[49]	[50]	[51]	**This Work (OSC1-R)**
Architecture	3 stage ring	RC - BPF	2 stage ring	2 stage ring	3 stage ring
Technology	0.35 μm	0.13 μm	0.18 μm	0.28 μm	0.13 μm
Frequency (GHz)	2.4	2.5	2	2.45	2.4
Power Consumption	15 mW	2.86 mW	0.7 mW	19.2 mW	2 mW
Phase Noise (dBc/Hz)	-97 at 1MHz	-95.4 at 1MHz	-90 at 1MHz	-96 at 1MHz	-93 at 1MHz
FOM (dBc/Hz)	-153	-159	-157	-151	-157.6

FOM of -159dBc/Hz, the oscillator area is $0.006mm^2$, four times that of the proposed OSC1-R oscillator. This also shows that while DCVSL based oscillator consumes less area than the one that is based on DCVSL-R cells, the overall area of OSC1-R is still very small. Therefore, this work demonstrates a good FOM and a low cost solution that consumes only $0.0015mm^2$ of area.

CHAPTER VI

ALL DIGITAL PHASE LOCKED LOOPS

A. Background and Motivation

The microchips that are employed in microprocessor and serial link applications are very digital intense, helping them benefit from technology scaling and the faster speeds of sub-micron technologies. Digital circuits are also easily controlled and calibrated via the DSP processor that is readily available in all such systems. However, most of today's microprocessor and serial link clock generators are based on analog charge-pump based PLLs.

With the migration towards sub-micron technologies, the design of high performance analog circuits became increasingly challenging. One such design challenge is the reduced voltage headroom which degrades SNR and in a PLL diminishes charge pump output impedance and VCO dynamic range. Smaller feature sizes also increase the impact of channel length modulation, leading to higher current mismatch and spurs. Moreover, the analog intense PLL features large capacitors as well as other non-scalable elements such as resistors and special RF process components as inductors and varactors. These components are not part of standard digital CMOS processes and require extra characterization diminishing yield and increasing cost.

All digital phase locked loops (ADPLL) were implemented to generate clock frequencies in several hundred MHz range in the past [52–54]. While these works created grounds for today's ADPLL architectures by proposing digitally controlled multi-mode loop architectures and oscillators [52], enable/disable inverter-cell-based matrix ring oscillators [53], the lack of good timing resolution prevented them from being utilized in high performance systems. While old technologies' coarse timing resolution and supply voltages exceeding 2.5V previously favored analog PLLs, modern CMOS technologies' picosecond gate-delay capabilities and 1-V supplies made high-resolution ADPLLs very attractive. This PLL design paradigm shift has motivated recent work in digital/hybrid PLL architectures [55–68].

B. DPLL Basics

Fig. 60 shows the main components of a conventional DPLL [69]. The time difference between the reference and divider output signals are converted into a digital word by a phase frequency to digital converter which often involves a high resolution multi-bit Time to Digital Converter (TDC) [55–60, 67]. This word is processed by an all digital loop filter (DLF) and is fed to a digitally controlled oscillator (DCO). The delta-sigma modulator (DSM) dithers DCO control bits to improve the finite resolution of the digital tuning word and reduce output jitter arising from DCO-control-word quantization noise.

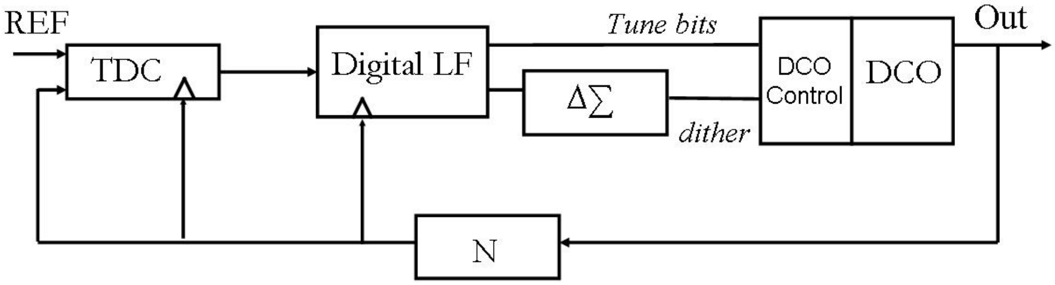


Fig. 60. Block diagram of a conventional DPLL

Delay line based structures [55], [57], [70], [71] and a gated ring oscillator (GRO) based structure [72] are among the popular TDC implementations while the DPLLs in [62], [65], [68] employ bang-bang phase-frequency detectors. To understand the basics of a the operation of a DPLL through a linear loop analysis, let's assume a TDC-based architecture as shown in Fig. 60.

In phase domain, the TDC can be viewed as an analog-to-digital converter with input phase difference $\Delta\phi_{in}$ with a digital output word W_{TDC}. Then, the transfer function of the TDC is given by:

$$H_{TDC} = \frac{W_{TDC}}{\Delta\phi_{in}} = \frac{Tref}{2\pi \times t_{res}} \tag{6.1}$$

where t_{res} is the time resolution of the TDC, Tref is the reference period and corresponds to the maximum phase difference of 2π.

The digital loop filter is often implemented as a proportional integral filter that corresponds to the first order passive low pass filter commonly employed in analog charge pump PLLs. Note that analog PLLs employ a second order loop filter as discussed in Chapter II and shown in Fig. 3 where a second capacitor is added to minimize control voltage ripples and the effect of the added pole is often ignored in loop analysis due to its placement. However, the analog voltage ripple is not a concern in a digital implementation , therefore, the first order proportional-integral loop filter is used. The digital loop filter and a DCO control interface corresponding to that filter is shown in Fig. 61.

The z-domain transfer function of a proportional integral filter is given below

$$H_{LF}(z) = \alpha \times \left(\frac{z - \left(1 - \frac{\beta}{\alpha}\right)}{z - 1} \right) \tag{6.2}$$

where α and β are proportional and integral path coefficients, respectively. Note that continuous time approximation of analog PLLs is widely studied in literature [21], [23]. Therefore, it is beneficial to derive the loop equations of the DPLL in the familiar

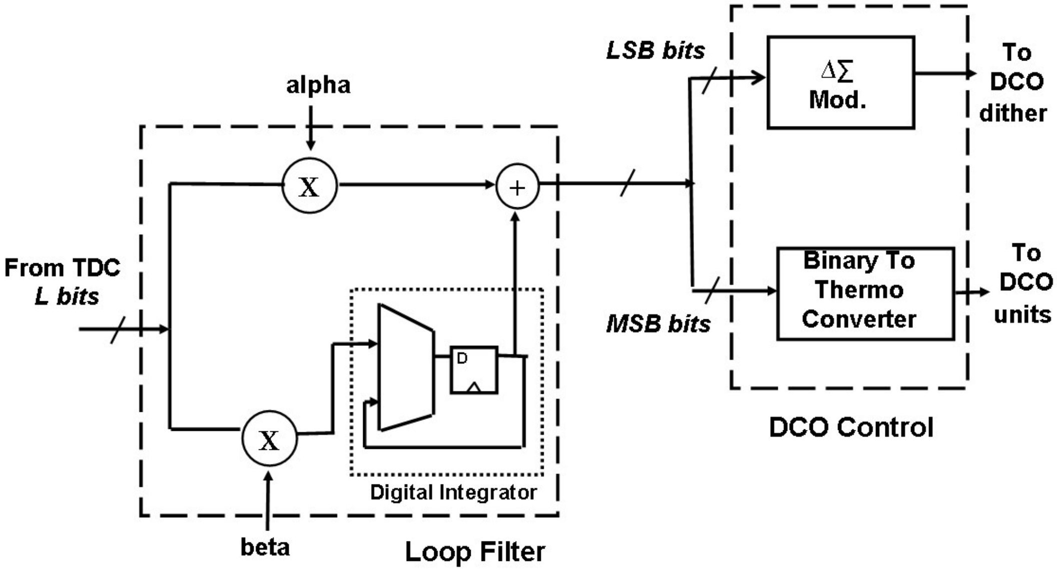


Fig. 61. A conventional proportional integral digital loop filter and DCO control interface

s-domain to aid the design process. We can use bilinear transform [73] as shown below:

$$z = \frac{2Fs + s}{2Fs - s} \tag{6.3}$$

where Fs is the sampling frequency of the discrete-time system.

Note that bilinear transform is accurate for operating frequencies that are much smaller than the Nyquist rate of the system. In the DPLL, the reference frequency is commonly used as the loop sampling frequency. Moreover, the frequencies of interest in the DPLL are much less than the reference frequency. Therefore, bilinear transform can be used to analyze the loop behavior for frequencies that are smaller than the loop sampling frequency. Further information on the z-domain analysis of discrete-time PLLs can be found in [24], [74].

$$Fs = Fref \tag{6.4}$$

and

$$Fref = 1/Tref \tag{6.5}$$

Then, using (6.2) and (6.3), the loop filter transfer function is given by (6.6).

$$H_{LF}(s) = (\alpha - \beta/2) \times \left(\frac{s + \dfrac{\beta}{(\alpha - \beta/2)} Fref}{s} \right) \tag{6.6}$$

Then, the loop has a zero placed at the digital loop filter zero w_z.

$$w_z = \frac{\beta}{(\alpha - \beta/2)} Fref \tag{6.7}$$

In most practical cases α is much larger than β. Therefore the loop filter equations can be approximated as:

$$H_{LF}(s) \approx \alpha \times \left(\frac{s + \frac{\beta}{\alpha} Fref}{s} \right) \tag{6.8}$$

and

$$w_z \approx \frac{\beta}{\alpha} Fref \tag{6.9}$$

The DCO contributes integration (from frequency to phase) to the loop. The digital bits that go through the delta-sigma modulator (DSM) effect the output frequency as fractional bits. Therefore, at the loop filter output, if F least significant bits of the loop filter output word W_{LF} are fed to the DSM, then the transfer function of the loop shown in Fig. 60 from the loop filter output to the DCO output is:

$$\frac{\phi_{out}}{W_{LF}} = \frac{1}{2^F} \frac{K_{DCO}}{s} \tag{6.10}$$

where K_{DCO} is the DCO phase gain in $radians/(seconds \times LSB)$ and can be expressed as

$$K_{DCO} = 2\pi \times f_{res} \tag{6.11}$$

where f_{res} is the DCO frequency resolution in Hz/LSB. From the expressions we derived in (6.1) to (6.10) we conclude that the continuous approximation closed loop phase transfer function of the DPLL is given by:

$$H_{CL\_DPLL}(s) = \frac{\phi_{out}}{\Delta\phi_{in}} = \frac{(K_{DLOOP} \times N)(s + w_z)}{s^2 + K_{DLOOP}s + K_{DLOOP}w_z} \tag{6.12}$$

where N is the feedback divider ratio and the digital loop gain factor K_{DLOOP} is given by:

$$K_{DLOOP} = \left(\frac{Tref}{2\pi \times t_{res}}\right)\left(\frac{\alpha 2\pi f_{res}}{N2^F}\right) \tag{6.13}$$

The closed loop transfer function of the DPLL given in (6.12) is in the same form as the transfer function (2.10) derived in Chapter II Section C. Therefore, the closed loop bandwidth, damping factor and natural frequency of the DPLL can also be determined in a similar fashion as Table IV where loop gain factor K_{LOOP} should be replaced by K_{DLOOP}. The relations between the loop parameters given in Table V also apply to the DPLL.

The DPLL loop design parameters introduced in this section are summarized in Table XV and the second order continuous approximation loop parameters are listed in Table XVI where GBW is the open loop gain bandwidth product. A direct analogy between the design parameters of an analog PLL and that of a DPLL is discussed in [69].

In addition to the expressions for loop parameters that are presented in this section, another useful relation is the one to convert a time increment Δt in the

Table XV. List of DPLL loop parameters

Loop Parameter	**Explanation**
Fref	Loop sampling frequency (Hz) (assumed equal to reference frequency)
t_{res}	TDC time resolution (seconds)
α	Proportional path gain
β	Integral path gain
f_{res}	DCO frequency resolution (Hz/LSB)
F	Number of fractional bits (connected to DSM)
N	Feedback division ratio

period of a signal into the corresponding frequency decrease Δf in its frequency or vice versa. If F_0 and T_0 are the frequency and period of the signal on which the time increment occurs, then the resulting frequency decrease is derived as follows [70].

$$\begin{aligned} T_1 &= T_0 + \Delta t \\ \Delta f &= F_0 - F_1 \\ \Delta f &= \frac{1}{T_0} - \frac{1}{T_0 + \Delta t} \\ \Delta f &= \frac{\Delta t}{T_0^2 + T_0 \Delta t} \end{aligned} \tag{6.14}$$

Then, for small time increments where $\Delta t \ll T_0$

$$\Delta f \approx \frac{\Delta t}{T_0^2} \tag{6.15}$$

Table XVI. Summary of the DPLL important second order loop expressions

Control Parameter	**Expression**
Natural Frequency	$w_n = \sqrt{K_{DLOOP} w_z} = \sqrt{\frac{f_{res}\beta}{t_{res}N2^F}}$
Loop Zero	$w_z = \frac{\beta}{\alpha} Fref$
Damping Factor	$\xi = \frac{1}{2}\sqrt{\frac{K_{DLOOP}}{w_z}} = \frac{1}{2}\frac{\alpha}{Fref}\sqrt{\frac{f_{res}}{\beta t_{res}N2^F}}$
Loop Bandwidth	$w_c \approx GBW = K_{DLOOP} = \frac{f_{res}\alpha Tref}{t_{res}N2^F}$

Similarly, the period decrease that results from a small frequency increment is:

$$\Delta t \approx \frac{\Delta f}{F_0^2} \tag{6.16}$$

The expressions of (6.15) and 6.16) are very useful in relating the effect of frequency increase/decrease at the DCO output to a change in signal period, especially because TDC deals with time domain rather than the frequency domain. For instance, for a DPLL with a division factor of 16 and a DCO frequency of 2GHz, a 10MHz frequency change at the DCO output results in 40ps time difference at the TDC's divider input period.

C. Noise in ADPLLs

In addition to the noise sources in a PLL (noise from the building blocks, supply noise, etc.), a DPLL has additional noise sources due to its digital nature. The time-difference of the two signals at the TDC input is converted into a digital word. One significant noise source in a DPLL is quantization noise due to the finite resolution of the TDC.

Note that quantization noise has uniform distribution. And the time resolution t_{res} of the TDC corresponds to a phase resolution of

$$\phi_{res} = \frac{2\pi t_{res}}{Tref} \tag{6.17}$$

Then, the phase noise of the TDC at the input of the PLL in dBc/Hz is:

$$\pounds\_TDC = 10\log\left(\frac{t_{res}^2(2\pi)^2}{12Tref}\right) \tag{6.18}$$

Noise coming from the input of the PLL is low-pass filtered [18] through the closed-loop transfer function given in (6.12). To find the accurate representation of the noise contribution of the TDC at the DPLL output, the phase noise given in (6.18) should be passed through the low-pass filter transfer function (6.12).

Note that the in-band noise due to the TDC, at the output of the PLL is given by (6.19) since phase at the TDC input is multiplied by the division factor when related to phase at the PLL output

$$\pounds\_TDC_{PLL\_inband} = 10\log\left(\frac{N^2 t_{res}^2(2\pi)^2}{12Tref}\right) \tag{6.19}$$

Other noise sources in the DPLL are due to the finite frequency resolution of the DCO and due to the DSM dithering of the DCO bits. A detailed analysis on the phase noise contribution of the DCO quantization and dithering noise is given in [70]. When analyzing the loop's effect on noise sources, the loop transfer function from the point where the additive noise is applied to the PLL output should be determined.

For the DCO (or the VCO in a charge-pump based PLL), a common confusion is in determining if the DCO noise should be placed as an additive noise source at the output of the DCO or at the input of the DCO. The loop transfer function for both cases are listed below:

$$\frac{\phi_{out}}{\phi_{DCOout}} = \frac{1}{1 + K_{DLOOP}\dfrac{(s + w_z)}{s^2}} = \frac{s^2}{s^2 + K_{DLOOP}s + K_{DLOOP}w_z} \tag{6.20}$$

$$\frac{\phi_{out}}{W_{CTRL}} = \frac{K_{DCO}/s}{1 + K_{DLOOP}\dfrac{(s + w_z)}{s^2}} = \frac{K_{DCO}s}{s^2 + K_{DLOOP}s + K_{DLOOP}w_z} \tag{6.21}$$

where W_{CTRL} is the DCO control word (would be V_{CTRL} for a VCO), ϕ_{DCOout} is the additive phase noise added at the DCO output, ϕ_{out} is the DPLL output phase, the noise of which is analyzed. Note that the loop acts as a high-pass filter to noise sources at the DCO output while it acts as a band-pass filter to the noise sources at the input of the DCO.

When the phase noise of an oscillator is concerned, it is expressed as the phase noise of the oscillator output signal. Therefore, it is customary to represent the oscillator noise source at its output and apply the loop transfer function of (6.20) to determine the effect of the loop on the output noise.

The phase noise due to DCO quantization and DSM dithering, at the DCO output are [70]:

$$\pounds\_DCO_Q(\Delta f) = 10\log\left(\frac{1}{12}\left(\frac{f'_{res}}{\Delta f}\right)^2\frac{1}{Fref}\left(\text{sinc}\frac{\Delta f}{Fref}\right)^2\right) \tag{6.22}$$

$$\pounds\_DCO_D(\Delta f) = 10\log\left(\frac{1}{12}\left(\frac{f'_{res}}{\Delta f}\right)^2\frac{1}{Fdith}\left(2\text{sin}\frac{\pi\Delta f}{Fdith}\right)^{2n}\right) \tag{6.23}$$

where $\pounds\_DCO_Q(\Delta f)$ and $\pounds\_DCO_D(\Delta f)$ are the phase noise of the DCO in (dBc/Hz) due to quantization (finite resolution) and DSM dithering, respectively, at an offset frequency of Δf. Note that f'_{res} is the DCO resolution with the DSM dithering taken into account, $Fdith$ is the DSM dithering frequency and n is the DSM order.

The resolution of the DCO, f_{res}, becomes f'_{res} due to the dithering. As shown in (6.10), the overall DCO and DSM gain is scaled due to the fractional dithering bits. This is equivalent to the DCO resolution being scaled. Therefore, assuming that the ratio of the dithering speed to the reference speed is enough [70] we have:

$$f'_{res} = \frac{f_{res}}{2^F} \tag{6.24}$$

where f_{res} is the DCO frequency resolution in (Hz/LSB) and F is the number of fractional bits.

Note that there are two noise components, one for regular quantization noise and one for dithering, because often, only F least significant bits of the DCO tuning word are connected to the DSM and dithered. For the dithered bits, their update frequency is $Fdith$ and they are subject to delta-sigma noise shaping which results in the noise expression in (6.23).

The remaining most significant bits of the DCO control word directly control the DCO, without dithering. Their update frequency is the loop update frequency $Fref$. The *sinc* function in the $\pounds\_DCO_Q$ noise term is due to the DCO control word being updated only at every $Fref$ and being held constant between the updates, similar to a zero-order hold [70]. Note that the resolution of the quantization noise in both noise components is f'_{res}. This resolution is employed in $\pounds\_DCO_Q$ too because removing the F least significant bits of the control word is equivalent to scaling it by 2^F.

If DSM dithering was not used, the DCO would have only the noise contribution of $\pounds\_DCO_Q$ due to its finite quantization, and would have a resolution of f_{res} where the choice of f_{res} also significantly effects the output tuning range. For instance, to cover a 10GHz range, a 5MHz DCO resolution would require 2000 units for a unit weighted structure. For a reference frequency of 100MHz, the phase noise at the DCO output due to its quantization, with a 5MHz resolution would be $\pounds\_DCO_Q =$

$-97dBc/Hz$ at 10MHz offset frequency. When the phase noise performances of state-of-the art ring oscillators listed in Table XIV of Chapter V are considered, it is seen that this noise is unacceptable and will be higher than the DCO's natural phase noise.

When a DSM is used to dither the DCO control bits, the improved resolution enhances the noise. For a first order DSM with dithering speed of 1GHz, and 5 fractional bits, the noise of the dithering bits at 10MHz offset frequency is $\pounds\_DCO_D = -161dBc/Hz$.Note that the remaining most significant bits of the DCO control word will still contribute quantization noise, $\pounds\_DCO_Q = -127dBc/Hz$ at 10MHz offset.

The phase noise expressions of (6.22) and (6.23) are the open-loop expressions at the DCO output, in other words, the effect of the feedback loop is not taken into account. As seen in (6.20), the loop acts as a high-pass filter to the noise at the DCO output. Then, for offset frequencies Δf outside of the loop bandwidth, the feedback loop will be ineffective and the DPLL output phase noise due to DCO quantization and DSM dithering will be equivalent to (6.22) and (6.23). For offset frequencies that are within the ADPLL loop bandwidth, the noise expressions of (6.22) and (6.23) should be passed through the loop transfer function of (6.20).

D. Design Challenges

The DPLL embodies several design challenges and trade-offs, as summarized below.

1. Stability vs. Complexity

As seen from the loop zero expression (6.9), the loop zero placement is proportional to the ratio of integral coefficient to the proportional coefficient. Therefore, from stability perspective, a large α/β is desired. This implies that as shown in Fig. 61, the TDC output, passing from the loop filter, will grow in the number of bits substantially, making the digital circuitry of the loop filter and the following DCO control interface bigger and more costly.

2. Noise vs. Complexity

Unlike an analog PLL where the time difference between the reference and divider frequencies is converted into a continuous voltage value, in a digital PLL the time difference is quantized and represented with a binary word in the TDC. As discussed in Section C, the finite quantization of the TDC adds jitter to the PLL output tone and the added phase noise is proportional to the inverse of the TDC time resolution, requiring a high resolution TDC to improve the PLL output noise. However, the increased number of bits generated by the TDC block must be processed by the digital loop filter and DCO control logic, significantly increasing the complexity of the proposed architectures.

A similar trade off exists in the DCO as well. The minimum frequency step of the DCO results in quantization noise as well. Note that the addition of sigma-delta modulation increases the DCO resolution and helps reduce the phase noise contribution of the DCO quantization.

3. DCO Control implementation vs. Complexity

Since the control word is a binary number, the DCO should either consist of binary weighted units, or of unit weighted blocks. In a high resolution and stable DPLL, the TDC output after passing through the filter will be a large digital word. Due to the high ratio of the most significant and least significant bits of this digital word, operation of binary weighted units suffer from mismatches, is not always monotonous, and the settling time of switching each binary bit is different, resulting in inconsistencies. However, while a unit weighted DCO will solve these problems, it will require a binary to thermometer (B-T) converter between the loop filter and the DCO, the design of which involve concerns such as erroneous decisions caused by bubbles in the converter. Moreover, the number of units in the DCO are proportional to the desired tuning range and in a wide tuning range setting, B-T converter complexity increases considerably.

4. Wide Tuning Range

In this dissertation, we target a wide range of operation for our DPLL to be able to cover multiple serial links and provide a multi-standard compatibility. However, a wide operation range trades with several DPLL design challenges such as complexity, power consumption and stability.

To achieve a wide range of operation, the DCO should have a very wide tuning range. In an LC tank architecture this requires large varactor banks and possibly the addition of multiple inductors. In a ring oscillator, wide tuning range is more easily obtained but it increases the power consumption of the oscillator significantly. Also note that due to quantization noise concerns, the DCO resolution f_{res} should be kept small. For a DCO that consist of equally weighted units with constant and low frequency gain f_{res}, the tuning range will be proportional to the number of DCO units, trading tuning range with complexity.

To adjust the PLL output frequency that operates from a constant reference frequency, the division ratio N is varied, as discussed in Chapter II Section C. Note that the division ratio N affects the loop bandwidth and the loop damping factor as shown in Table XVI. Then, to achieve a wide range of DPLL output frequencies, the stability of the loop should be secured for all of the values of N.

The DPLL proposed in this dissertation addresses wide tuning range while maintaining stability and minimizing complexity as discussed in Chapter VII.

CHAPTER VII

A WIDE RANGE ALL DIGITAL PLL

One of the most important benefits of an all digital PLL is its programmability and flexibility due to all digital controls. In addition, if the implementation technology is a standard digital CMOS technology, the DPLL will be a low-cost solution. The idea of a highly programmable, flexible, digitally controlled DPLL, that is manufactured in a low-cost standard digital CMOS process motivates the design of a multi-standard ADPLL, which can be programmed to implement clock signal for various protocols. Therefore, instead of employing various custom designed PLLs for different wireline protocols, a single, programmable, all digital solution can be used as a multi-purpose unit. However, the supported data rate for serial link protocols vary in a very wide range (PCI express supports 2.5Gbps and 5Gbps, SONNET supports 2.488Gbps and 9.95Gbps [75]). Therefore, a multi-protocol ADPLL should be capable of wide range of operation.

A. Previous Work

While numerous DPLL architectures have been proposed to support narrow range wireless applications [55–62], these synthesizers generally employ highly tunable varactors and inductors that consume metal resources and introduce significant process complexity. Moreover, their limited frequency range (smaller than 1GHz) is not suitable to support a wide range of serial link standards.

[64] offers a wide range operation (24GHz-32GHz) in 65nm CMOS technology but employs an LC tank based VCO which requires costly mixed signal process in fabrication due to the use of inductors and varactors. [65, 66] offer a wide range operation and do not employ R/L/C components or DAC and B-T converters.

However, [65, 66] are implemented on special and expensive SOI process and the loop architecture features a single-bit shifter to control the integer DCO frequency which limits the loop's ability to move the DCO frequency only one unit per update cycle. [65, 66] also feature a bang-bang PFD (BBPFD) which simplifies the phase-frequency detection and the loop filter but brings nonlinear loop dynamics and lack the high resolution of a multi-bit TDC. Moreover, BBPFDs have uncontrolled loop bandwidth, and limited frequency pull-in range [68]. Other BBPFD works [68] propose special frequency-locking circuitry, creating a dual path architecture that reduces TDC complexity by appreciably complicating the remainder of the loop.

Recently, high resolution TDCs that minimize quantization noise became key building blocks of DPLLs [55–60, 67]. High resolution implies large number of data bits to be processed in the loop, increasing the size and complexity of the loop digital circuits such as the loop-filter and the DCO/loop-filter interface (B-T converters [55, 57, 58, 62]). Moreover, recent DPLLs employ digital-to-analog converters (DACs) followed by a VCO [67, 68], bringing analog design constraints and/or non-scalable elements such as resistors into the picture.

For instance, [55] employs a binary-to-thermometer converter at the DCO control as well as three loop modes (coarse, tracking, fine) in the system, each of which requires separate circuitry, increasing cost/complexity. [56], [68] utilize dual-path architectures that require two digital loop filters, two DACs and a circuit that switches between the two paths, increasing area, power and complexity. Moreover, building blocks such as DACs and binary-to-thermometer converters employ analog components (e.g. amps/resistors), defeating the purpose of scalable ADPLLs.

In this work, an all digital PLL that targets a wide operation range to serve as a multi-protocol programmable ADPLL is presented. The proposed ADPLL system addresses the challenges of wide range operation such as stability and large frequency error range. It accommodates a multi-bit linear TDC, does not use non-scalable R/L/C components and avoids DACs or B-T converters commonly employed in DPLLs. The digital building block complexity is minimized by processing only the least significant bits (LSB) of the TDC output in the loop-filter, decreasing the adder/subtractor sizes. The remaining most significant bits (MSB) are directly connected to the proposed row/column matrix shifter (smart shifter) that controls the DCO. The multi-bit shifter facilitates faster frequency tuning per loop cycle for the wide-range ADPLL.

B. Proposed System Design

The proposed loop is completely digital, does not employ any nonscalable elements such as R/L/C and it eliminates the need for DACs , removing analog design concerns altogether from the design. It features a coarse path that inherently enables/disables itself through frequency lock, without requiring explicit additional lock detection or a dual loop architecture. It employs only 5-bit binary weighted DCO controls and the rest of the DCO controls are unit-weighted, without the need for a thermometer converter. Therefore the proposed loop, the details of which are described below, minimizes design complexity while maintaining wide range of operation.

Fig. 62 shows the proposed loop structure employed in this work and Table XVII summarizes the design parameters used in this prototype. A ring oscillator is used to provide the wide output tuning range and to avoid the use of special RF process components such as varactors/inductors. A high loop bandwidth is desired to balance the noise contributions of the TDC and the ring oscillator to the PLL output. A variable loop gain element, A, modifies the loop bandwidth through the various division ratios to maintain stability over wide range of operation. External controls of the frequency dividers determine the division ratio N where the values of N are

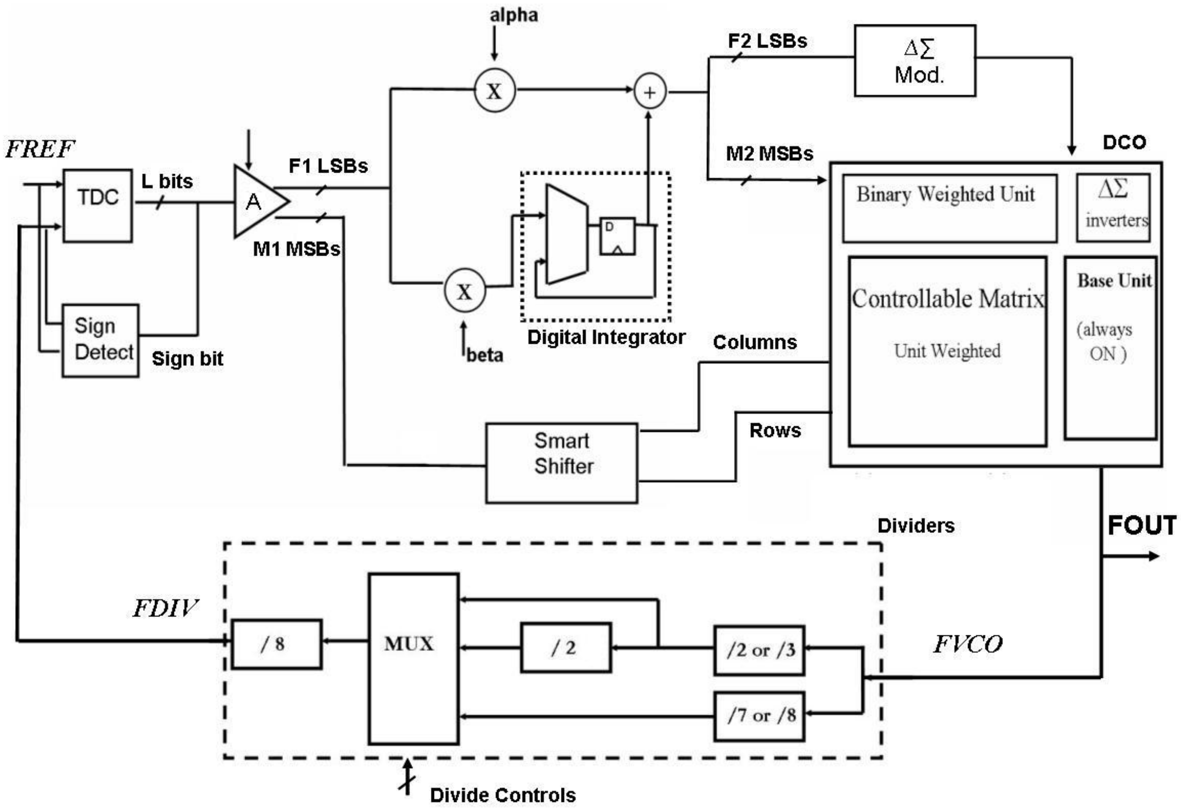


Fig. 62. Block diagram of the proposed all digital PLL

given by (7.1).

$$N = 8 \times (2, 3, 4, 6, 7, 8) \tag{7.1}$$

To avoid the complexity vs. performance trade off, only F1 least significant bits of TDC output are processed in the first order, digital, proportional integral loop filter, making this a fine control path. The output word of the loop filter, W_{FINE}, is further separated into its F2 least significant bits, W_{FINE_F}, that serve as fractional dithering bits through the DSM resulting in a fine resolution that is a fraction of the DCO gain KDCO (Hz/LSB).

Note that the output of the loop filter could all be fed to the DSM depending on

Table XVII. Summary of the loop parameters of the implemented ADPLL

Loop Parameter	Value
L	9 bits
F1	6 bits
M1	5 bits
A	1, 2 or 4
α/β	2^4
F2	5 bits
M2	5 bits

the desired bandwidth, since this would create a very low gain (hence low bandwidth) path. In the proposed prototype, a high bandwidth requires some of these bits to be processed as integer bits. Therefore, remaining M2 most significant bits of the filter output serve as integer control bits, W_{FINE_I} and are fed directly to binary weighted inverter units in the oscillator, eliminating the binary-to-thermometer converter.

Since the TDC output and the loop filter output are separated into their LSB/MSB components, the number of binary weighted bits can remain small, in this case only $M2 = 5$ bits, resulting in only a 16-to-1 weight ratio between the weighted units. Overall, the loop filter holds the fine control word given as:

$$W_{FINE} = W_{FINE\_I} + 2^{-F2} \times W_{FINE\_F} \tag{7.2}$$

The most significant bits of the TDC output serve as a coarse frequency tuning path. If the total coarse control word, W_{COARSE} was held in the loop, similar to the loop filter output holding W_{FINE}, then the DCO frequency would be:

$$FOUT = F_0 + KDCO \times (W_{FINE} + W_{COARSE}) \tag{7.3}$$

where F_0 is the DCO free running frequency. However, to maintain a very wide output frequency range through the use of a unit-based DCO with a constant KDCO, W_{COARSE} should be a large number. To minimize complexity, instead of holding the explicit control word W_{COARSE} with a large accumulator in the coarse path and using a binary-thermometer converter interface to control the DCO, the inherent memory in the DCO can be exploited, causing it to double as an accumulator [65]. Therefore, instead of indicating the absolute number of inverter cells that are active, W_{COARSE}, we provide the change in number of cells, Δx that should be activated, to the DCO. Then, at loop cycle n:

$$W_{COARSE}(n) = W_{COARSE}(n-1) + \Delta x(n) \tag{7.4}$$

Note that W_{COARSE} is no longer explicitly held in the loop, but is represented through the total inverters that are on in the variable matrix of the DCO. This strategy not only obviates a very large accumulator to hold the total DCO control word, but also eliminates the large binary-to-thermometer converter at the DCO control input. Note that while a simple single bit shifter can move the DCO [65], in the proposed prototype, 5 MSBs of the TDC output are fed to a proposed smart shifter and the DCO frequency can move as much as 31 units per cycle. The challenges of implementing the multi-bit smart shifter will be discussed in more detailed in Section III.

The MSBs of the TDC are on a path with two implicit integrations, one for the DCO acting as an accumulator to perform (7.4), the second is due to the DCO converting frequency to phase, in linear analysis. Therefore, the coarse path is an unstable path. However, the coarse bits will place the DCO in the vicinity of the target lock frequency during the non-linear frequency lock and then the LSBs of the TDC and the fine path will dominate, placing the loop to the phase lock. To achieve the wide continuous operation range, it is important to be able to move the DCO from one end of the operation range (7GHz in this prototype) to the other (2GHz) in a quick coarse path. Since the coarse and fine paths are the most and least significant bits of the TDC output, no explicit lock detection or dual loop enable disable mechanism is needed.

1. Phase Frequency Detection

Fig. 63 shows the details of phase frequency detection and its conversion to a digital word. It is common practice to use the TDC for fractional phase error detection and employ a coarse counter or frequency detector for frequency detection [55, 57–60, 67]. To utilize the same TDC core for both phase and frequency detection, we use a PFD block to generate the enable signal of the TDC core. The TDC core is based on a multi-path gated ring oscillator (GRO) structure [72] which counts the time-width of its enable signal (with a resolution of 20ps in this prototype).

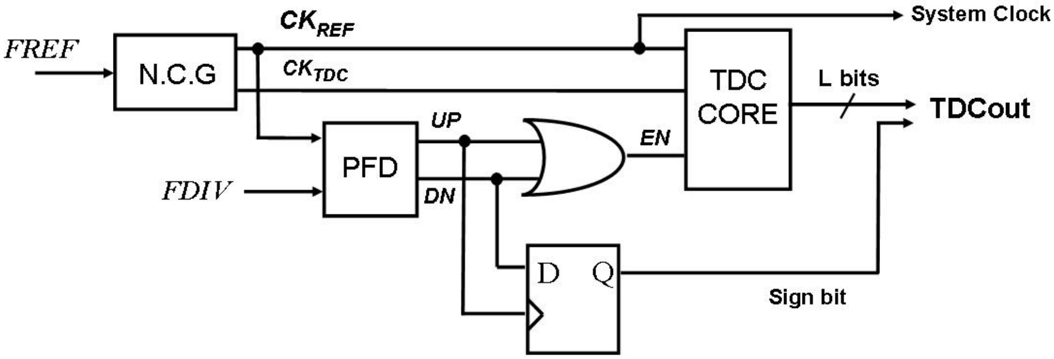


Fig. 63. System level diagram of the time to digital converter

An internal delayed version of reference signal, CK_{REF} is used as the reference clock and a non-overlapping version of this signal, CK_{TDC} is generated to be used internally by the TDC processing circuitry. The OR operation on the UP and DN signals remove the lead/lag information on CK_{REF} and CK_{DIV}, therefore we deduct the sign bit separately through an early/late detection flop.

Within the TDC core, a GRO will count the enable signal while processing circuitry will process this count to generate a proper TDC output word. Once sampled, the count value can be reset, or similar to a reset, the previous count can be subtracted from the current one [72]. Since resetting the count might delay the circuit before a new count can begin, we perform the latter. Note that CK_{REF} is used as the system clock by the digital circuitry in the loop. This ensures that the timing constraints of the digital circuitry in the loop is determined by the reference clock speed and is independent of the DCO frequency which varies in a very wide range.

C. Digitally Controlled Oscillator

Various DCO implementations exist in literature. It can be implemented as a DAC followed by a standard VCO. [67] employs 2 current base DACs and two loop filters while [68] employs a 10 bit current mode DAC and [76] a resistor string based DAC. To avoid the analog nature and design constraints of a DAC, the DCO might consist of units such as varactors in LC tank DCOs [55–58, 62] or inverters in ring oscillator DCOs [64–66] that are turned on or off depending on the value of the control word during operation to change the output frequency.

The DCO in this work is implemented as a fully digital three stage ring oscillator as shown in Fig. 64. Turning more inverters ON/OFF in parallel at each node (A,B,C) increases/decreases the frequency [53, 54, 65]. The many units that are connected between the three phases of the ring oscillator are designed and placed as a row and column matrix as shown in Fig. 65 to simplify the control circuitry.

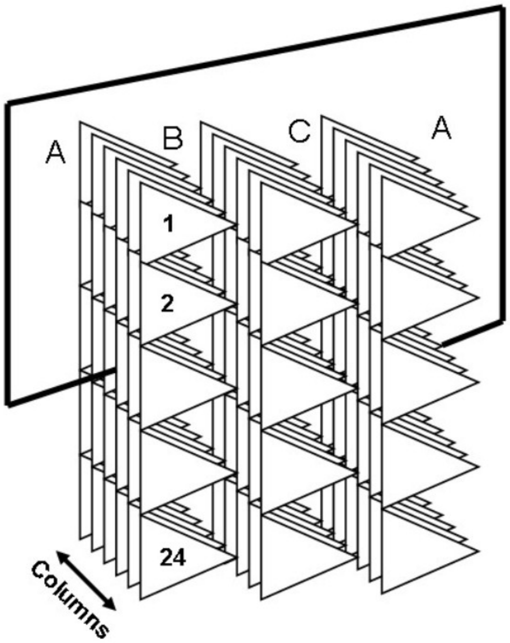


Fig. 64. Three stage ring oscillator based DCO 3-D representation

As seen in Fig. 62, the DCO is implemented as a hybrid combination of a matrix of unit weighted inverters and binary weighted inverters that are controlled by the smart shifter logic and digital loop filter output integer bits, respectively. This hybrid approach avoids binary to thermometer code converters and employs a single small digital loop filter as well as a direct path through a smart shifter to the DCO. To set the free-running frequency, the oscillator also contains several base inverters that are always active. And finally, some of the unit weighted inverters are controlled by the DSM to dither the output frequency.

In this prototype, the binary weighted control word is 5 bits. The LSB of the binary weighted control is connected to a single unit while the MSB has 16 units connected in parallel. Overall, the binary weighted portion of the DCO has 31 units, the DSM controls are connected to 7 units, the base DCO has 192 units and finally

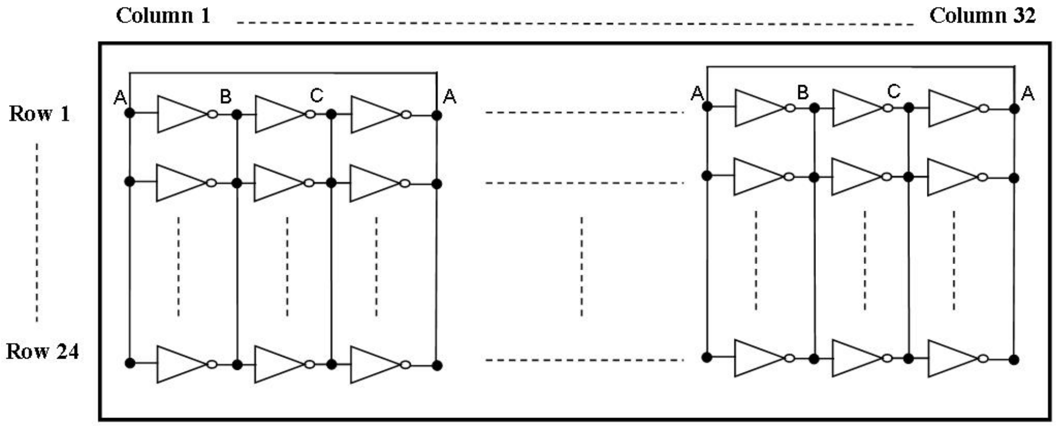


Fig. 65. Three stage ring oscillator based DCO put in a row-column matrix for ease in control

the row/column controllable matrix has 768 units (24 rows, 32 columns).

As explained in Section II. and Fig. 62 the MSBs of the TDC are passed directly to the DCO as a delta-control word (Δx) to turn ON/OFF units and adjust it's frequency, functioning like a coarse control path. The DCO shifts Δx units, acting like a big accumulator. Note that Δx is signed since the DCO frequency can be increased or decreased. The DCO control block that performs this shifting and controls the row/column matrix of the DCO is the smart shifter.

It should be noted that the proposed architecture employs a ring DCO to provide an all digital approach utilizing only digital logic cells. However, for applications that target stringent phase noise specifications, the same architecture can be used by replacing the ring oscillator with an ultra-low phase noise LC oscillator with varactor banks since the system treats the DCO as a black-box and is independent from it's circuit-level implementation.

D. Smart Shifter

If the DCO had a single row, the multi-bit shifting could be easily performed with a barrel shifter. If S is a m bit shift word, a barrel shifter can shift it's input word by S bits and consists of m consecutive MUX stages. Fig. 66 shows a standard three stage barrel shifter. Note that X and s are the input and shift control words, respectively. Xi is the input word of stage i, that is controlled by shift bit s_i. If s_i is a bit '1', Xi is shifted by 2^i bits. Note that in this explanation we will be referring to a left-shift of '1' and the thermometer code of the column word consists of ones in it's LSBs and zeros in it's MSBs.

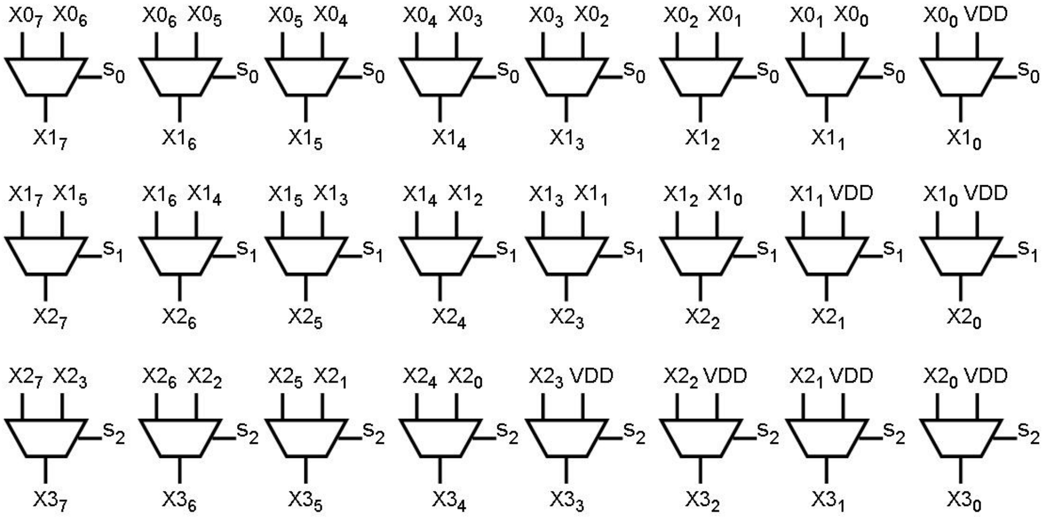

$X0_7$ $X0_6$ s_0 $X1_7$
$X0_6$ $X0_5$ s_0 $X1_6$
$X0_5$ $X0_4$ s_0 $X1_5$
$X0_4$ $X0_3$ s_0 $X1_4$
$X0_3$ $X0_2$ s_0 $X1_3$
$X0_2$ $X0_1$ s_0 $X1_2$
$X0_1$ $X0_0$ s_0 $X1_1$
$X0_0$ VDD s_0 $X1_0$
$X1_7$ $X1_5$ s_1 $X2_7$
$X1_6$ $X1_4$ s_1 $X2_6$
$X1_5$ $X1_3$ s_1 $X2_5$
$X1_4$ $X1_2$ s_1 $X2_4$
$X1_3$ $X1_1$ s_1 $X2_3$
$X1_2$ $X1_0$ s_1 $X2_2$
$X1_1$ VDD s_1 $X2_1$
$X1_0$ VDD s_1 $X2_0$
$X2_7$ $X2_3$ s_2 $X3_7$
$X2_6$ $X2_2$ s_2 $X3_6$
$X2_5$ $X2_1$ s_2 $X3_5$
$X2_4$ $X2_0$ s_2 $X3_4$
$X2_3$ VDD s_2 $X3_3$
$X2_2$ VDD s_2 $X3_2$
$X2_1$ VDD s_2 $X3_1$
$X2_0$ VDD s_2 $X3_0$

Fig. 66. Block diagram of a conventional 3-bit barrel shifter

The DCO is a row-column matrix where each row has a $WHOLEROW\_ON$ signal that overwrites the column setting and turns on the units of the whole row and a select signal ROW_SEL that allows a unit to be turned on, only when it's column control is also ON. In this setting, we need to not only shift units, but also detect how many available spots are left in the column word on the current row, and then wrap around to the next row and turn on more units if needed. Fig. 67 summarizes the algorithm that should be implemented in this shifter where Δx is the input shift word, $C_{av}(n)$ is the number of available units in the column word at time n and C_{max} is the total number of columns. The column word is in thermometer code, therefore, $C_{av}(n)$ as a binary word, is not available.

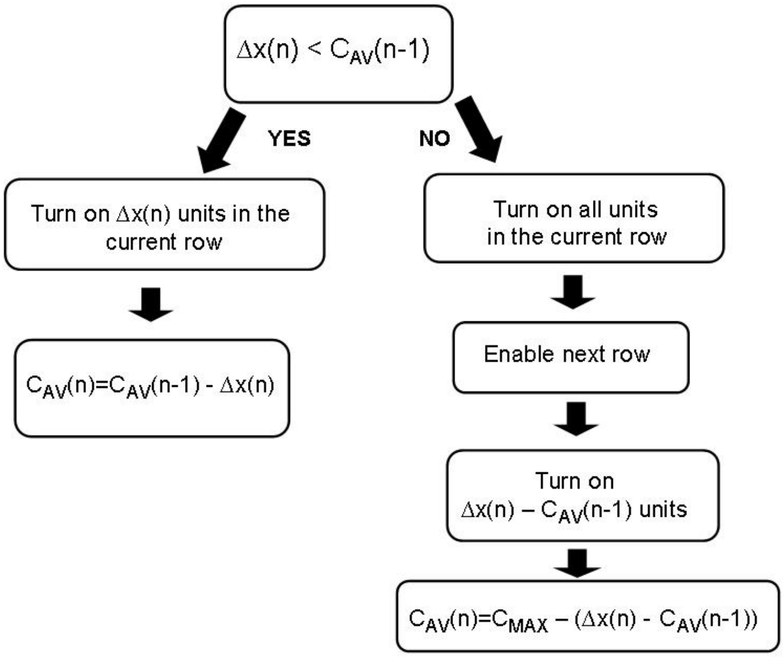

Δx(n) < C_AV(n-1)
YES
NO
Turn on Δx(n) units in the current row
C_AV(n)=C_AV(n-1) - Δx(n)
Turn on all units in the current row
Enable next row
Turn on Δx(n) – C_AV(n-1) units
C_AV(n)=C_MAX – (Δx(n) - C_AV(n-1))

Fig. 67. Operational flow diagram of the smart shifter

Note that to implement the algorithm of Fig. 67, a synthesis tool would use an accumulator to keep track of the total number of units that are shifted, and a subtractor to detect, if moved to the next row, how many more units to be turned on in the next row. This would require determining the value of $C_{av}(n)$ as a binary word while in implementation the column word C is available in thermometer code.

Therefore, the tool would have converted the column word into binary code, defeating the purpose of using this shifter in the first place.

The proposed shifter performs the functions described in Fig. 67 in a single loop cycle while maintaining a simple structure similar to a barrel shifter. The block diagram of a 3 bit version of the proposed smart shifter is shown in Fig. 68. Wire connections are not drawn but are implied through shared wire-names for simplicity. Also, the shifter shifts the column word from the previous update cycle to perform the accumulation in (7.4), then:

$$X0(n) = X3(n-1) \tag{7.5}$$

In the DPLL prototype, the shift word Δx is a 5-bit word and there are 32 columns, hence, a 5 stage version of the shifter of Fig. 68 is implemented. Δx is signed which means the shifting should be bidirectional. This will be discussed in the next section. For now, let's assume Δx is positive and we are only turning more units ON.

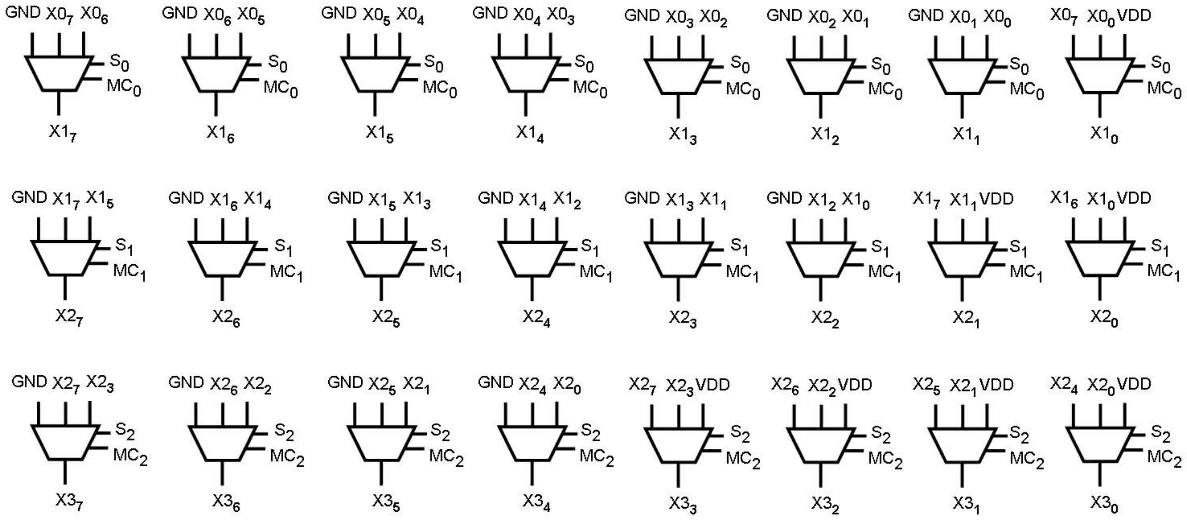


Fig. 68. Block diagram of a 3-bit implementation of the proposed smart shifter

Table XVIII. Truth table of 2-bit controlled MUX

S	MC	OUT
0	X	IN2
1	0	IN1
1	0	IN3

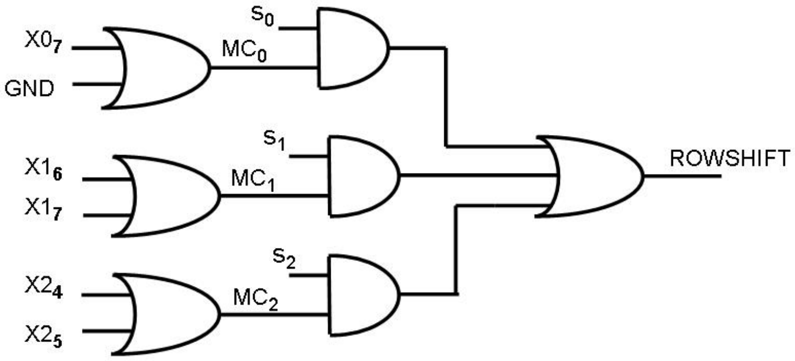


Fig. 69. Generation of MC controls and rowshift signals in 3-bit smart shifter

Similar to a barrel shifter, shifting of the column word is divided into stages where the input word Xi of stage i is shifted based on the value of s_i and the new control MC_i. Table XVIII summarizes the operation of the MUXs. If:

$$Zeros(Xi) < 2^i \ \ AND \ \ s_i = 1 \tag{7.6}$$

where $Zeros(Xi)$ is the number of units available in word Xi. Then,

$$MC_i = 1 \ , \ \ ROWSHIFT = 1$$

$$WHOLEROW\_ON_J = ROWSEL_{J+1} = 1 \tag{7.7}$$

where j is the row index and $ROWSHIFT$ signal causes the current row to be turned ON completely and the next row to be enabled while MC_i causes the MIXES of stage i to pass their third input.

Note that our goal was to turn on 2^i units in stage i but enough units were not available based on (7.6). Then, after turning all the available units in the current row and moving to the next row, we should reset all the columns (new row has all the units available) and then turn on the remaining required number of units from 2^i as was summarized in Fig. 67.

The resetting is done by connecting the IN3 ports of the MUXs to ground while turning on the remaining required units is done by connecting the 2^i MSBs of Xi to the first 2^i inputs (IN3) of the MIXES of stage i. To understand this, let's assume that the last (MSB) 2^i bits of X_i were high, it would mean that the input word of stage i is all ones and in stage i we wouldn't be able to shift any new units. So we would turn the current row ON, enable the next row, reset the column word (pass all zeros) and turn on the first (LSB) 2^i units as the output of stage i, $Xi + 1$.

In another example, if there was only 1 unit available in the input X_i, then we would follow the same steps but then after resetting the column word, we would turn on $2^i - 1$ units as the output of stage i, since we already turned on 1 unit at the previous row. Likewise, by connecting the leftmost 2^i bits of X_i to the first 2^i IN3 connections of the next stage, we ensure that the units that were not available (already on) among the last 2^i in the previous stage, would lead to new units that are turned on in the next stage. The IN3 inputs of the remaining MIXES in this stage are connected to ground since the column word is reset as the new row is enabled.

While theoretically we should check the last 2^i bits of X_i to check if (7.6) is true, since X_i are thermometer coded, ideally, if a bit is '1', all bits towards it's right will be '1' and if a bit is '0', all bits towards it's left will be '0'. Then, instead of checking all of the 2^i most significant bits, checking only the 2^ith bit from the left will be enough. If it is high, it means there are not enough units to turn on at stage i and we should turn on MC_i. For instance at the input of stage 2 where 4 units will be turned ON, if the 4th bit from the left at the input is high then we have at most 3 or less units available at the input so we can turn MC_2 high. If it is a '0', then all bits towards it's left should be zero so the regular shifting can continue.

However, this is assuming there are no bubbles or errors in the thermometer code. To avoid their effect, we can check more bits as a precaution. Fig. 69 shows the generation of MC_i signals, where we check the 2^ith and the 2^{i-1}th bits from the left, to ensure that if 2^ith bit is a '0' (a bubble) but the bit towards it's left is a '1', we still detect it.

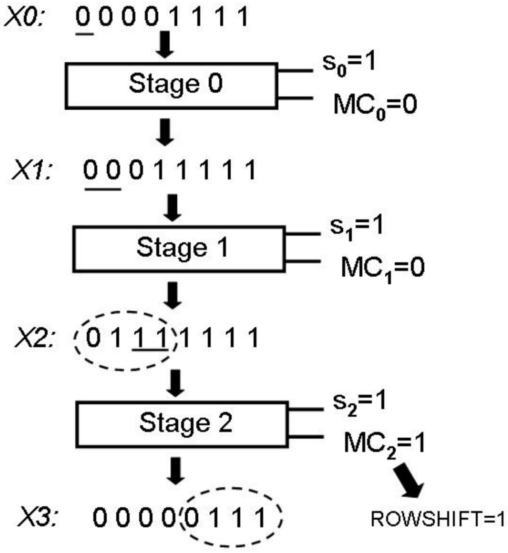


Fig. 70. Sample operation of the 3-bit smart shifter

If (7.6) is not true, MC_i remains low and the shift continues like a barrel shifter within the same row. The same condition is checked for every stage of the shifter. The row shifter is a simple one-bit shifter since the maximum value of Δx can be 31 and therefore, we can't shift more than one row per update cycle. The $ROWSHIFT$ signal generated by the smart shifter is fed to the 1-bit rowshifter where

$$ROWSEL_{J+1} = WHOLEROW\_ON_J \tag{7.8}$$

and j is the row index.

A sample operation of a 3-bit smart shifter is demonstrated in Fig. 70 where the column word shows that the current row has only 4 units that are off while the shift word requires 8 more units to be turned on. Then, the current row should be turned ON as a whole, next row should be enabled and the column word should have 4 units that are ON to achieve a total of 8 new turned on units. The bits that generate MC_i in each stage are underlined. In this example, stages 0 and 1 act as a regular barrel shifter while in stage 2 MC_2 is set, $ROWSHIFT$ is enabled and the final output represents the column word of the next row.

1. Bidirectional Shifting

Since the input of the shifter is Δx which is the MSBs of the output of TDC and loop gain A, it is signed (DCO frequency can be increased and decreased). Therefore we need bi-directional shifting such that if Δx is positive, we shift the column word towards left with ones (turn on units) and turn on more rows and if negative, we shift zeros towards right and turn off rows. In it's simplest form, bidirectional operation can be achieved by implementing two separate shifters where one shifts ones and the other zeros, and choose the output of one of them with an additional MUX at the end, controlled by the sign bit. However, this would result in two 5-stage smart shifters increasing the complexity of the DCO control scheme.

To employ only a single smart shifter, we propose the bidirectional shifter as described in Fig. 71 which utilizes the smart shifter that only performs left-shift by ones. The shifter benefits from inverting and bit-swapping of the bits (for an m bit word, the m^{th} bit becomes the first, $(m-1)^{th}$ bit becomes the second and so on) before using the smart left shifter. Fig. 71 demonstrates an example right-shift of zeros that is achieved by the left-shifter.

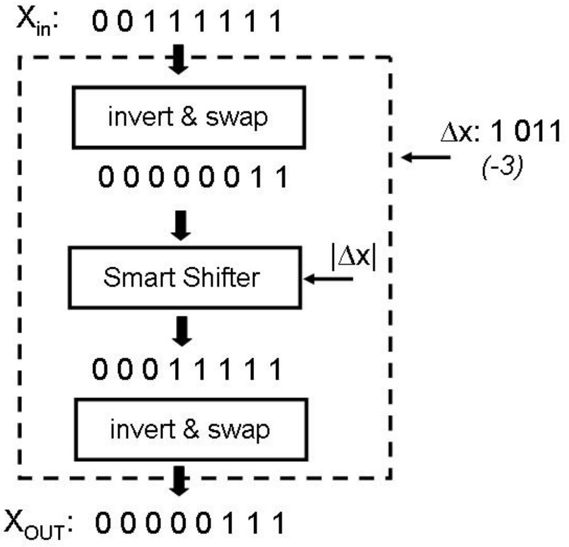


Fig. 71. Block diagram and sample operation of right-shifting using the left-shift smart shifter

The overall bidirectional shifter has two stages of MUXs that are controlled by the sign bit of Δx. In terms of added delay, a bidirectional shifter that employs two shifters and chooses one, would add one additional level of MUX to the overall shifter delay while the proposed bidirectional shifting adds two levels of MUXs to the overall shifter delay, assuming that the inverted versions of the shifter output word is already available. The bit-swapping complicates the routing but it is still less area intense than two separate shifters.

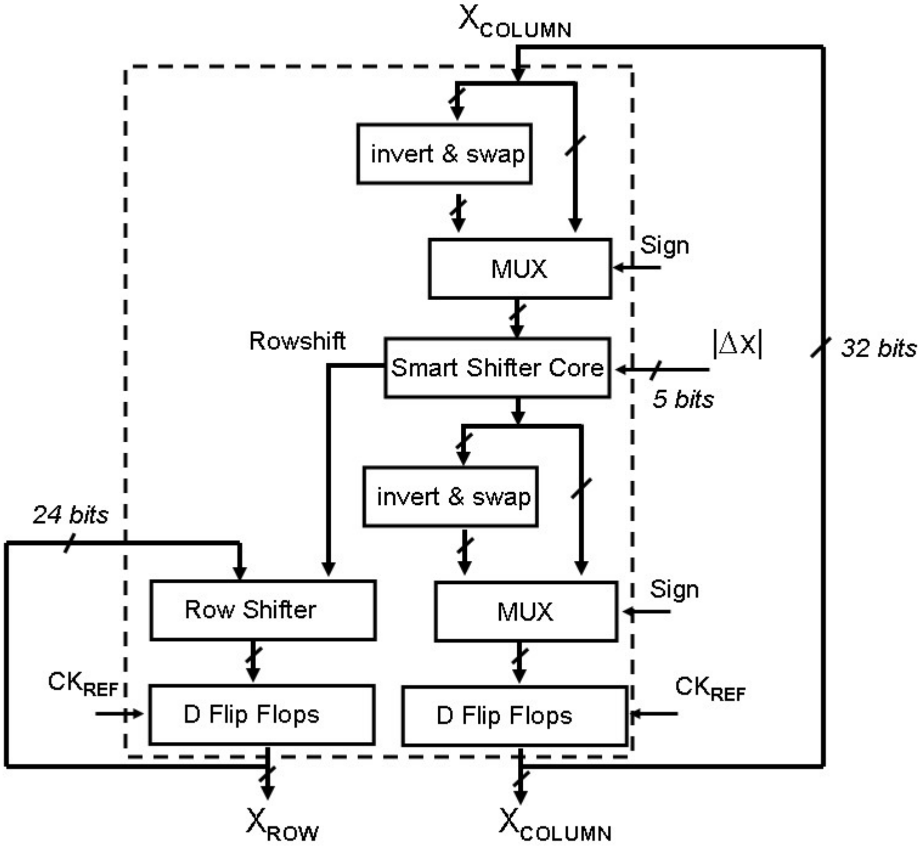


Fig. 72. Block diagram of the complete bidirectional row/column shifter as the DCO interface

The final bidirectional smart row/column shifter that serves as the coarse DCO interface is shown in Fig. 72. The smart shifter core is a 5-bit version of the shifter shown in Fig. 68 that takes Δx as its shift control and performs left-shift of ones. The final output of the shifter passes through flip-flops clocked with the system-clock before being fed to the DCO as row and column words where the column word is 32 bits and the row word is 24 bits.

E. Digital Loop Filter

The digital loop filter is a proportional integral filter, as shown in Fig. 62. It consists of two 10-bit adder/subtractors and is custom-designed in this ADPLL prototype. To simplify the implementation of the multiplication factors, the loop filter coefficients α and β are both powers of 2 as shown in Table XVII. Therefore, the multiplication in the loop filter is performed through shifting.

The adder/subtractors in the loop filter as well as the adders in the DSM and the TDC core that process the GRO output, are all implemented as Carry-Skip Adders with Manchester Carry Chain [77]. This architecture is used for its superior speed performance with respect to ripple carry adders and simpler circuitry than look-ahead architectures.

A Manchester Carry Chain (MCC) [78] is used to produce the carry output of the adder and is similar to a standard ripple carry architecture. However, the carry is produced through a dynamic path (often implemented through transmission gates). In addition to the standard *Propagation* signal for the carry, the MCC also uses *Generate* and *Kill* signals that determine the output carry through a quick path to minimize the carry output delay. The *Generate* signal pulls the carry output to a logic high and *Kill* signal pulls the carry output to a logic low through dynamic quick-paths. The adder block creates the signals that control the MCC as follows [77]:

$$Propagate_i = A_i \text{ XOR } B_i \tag{7.9}$$

$$Generate_i = A_i \text{ AND } B_i \tag{7.10}$$

$$Kill_i = A_i \text{ NAND } B_i \tag{7.11}$$

where A_i and B_i are the i^{th} bits of inputs A and B.

Fig. 73 shows the MCC implementation used in the loop filter adders. Note that

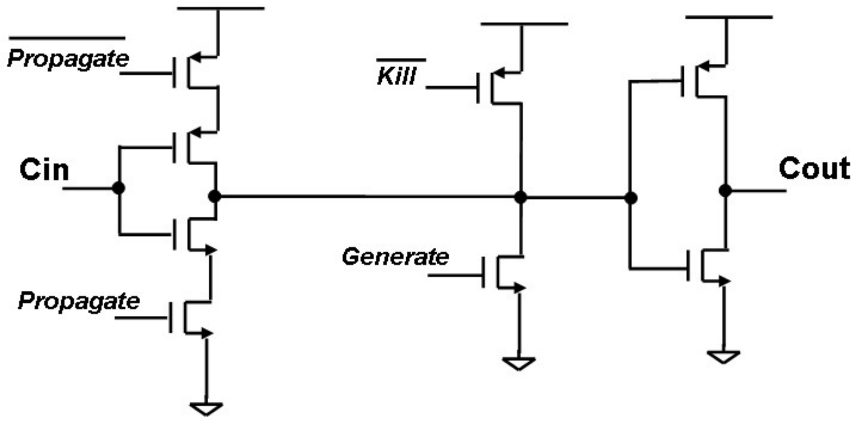


Fig. 73. Transistor level implementation of a Manchester Carry Chain

instead of transmission gates, enable/disable inverters are used to maintain the signal level and integrity throughout the 10 stages of the 10-bit adders. Also note that the XOR logic in the adders to produce the propagate the signal (as well as the ones that produce the adder output), are implemented through MUXs. Theese MUX blocks are also implemented through enable/disable inverters, rather than transmission gates, to maintain signal level integrity. Fig. 74 shows the implementation of a 1-bit full adder that uses MCC to produce its carry output. The 10-bit adder/subtractors in the loop filter consist of 1-bit full adder units of Fig. 74.

In a multi-bit adder such as a 10-bit adder, the worst case delay is determined by the case where the input carry is propagated to the final carry output, hence 10 serial stages of carry propagation. Since this is a limiting factor on the speed of the adder, a Carry-Skip Adder (CSA) [79], which provides a shortcut to the carry propagation to minimize the total stages that carry has to go through, is used.

In a CSA, the adder is designed into n bit groups such that, if all the n stages propagate their input carry, a skip-logic passes the input carry to the output through

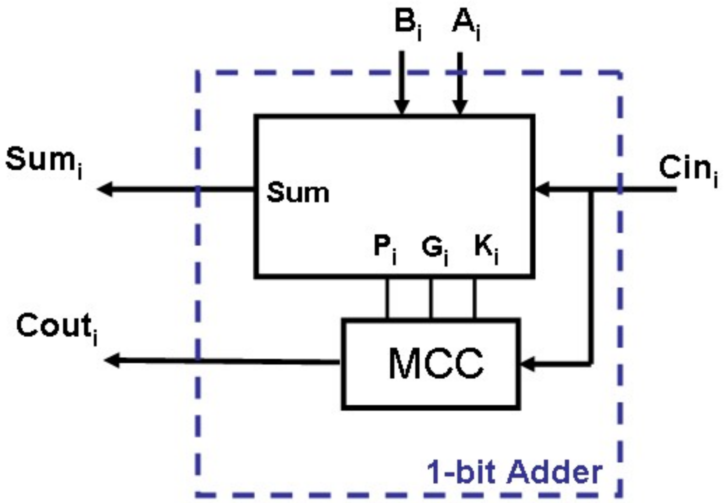

B_i A_i
Sum_i
Sum
Cin_i
P_i G_i K_i
Cout_i
MCC
1-bit Adder

Fig. 74. A 1-bit full adder using Manchester Carry Chain

a quick path. Fig. 75 shows the block-level implementation of a 3-bit CSA where the adder is implemented with MCC based full-adders of Fig. 74. The skip signal provides a quick path to deliver the input carry to the output. In the loop filter, the 10-bit adder/subtractors are designed with the 3-bit CSA groups of Fig. 75 such that the final adder has a 3-3-3-1 bit grouping.

Both of the adders in the loop-filter of Fig. 76, (the adder in the digital integrator and the sum block at the output of the loop filter) are adder/subtractors since the TDC output is signed. Note that a signed-magnitude representation [80] is used throughout the ADPLL. To check if addition or subtraction is to be performed, we check the sign bit of the signed input (a sign bit of '1' represents a negative number and a bit '0' represents a positive one).

The digital integrator output (D in Fig. 76) and the summer output (E) in the loop filter should be positive since the DCO cannot hold a frequency less than its free-running frequency (in an analogy to the charge-pump based PLL, the control voltage

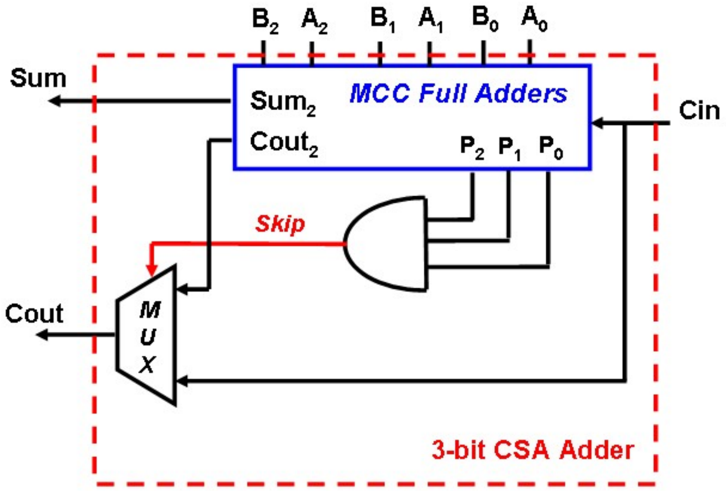


Fig. 75. Block diagram of the 3-bit Carry-Skip Adder used in the loop filter

is not negative since the VCO should oscillate at a frequency equal to or larger than its free-running frequency). Then, in subtraction mode, the adder/subtractors shown in Fig. 76 perform:

$$D(n) = D(n-1) - C(n-1) \quad \text{(for sign(C(n-1))=1)}$$
$$E(n) = D(n) - B(n) \quad \text{(for sign(B(n))=1)} \tag{7.12}$$

The subtraction is performed by using the two's complement of the input to be subtracted. The two's complement (TC) of an m-bit binary number X is defined as [80]:

$$TC(X) = 2^m - X \tag{7.13}$$

which, can be implemented as:

$$TC(X) = 2^m - X = \overline{X} + 1 \tag{7.14}$$

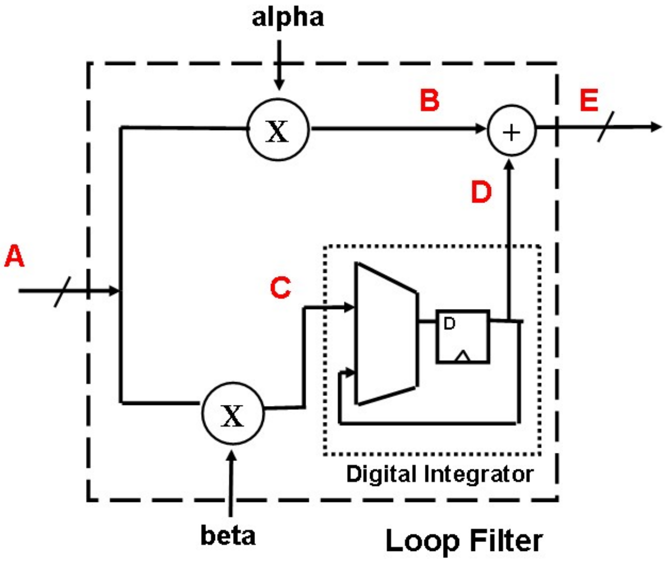


Fig. 76. Block diagram of the loop filter

where $\overline{X}$ is the one's complement, or the complement (inverted version of) of X, which can be obtained through simple inverters.

Then, to perform the subtraction of two words, Y and X, to obtain Z such that:

$$Z = Y - X \tag{7.15}$$

first, we create the two's complement of the word to be subtracted (X):

$$Z' = Y + TC(X(n)) = Y + \overline{X} + 1 \tag{7.16}$$

while $\overline{X}$ is implemented with inverters, the addition of 1 to $\overline{X}$ is performed by connecting the sign bit to the carry input of the adder. Z' is given by:

$$Z' = Y - X + 2^m = Z + 2^m \tag{7.17}$$

Note that during the subtraction, the subtractor generates a carry output of '1'

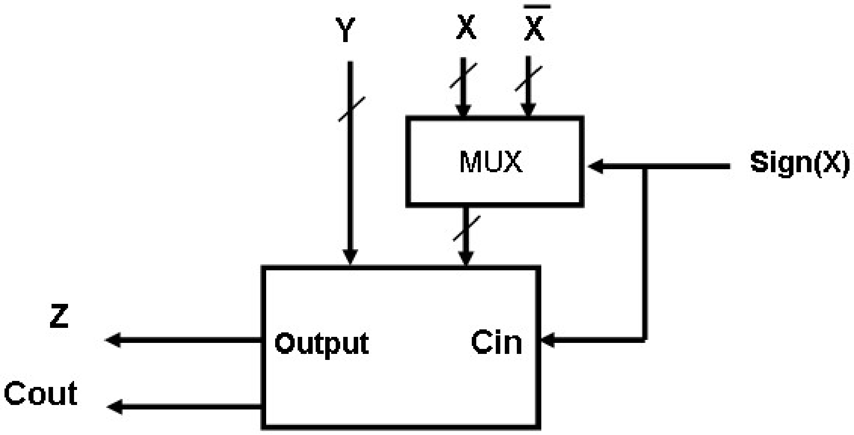


Fig. 77. Block diagram of an adder/subtractor

(the 2^m element in 7.17) which should be discarded to obtain the desired output Z. Fig. 77 demonstrates the adder/subtractor described above.

The adder/subtractor of Fig. 77, generates an erroneous result for $Y - X$ if $X > Y$ [80]. As explained earlier, the loop filter output cannot be negative and therefore this case should not occur. If the subtractor makes such an erroneous calculation, the carry output of the subtractor will be '0'. Then, such an errors can be corrected by checking the sign bit (that determines add or subtract mode) and the carry output (a '1' means correct subtraction, a '0' means erroneous result).

The 10-bit adders (that consist of three 3-bit CSA adders of Fig. 75 and one 1-bit full adder of Fig. 74) are converted into adder/subtractors with additional complement and MUX logic as shown in Fig. 77.

F. Other ADPLL Building Blocks

As discussed in Section II. C. and shown in Fig. 63, the time to digital conversion block consists of a standard PFD, a non-overlapping clock generator (N.C.G) and a TDC core that converts the time-width of it's enable signal into a digital word. As explained earlier, the same TDC core serves to detect both phase and frequency difference between the reference and the divider outputs. TDC designs, in general, consist of coarse and fine time resolution calculations.

The TDC core architecture is based on the Multiphase Gated-Ring Oscillator (GRO) TDC [72]. This method presents a linear transfer characteristic due to the scrambled phase states that are inherent in the oscillator which allows linear techniques for loop analysis. Furthermore, high resolution is achieved due to the lower delay per stage and high matching between delay stages.

The frequency dividers are implemented with dynamic True Single Phase Clocking (TSPC) and Extended True Single Phase Clocking (E-TSPC) techniques [5] since the DCO is a single-ended inverter based ring oscillator. As shown in Fig. 62, multiple division paths including a /2 prescaler and dual modulus prescalers that divide by 2/3 and by 7/8 are implemented to realize 6 different division ratios as listed in (7.1). E-TSPC is used for dual modulus prescalers and TSPC is used to implement divide by 2 prescaler. A MUX is used to select a divider path based on the digital controls. The lower frequency /8 divider which follows the MUX is implemented using standard transmission gate based flip-flops. As a power-saving measure, each division path is preceded by a buffer. The MUX controls also enable/disable these buffers to ensure that during the ADPLL operation only one division path is enabled.

The digital DSM is a three stage MASH structure that is clocked at:

$$F_{DSM} = \frac{F_{DCO}}{N/8} \tag{7.18}$$

where N is the total divide ratio and based on the external controls selected by the user to determine the operating frequency, it takes on values given by (7.1). Note that in lock $F_{DSM} = 8F_{REF}$. The flip-flops of the DSM are implemented in E-TSPC logic.

G. System Simulations

A duplicate of the system is built in MATLAB Simulink environment to analyze the time-domain behavior of the system. In the TDC, since we sample the count value and it is processed by the TDC and delivered to the subsequent blocks in the next cycle, we introduce a delay in the forward path of the loop, this is taken into account in system simulations that ensure stability. A second delay in the loop forward path is added at the DCO control at the smart shifter and row shifter outputs. The Simulink time-domain system simulations employ 20ps TDC resolution and 6.8MHz/LSB DCO unit gain (KDCO) and a reference frequency of 125MHz. The DCO free-running frequency is 1.89GHz, an intentional fractional number to imitate a realistic response where the DSM will dither the DCO output in lock.

In the circuit implementation, unlike the fine path where the loop filter output explicitly holds the control word W_{FINE}, in the coarse path the smart shifter provides a delta-unit value (Δx) per cycle and there is no explicit accumulator that holds W_{COARSE}. However, in time-domain simulations, the implicit accumulation of the DCO smart shifter is represented explicitly and therefore we can monitor the total control word W_{TOTAL} that holds the DCO frequency information to observe loop dynamics.

$$W_{TOTAL} = W_{COARSE} + W_{FINE} \tag{7.19}$$

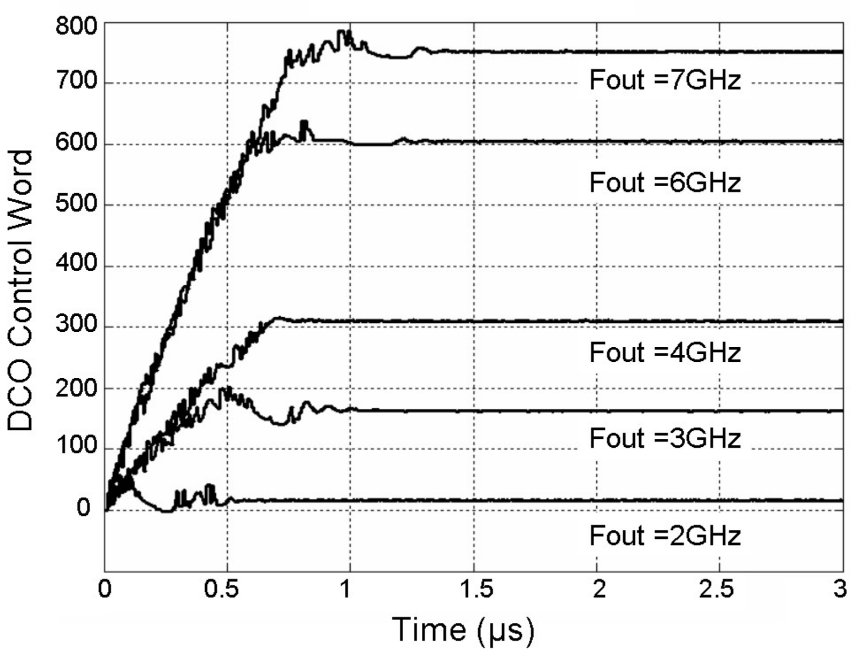


Fig. 78. Simulink time-domain simulations of the ADPLL, DCO total control word for ADPLL operation frequencies between 2GHz-7GHz

Fig. 78 shows the DCO total control word for ADPLL output frequencies of 2GHZ to 7GHz. Through the use of the variable gain block A in the system, loop stability is maintained for the wide range of operating frequencies. Due to the quick coarse path, the difference between the settling time of 7GHz and 2GHz operations is less than a microsecond in these simulations.

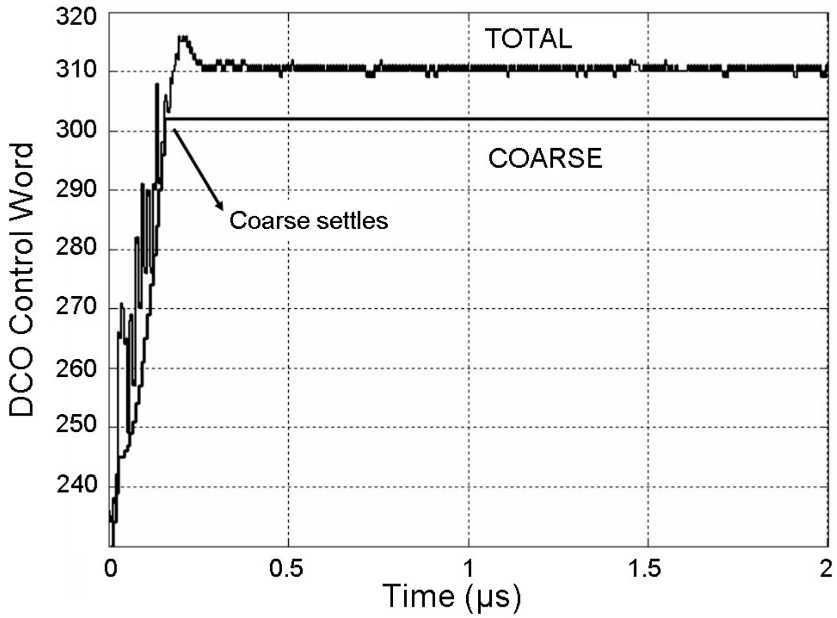


Fig. 79. Simulink time-domain simulations of the ADPLL, detail of total and coarse DCO control words for 4GHz operation

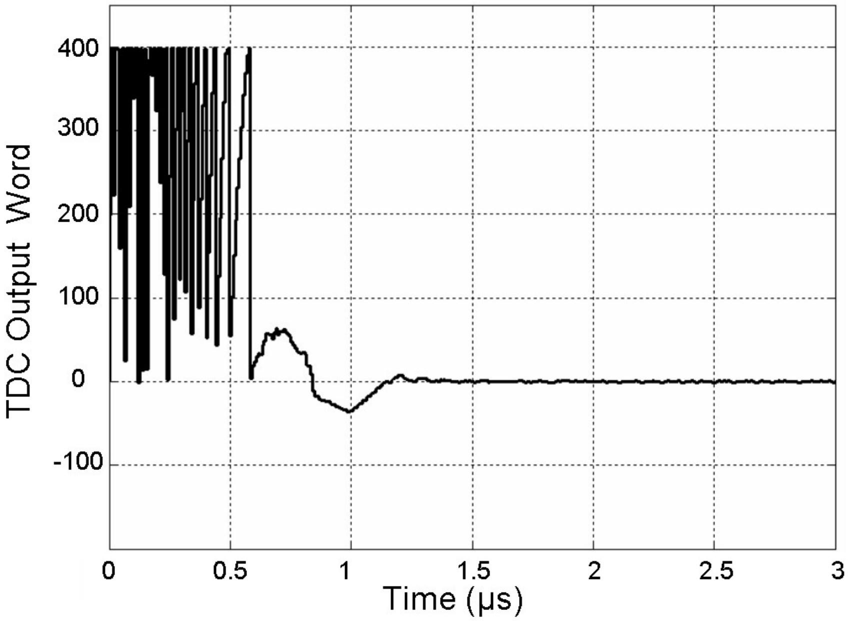


Fig. 80. Simulink time-domain simulations of the ADPLL, TDC output at 6GHz operation

Fig. 79 provides a detailed look at the coarse and total control words of the DCO for 4GHz operation. It is seen that the coarse path settles when the DCO is in the close vicinity of the target frequency and the fine path sets the lock. This shows that by exploiting the inherent digital nature of the loop and using the MSBs and LSBs of the digital TDC output we can avoid lock-detectors or explicit dual-loop architectures. Fig. 80 shows the TDC output for 6GHz operation, which demonstrates the loop error settling to zero.

H. Measurement Results

The ADPLL prototype is fabricated in UMC 90nm digital CMOS technology. The dies are packaged in a QFN type surface mount package and mounted on an FR-4 printed-circuit-board (PCB) for measurements.

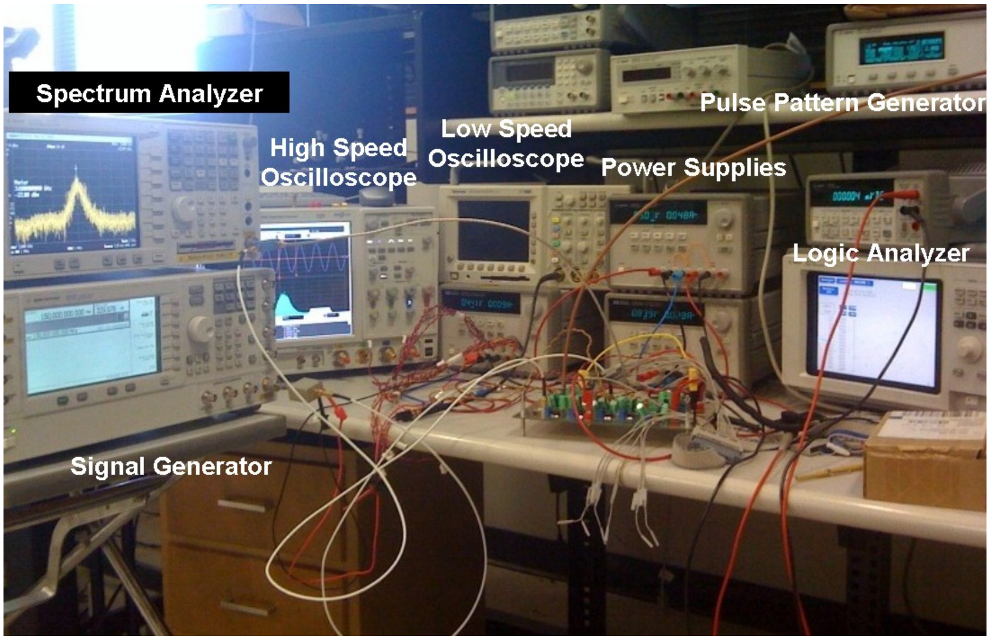


Fig. 81. Measurement instruments and setup for ADPLL

Fig. 81 demonstrate the laboratory instruments that are used in the measurements of the ADPLL. The high speed oscilloscope is an Agilent Infinium DSA91304A Digital Signal Analyzer (13GHz bandwidth, 40Gsa/s) which is used for jitter measurements. The spectrum analyzer is used to measure the phase noise and output frequency spectrum. The signal generator is used to generate the reference signal of the ADPLL while the low frequency oscilloscope is utilized to observe the reference signal as well as other control signals applied to the loop.

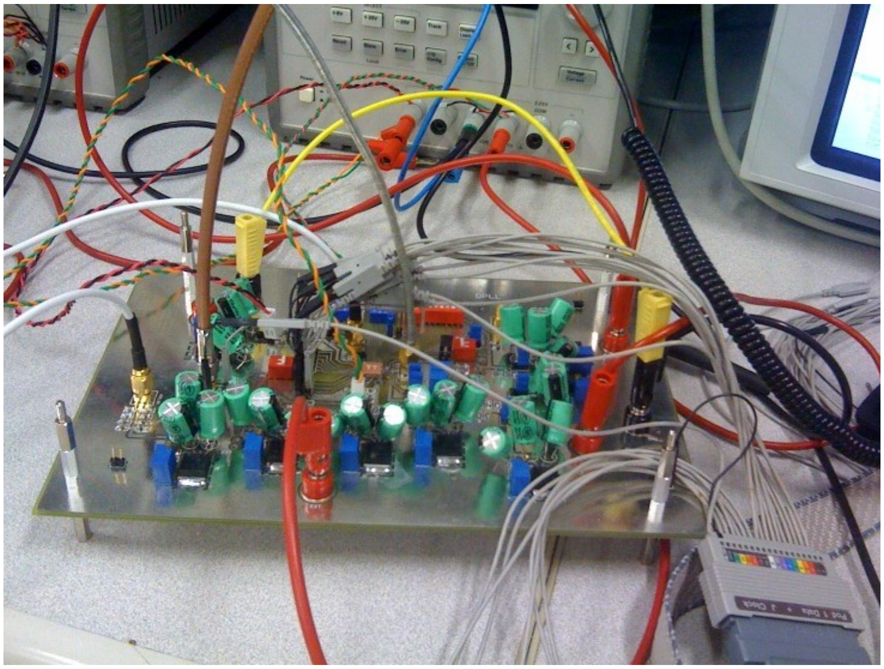


Fig. 82. Printed circuit board of ADPLL with connecting cables

Fig. 82 shows the printed circuit board (PCB) that was designed to measure the ADPLL with the connector cables for instruments during measurements. Fig. 83 shows the PCB in detail. Since the ADPLL is fully digital, the building blocks are

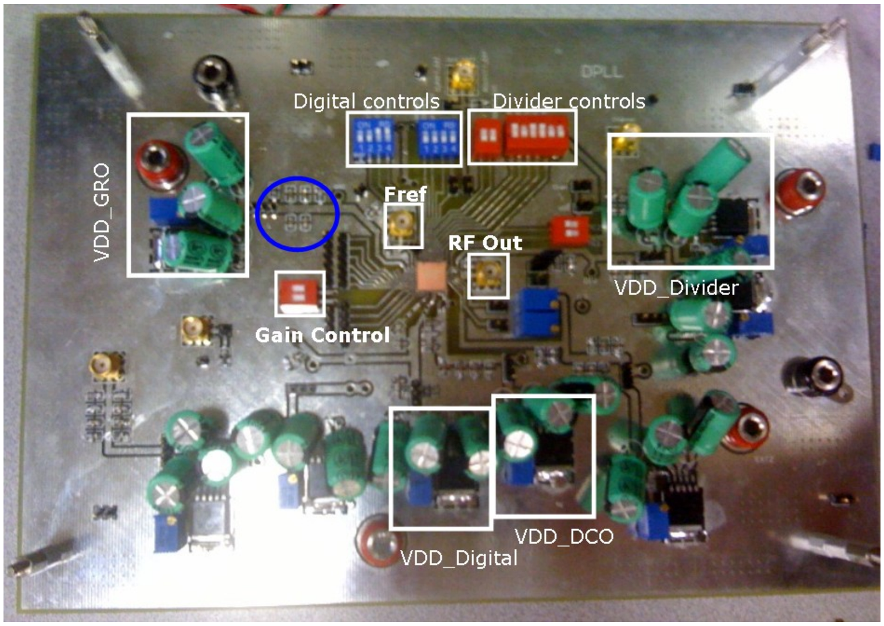


Fig. 83. Printed circuit board of ADPLL

digitally controlled. Therefore, the PCB mainly consists of digital controls and power supply generation blocks (some are marked on the PCB in the figure) that perform on-board supply regulation. The blue circle points to the supply decoupling capacitors used on the PCB for the GRO supply. Such surface-mount capacitors are used for all of the supplies regulated on the board. The reference frequency input of the chip is marked as well as the RF output of the ADPLL.

1. DCO Measurements

The performance of the DCO is very significant in determining the performance of the overall ADPLL. The DCO phase noise is key in determining the ADPLL noise performance at the out-of-band offset frequencies, while its tuning range is the limiting factor that determines the ADPLL wide range capability. Finally, due to its very wide range and high frequency of operation, the power consumption of the DCO is the biggest contributor to the total power consumption of the ADPLL.

Table XIX summarizes the measured tuning range of the DCO for various supply voltage levels. It is seen that at 1V supply, which is the nominal supply voltage for the 90nm CMOS technology, a 4.8GHz tuning range is achieved. While a larger range is achieved at 1.2V supply, it is seen that the power consumption also increases.

Table XIX. Measured DCO power supply level and tuning range

DCO Supply	Tuning Range	DCO Total Current
VDD=0.8V	1.45GHz - 5.05GHz	11mA - 40mA
VDD=0.9V	1.95GHz - 6.15GHz	12mA - 54mA
VDD=0.95V	2.25GHz - 6.7GHz	14mA - 61mA
VDD=1V	**2.5GHz - 7.3GHz**	**16mA - 69mA**
VDD=1.2V	3.37GHz - 9.17GHz	23mA - 102mA

The measured frequency spectrum of the DCO at the minimum and maximum frequencies for 1V supply voltage are shown in Fig. 84 and Fig. 85 while the minimum and maximum frequencies are demonstrated for 0.9V supply voltage measurements in Fig. 86 and Fig. 87, respectively.

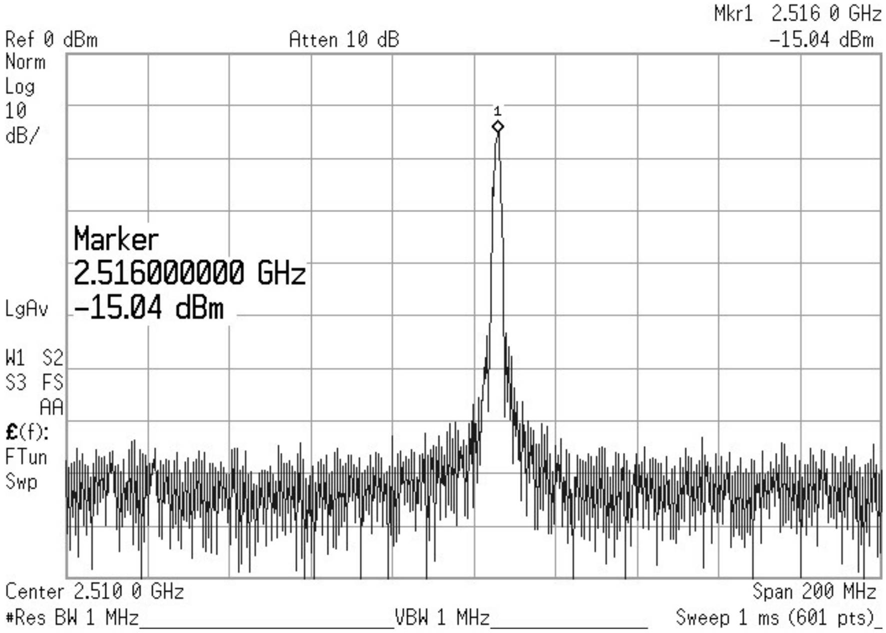


Fig. 84. DCO Output frequency spectrum - minimum frequency for VDD=1V

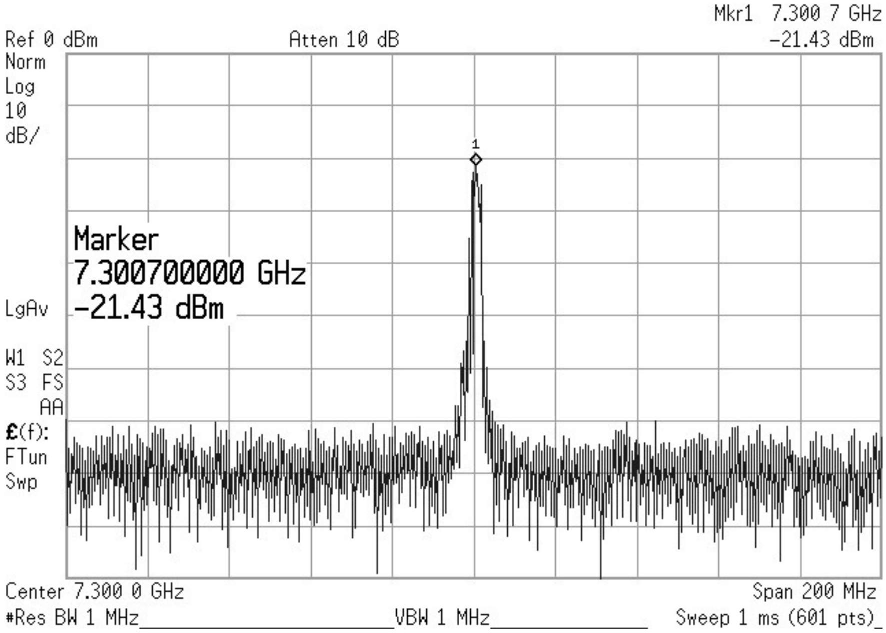


Fig. 85. DCO Output frequency spectrum - maximum frequency for VDD=1V

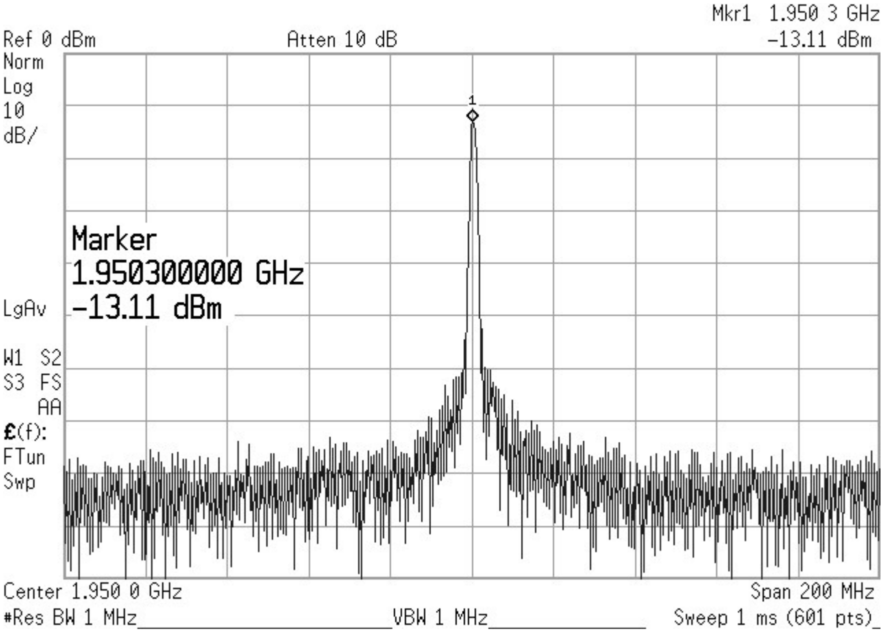


Fig. 86. DCO Output frequency spectrum - minimum frequency for VDD=0.9V

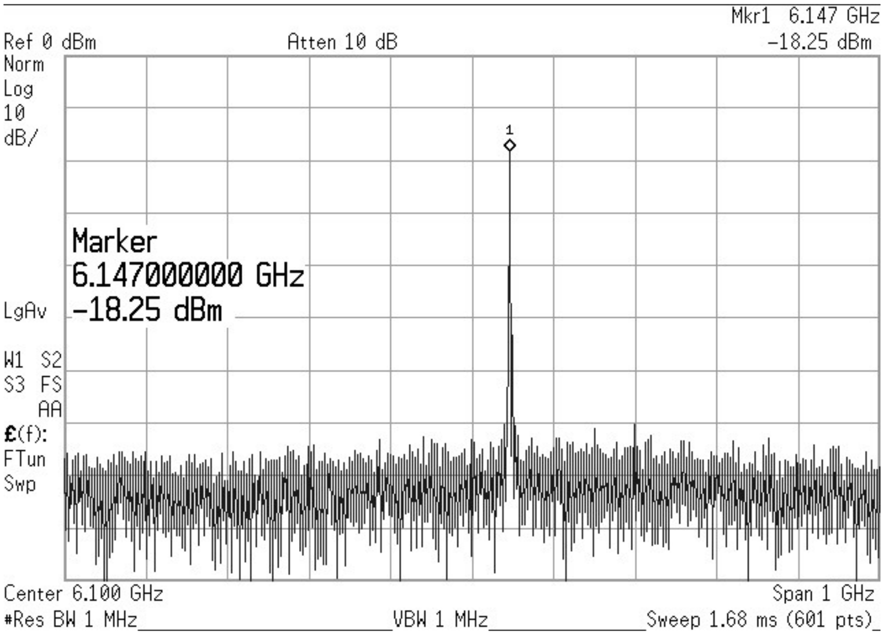


Fig. 87. DCO Output frequency spectrum - maximum frequency for VDD=0.9V

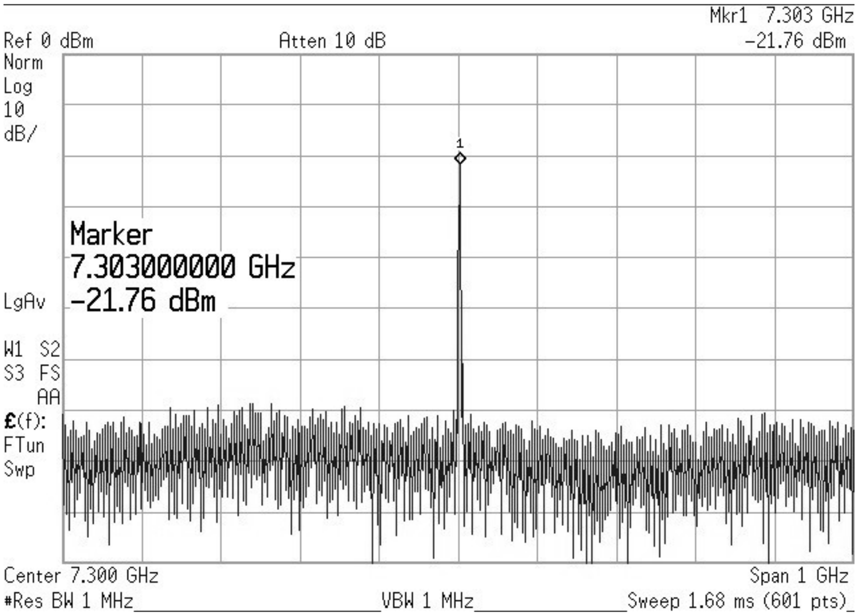
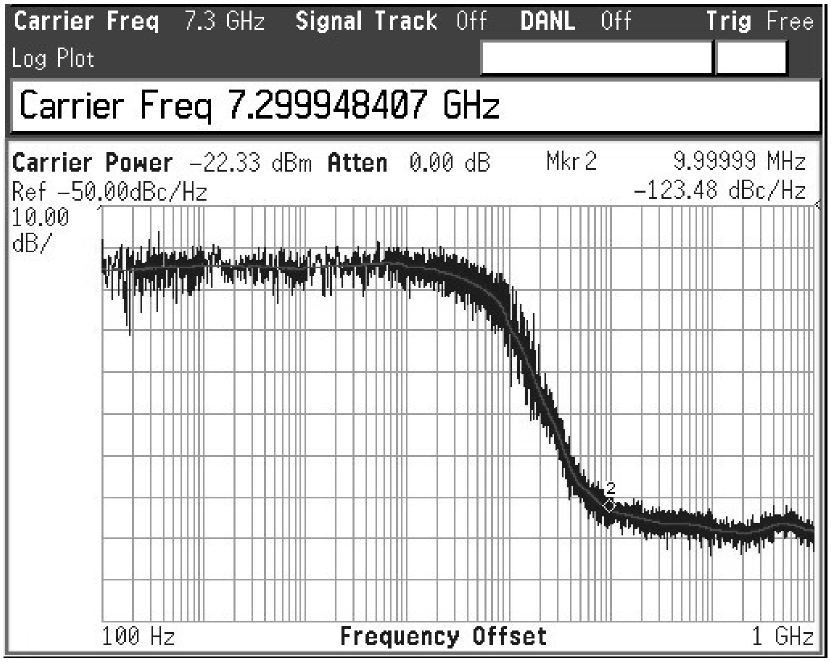


Fig. 88. DCO wide span output frequency spectrum at 7.3GHz operation

Fig. 88 shows the frequency spectrum of the DCO, in a 1GHz span setting. This demonstrates the clean output spectrum of the DCO for a wide span.

The PLL feedback loop acts as a low-pass filter to noise sources at the input of the PLL while it behaves as a high-pass filter to the noise sources in the oscillator as discussed in Chapter VI Section C. Therefore, the phase noise of the DCO is critical in determining the out-of-band noise performance of the DPLL. Note that LC tank based oscillators demonstrate better phase noise than ring oscillators, due to their band-pass filter shaped frequency response that suppresses the undesired noise elements in the frequency spectrum [49]. However, the band-pass filtering nature of these oscillators result in a good noise performance for narrow-range operations. In the proposed ADPLL, our goal is to demonstrate a wide-range operation to achieve a multi-protocol compatible PLL. Therefore, it is important to achieve a good noise performance in this wide-range operation.



Carrier Freq 7.3 GHz Signal Track Off DANL Off Trig Free

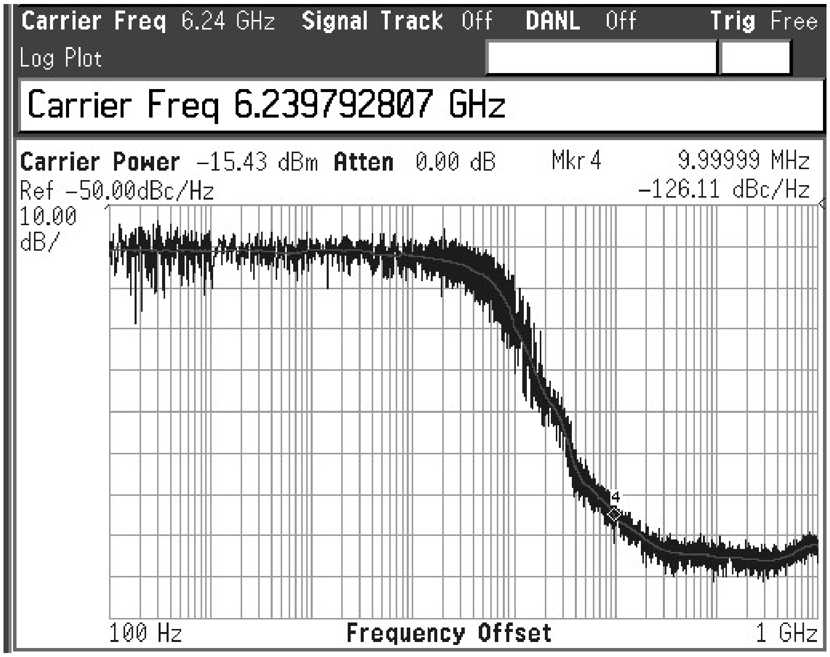
Log Plot

Carrier Freq 7.299948407 GHz

Carrier Power -22.33 dBm Atten 0.00 dB Mkr2 9.99999 MHz

Ref -50.00dBc/Hz -123.48 dBc/Hz

Fig. 89. DCO phase noise spectrum for 7.3GHz operation



Carrier Freq 6.24 GHz Signal Track Off DANL Off Trig Free

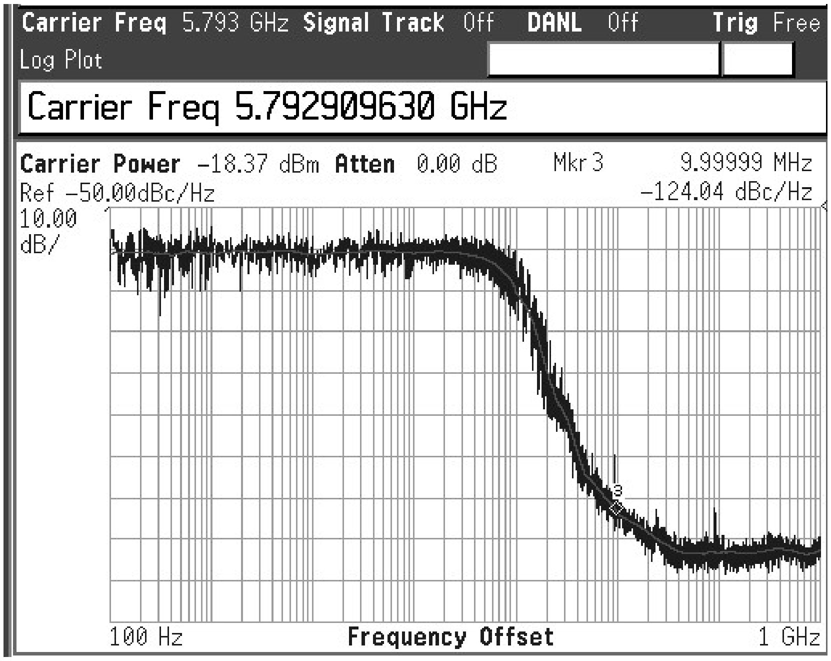
Log Plot

Carrier Freq 6.239792807 GHz

Carrier Power -15.43 dBm Atten 0.00 dB Mkr4 9.99999 MHz

Ref -50.00dBc/Hz -126.11 dBc/Hz

Fig. 90. DCO phase noise spectrum for 6.24GHz operation



Carrier Freq 5.793 GHz Signal Track Off DANL Off Trig Free

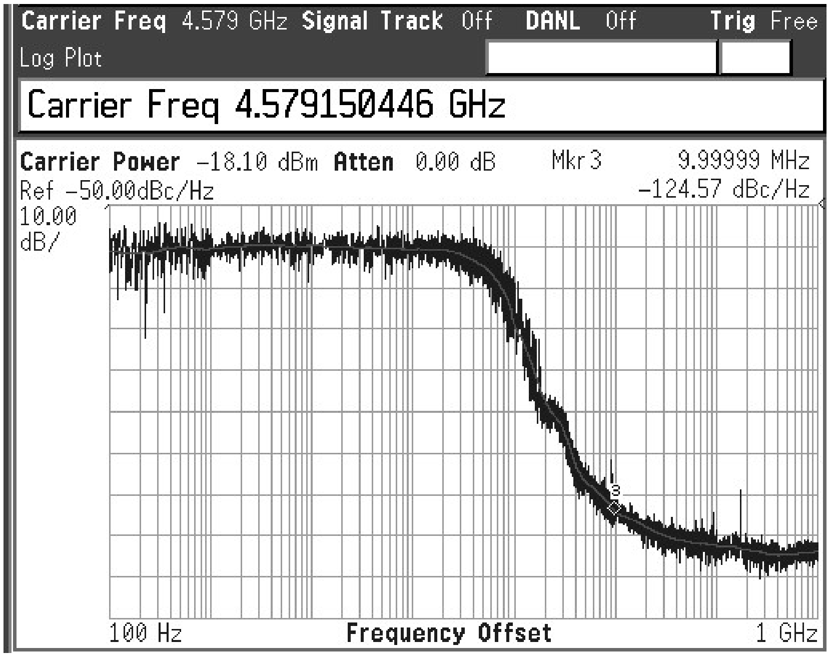
Log Plot

Carrier Freq 5.792909630 GHz

Carrier Power -18.37 dBm Atten 0.00 dB Mkr3 9.99999 MHz

Ref -50.00dBc/Hz -124.04 dBc/Hz

Fig. 91. DCO phase noise spectrum for 5.8GHz operation



Carrier Freq 4.579 GHz Signal Track Off DANL Off Trig Free

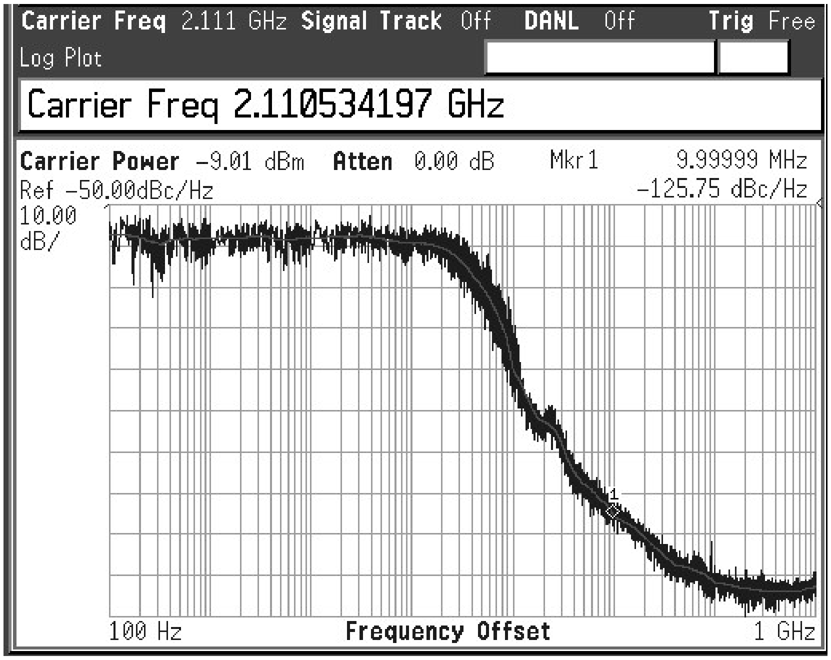
Log Plot

Carrier Freq 4.579150446 GHz

Carrier Power -18.10 dBm Atten 0.00 dB Mkr3 9.99999 MHz

Ref -50.00dBc/Hz -124.57 dBc/Hz

Fig. 92. DCO phase noise spectrum for 4.58GHz operation



Carrier Freq 2.111 GHz Signal Track Off DANL Off Trig Free

Log Plot

Carrier Freq 2.110534197 GHz

Carrier Power -9.01 dBm Atten 0.00 dB Mkr1 9.99999 MHz

Ref -50.00dBc/Hz -125.75 dBc/Hz

Fig. 93. DCO phase noise spectrum for 2.11GHz operation

The measured phase noise performance of the DCO for various operating frequencies are shown in Fig. 89, Fig. 90, Fig. 91, Fig. 92, Fig. 93. Note that at 7.3GHz operating frequency, the DCO demonstrates -123.48dBc/Hz phase noise at 10MHz offset and at 6.24GHz the DCO has -126.11dBc/Hz phase noise at 10MHz offset.

A Figure-of-Merit (FOM) was defined in Chapter V Section C for ring oscillators. Such a FOM is helpful in determining if an oscillator's power consumption, oscillation frequency and phase noise performance as a combination, is competitive. Note that the FOM does not take tuning range into account.

The FOM of the DCO for various operating frequencies are listed in Table XX. It is seen that the FOM of the DCO is better than -161dBc/Hz for the wide range of operation frequencies. In Chapter V Section C Table XIV, a performance comparison for state-of-the-art ring oscillators was provided. It is seen that in measurements, the

DCO demonstrates a better FOM than all of the works listed in Table XIV with the additional benefit of very wide tuning range.

It is demonstrated in [47] that for ring oscillators, the theoretical minimum achievable FOM is -165.2dBc/Hz ($7.33 \times kT$, where k is Boltzmann constant and T is temperature). Then, with an FOM of -164.46 dBc/Hz at 6.24GHz operation, the DCO comes very close to the theoretical achievable limit of FOM for ring oscillators.

Table XX. Measured DCO figure of merit for various frequencies

DCO Frequency	FOM (dBc/Hz)
7.3GHz	-162.38
6.24GHz	-164.46
5.8GHz	-162.42
4.58GHz	-161.8
2.11GHz	-161.3

The DCO measurement results presented in this section demonstrate that the implemented DCO achieves a wide tuning range and demonstrates a good phase noise and performance figure of merit for its wide range of operation frequencies and is therefore suitable for use in the proposed multi-protocol wide range ADPLL.

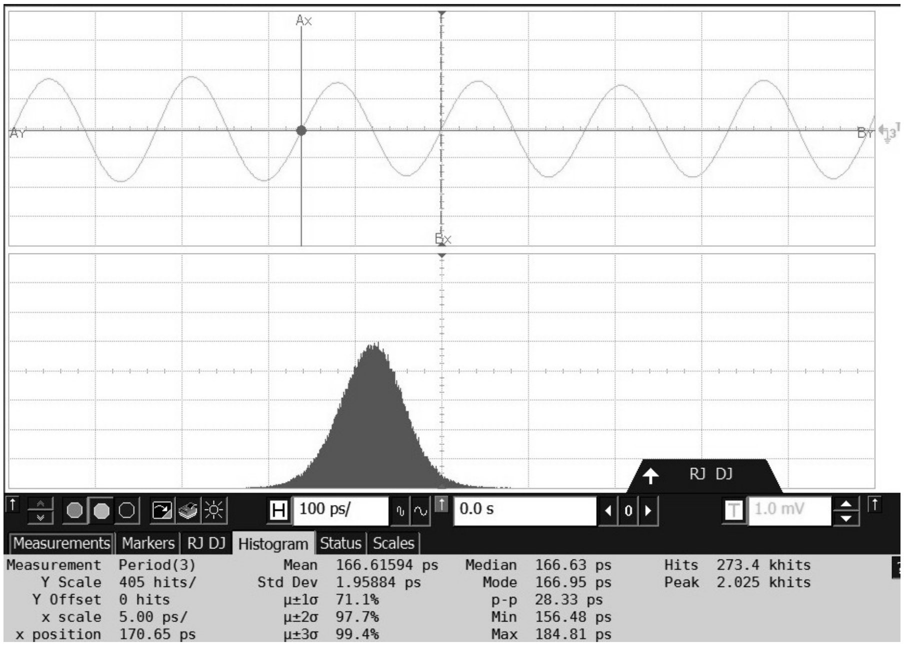
2. ADPLL Measurements

The output range of the ADPLL is determined by the ring DCO tuning range. As discussed in the previous section, the DCO achieves 2.5GHz-7.3GHz range for 1V and 1.95GHz-6.15GHz range for 0.9V supply voltage. In measurements, the high frequency portions of the ADPLL (the DCO, frequency dividers and the DSM) are operated at the same supply level to be able to achieve operation as high as 7.3GHz. However, since the speed of the rest of the DPLL is much lower at the reference frequency, the supply voltage of the loop filter, the DCO controls and TDC (including the GRO) can be set to lower levels such as 0.7V.

Period jitter (rms and peak-to-peak) is an important design metric for digital designers since it conveys information on the clock period variation (rms) and the maximum and minimum clock period (peak-to-peak) that the digital circuits in the system will experience. For a reference frequency of 125MHz and 6GHz operation, the period jitter of the ADPLL is measured to be 1.9ps rms and 28ps peak-to-peak. The power consumption of the whole ADPLL at this setting is 62mW. Fig. 94 demonstrates the measured period histogram at 6GHz operation along with the mean period value, the rms and peak-to-peak jitter.

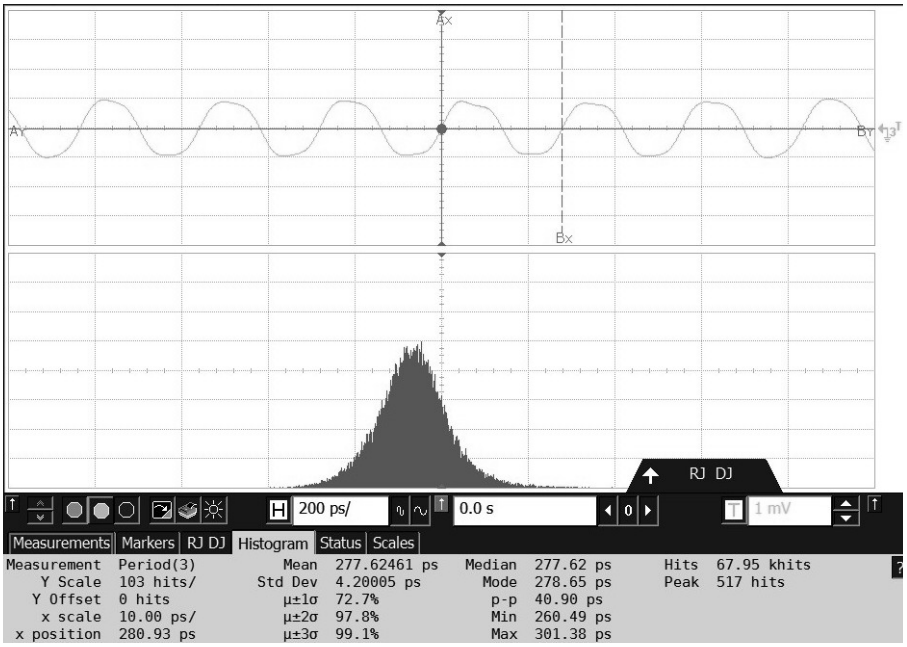
Fig. 95 shows the ADPLL output signal and its period histogram for an output frequency of 3.6GHZ (150MHz reference). The period jitter is measured as 4.2ps rms and 41ps peak-peak. The power consumption of the whole DPLL in this setting is 34mW.

Throughout the wide-range of ADPLL frequencies, for 1V supply (DCO, dividers and DSM), the ADPLL consumes ≈10mW/GHz. When the ADPLL is operated in the low-power mode (all supplies set to 0.72V), at 4GHz operation the ADPLL consumes 32mW (≈8.5mW/GHz). The ADPLL output phase noise spectrum, achieved under this low power setting, is given in Fig. 96. It is seen that the ADPLL loop bandwidth is around 8MHz-10MHz and the in-band noise is -65 dBc/Hz.



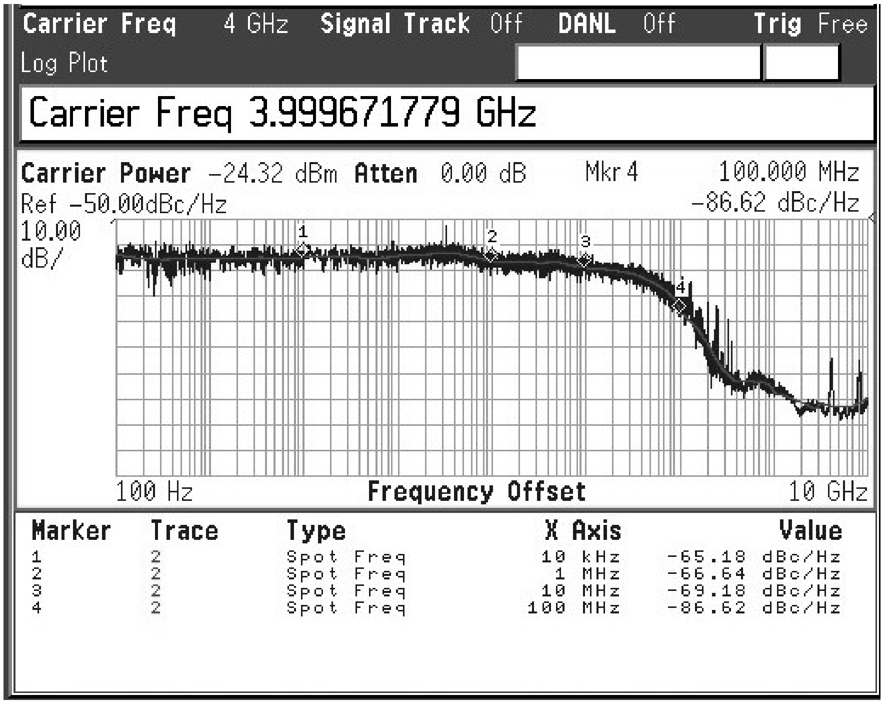
```
Measurement  Period(3)       Mean  166.61594 ps    Median  166.63 ps     Hits  273.4 khits
   Y Scale  405 hits/    Std Dev  1.95884 ps        Mode  166.95 ps     Peak  2.025 khits
  Y Offset  0 hits          μ±1σ  71.1%              p-p  28.33 ps
   x scale  5.00 ps/        μ±2σ  97.7%              Min  156.48 ps
x position  170.65 ps       μ±3σ  99.4%              Max  184.81 ps
```

Fig. 94. ADPLL output and its period histogram at 6GHz operation with 1.9ps rms period jitter



```
Measurement  Period(3)       Mean  277.62461 ps    Median  277.62 ps     Hits  67.95 khits
   Y Scale  103 hits/    Std Dev  4.20005 ps        Mode  278.65 ps     Peak  517 hits
  Y Offset  0 hits          μ±1σ  72.7%              p-p  40.90 ps
   x scale  10.00 ps/       μ±2σ  97.8%              Min  260.49 ps
x position  280.93 ps       μ±3σ  99.1%              Max  301.38 ps
```

Fig. 95. ADPLL output and its period histogram at 3.6GHz operation with 4.2ps rms period jitter



Carrier Freq 4 GHz Signal Track Off DANL Off Trig Free

Log Plot

Carrier Freq 3.999671779 GHz

Carrier Power -24.32 dBm Atten 0.00 dB Mkr4 100.000 MHz

Ref -50.00dBc/Hz -86.62 dBc/Hz

10.00 dB/

Marker	Trace	Type	X Axis	Value
1	2	Spot Freq	10 kHz	-65.18 dBc/Hz
2	2	Spot Freq	1 MHz	-66.64 dBc/Hz
3	2	Spot Freq	10 MHz	-69.18 dBc/Hz
4	2	Spot Freq	100 MHz	-86.62 dBc/Hz

Fig. 96. ADPLL output phase noise spectrum for 4GHz operation

In the previous section, it was seen that the DCO achieves very good phase noise performance (-126.11dBc/Hz at 10MHz offset for 6.24GHz operation). Therefore, we conclude that the ADPLL output noise demonstrated in Fig. 96 is limited by the TDC resolution and the large loop bandwidth (8MHz-10MHz).

The die micrograph is shown in Fig. 97 where the ADPLL active area (excluding decoupling capacitors and testing blocks such as output buffers) is $0.23mm^2$. Layout of the ADPLL active area in detail is shown in Fig. 98. Table XXI summarizes the measurement results of the ADPLL.

Table XXI. ADPLL measured performance summary

Performance Metric	Value
Output Frequency Range	2.5GHz - 7.3GHz ($VDD_{DCO} = 1$V) 1.95GHz - 6.15GHz ($VDD_{DCO} = 0.9$V)
Period Jitter	1.9ps rms (6 GHz operation)
Total Power Consumption	62mW (6 GHz operation) 34 mW (3.6 GHz operation) ≈ 10mW/GHz
Technology	90nm bulk CMOS
Area	0.23 mm^2

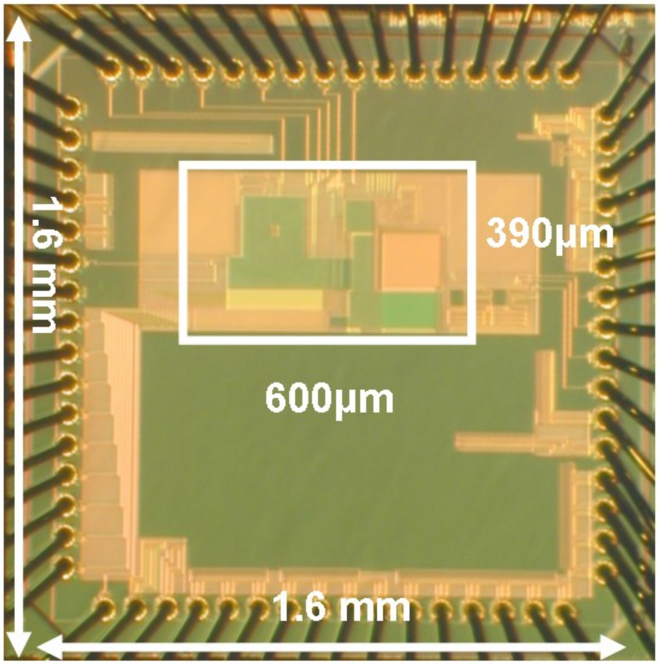


Fig. 97. ADPLL die micrograph

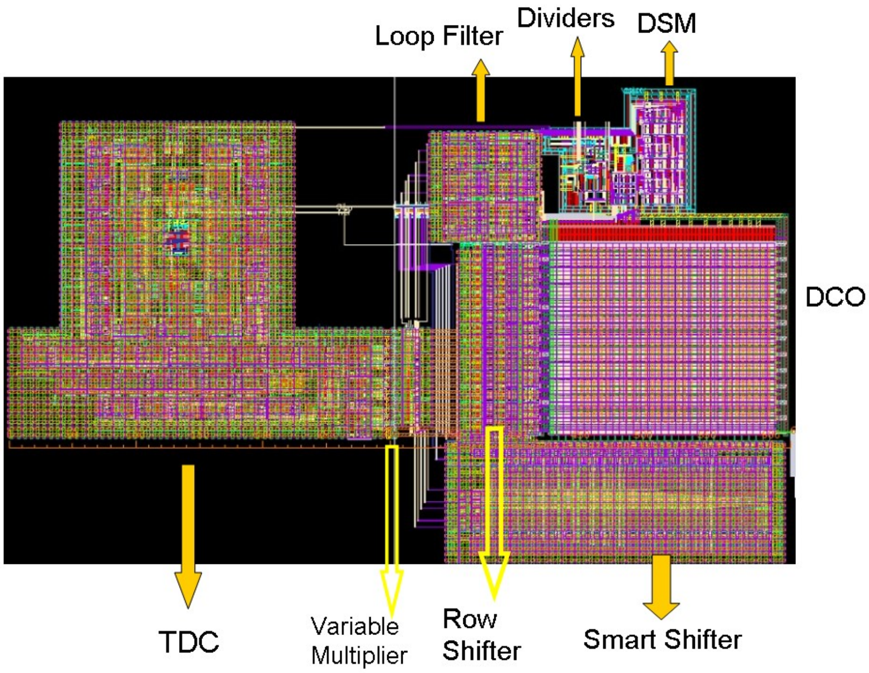


Fig. 98. Layout of the ADPLL active area implemented in 90nm digital CMOS

I. Layout Techniques in the ADPLL

Due to its all digital nature, the ADPLL can be synthesized from hardware-description language based codes, and its layout can be generated through automation tools. However in this prototype, for characterization and analysis purposes, customization is preferred. Therefore, the design and layout of all of the building blocks, including all of the digital circuits, are custom and no automation or synthesis tool is used. Note that the size of the ADPLL system and the digital design and layout required to implement it, is significant. Then, a systematic approach in the digital layout should be followed. In this section, the layout techniques employed in the ADPLL, are discussed.

1. Standard Cell Design

To speed up the layout process for the digital circuits, a custom-designed standard cell library that consists of logic gates (AND, OR, NAND, NOR, XOR, Inverter, etc.) is created for the ADPLL. The key points in the design of a standard cell layout are summarized as follows:

- A uniform height is assigned to the standard library cell layouts. This height is determined by the most complex building block in the standard cell library that would require the largest transistor sizing, hence the largest cell height.
- Each standard cell layout has a boundary layer. Each cell is Design Rule Check (DRC) free as a stand-alone layout and is DRC-free when combined with other standard cell layouts at an upper hierarchy.
- Supply voltage rails are self-routing when multiple standard cell layouts are combined at an upper hierarchy.
- Only metal 1 and metal 2 layers are employed in the standard cells. Therefore, at upper hierarchy levels, signal routing can be done with metal 3 or higher levels of metal, without the concern of a short with the standard cells.

Fig. 99 demonstrates the idea of a standard cell layout. The orange border represents the boundary of the standard cell. When combining multiple standard library cells on upper hierarchies in the design, the smallest distance between the boundaries of different cells is 0. The design of the standard cell ensures that at zero distance from another standard cell, the layout will be DRC-free. The blue vertical lines represent connections such as metal connections between the PMOS and NMOS. Note that the spacing of such connections to the cell boundary is $W\_spacing/2$ where

$W\_spacing$ is the minimum spacing allowed by the technology rules between two such layers (metal, poly, etc.). Note that the supply rail metals exceed the horizontal cell boundaries and should be aligned at the same position in all of the standard cells such that they are self-routing when multiple blocks are connected.

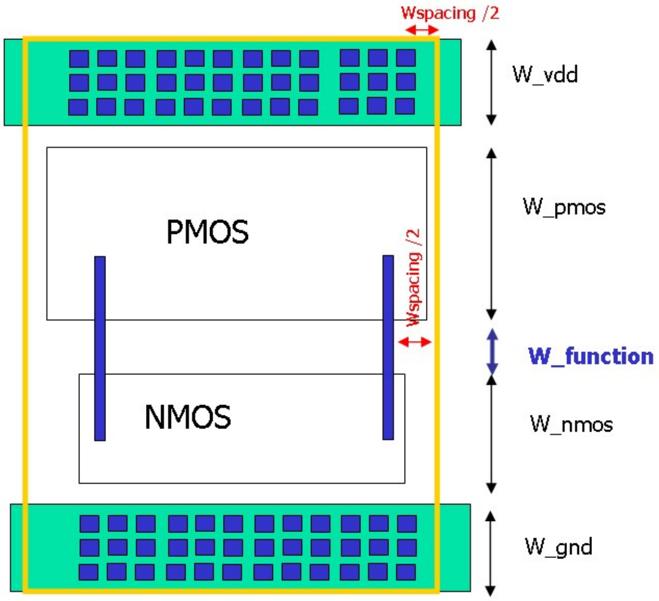


Fig. 99. Demonstration of standard library cell layout

The cell height is determined by the supply rail metal widths, PMOS and NMOS transistor sizes as well as $W\_function$, which is the spacing between PMOS and NMOS, determined by the gate connections and routing within the cell. Since the standard cell height is determined by the most complex function (that would take the largest height) in the library, in some cells, extra space is left between the PMOS and NMOS transistors. If a cell requires very small area when compared to the standard cell height, its supply rails can be widened to include more contacts since bulk and n-well contacts are beneficial to avoid regions without a well-defined potential. Also note that the routing is done only in metal 1 (blue) and metal 2 (yellow) at this lowest hierarchy level.

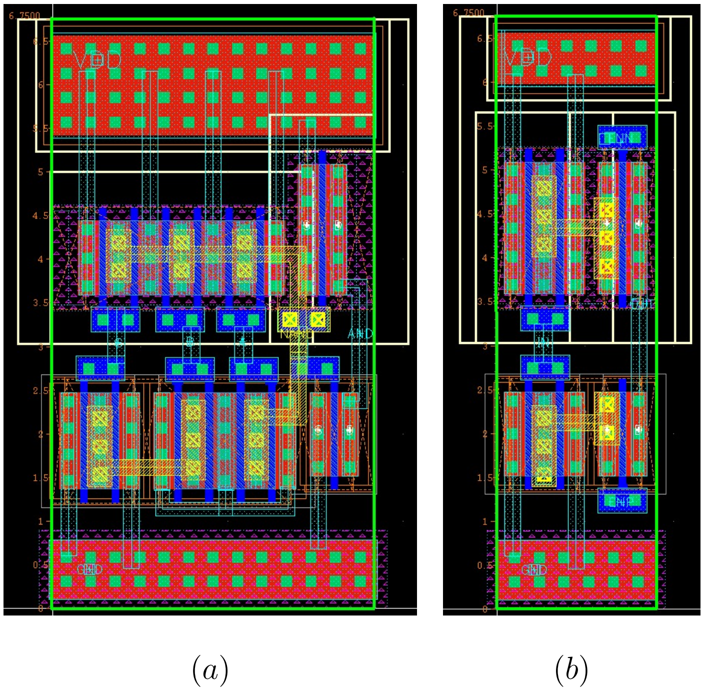


Fig. 100. Standard library cell layout examples Height=6.75 μm (a)a 3-input AND gate (b) a current starved inverter

Fig. 100 shows two example standard cell layouts that were custom designed. The cell boundary is marked with green. The cell height is 6.75 μm. It is seen in Fig. 100 (a) that due to the small PMOS sizes the VDD supply rail is widened to make use of the available space. Fig. 101 shows the layout of a full adder that is built with standard cells from the custom-designed standard-cell library. To create a compact layout, boundary spacing between the cells is set to zero, therefore all of the boundaries exactly overlap and the layout is DRC-free. At this higher level

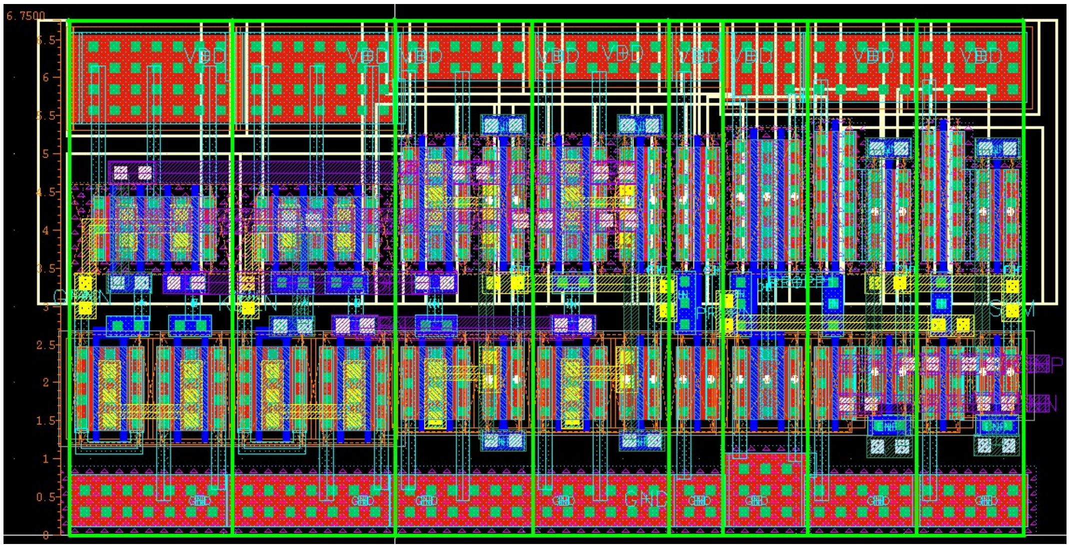


Fig. 101. Layout of a full adder consisting of standard cells

of hierarchy, there is also vertical metal 3 (green) and horizontal metal 4 (purple) routing. The ADPLL is implemented in a 9-metal process. Throughout the layout only metal 7 and lower layers are used in routing since metal 8 and metal 9 are reserved for power supply distribution.

2. Power Supply Distribution

Supply voltage level is very important in determining digital circuits' performance. Voltage drop (also called IxR drop) due to long, thin, high resistivity supply routing might result in degraded performance. This is critical especially in digital circuits that are far from the pad that provides the supply voltage. A supply voltage gradient due to IxR drop might compromise speed and signal integrity and decrease noise margins of the digital circuits.

To avoid IxR drop and maintain supply level integrity, a power supply grid is placed on the digital circuits. In the ADPLL, low-speed digital logic such as DCO controls, variable loop gain, loop filter and TDC processing circuitry share the same supply and therefore are combined under the same supply grid. Fig. 102 shows a detailed view of the supply grid where the top two metal layers metal 8 (vertical) and metal 9 (horizontal) are employed. The metal width for the grid lines are 2.1 μm and both horizontal and vertical grid line spacing is 3 μm. Fig. 103 shows the layout of the loop filter with the supply grid that distributes the power supplies to the building blocks of the loop filter as well as to the rest of the low-speed digital logic in the ADPLL.

The distribution of the power supplies from the pads to the circuits is also a critical layout concern. Fig. 104 shows the chip layout with the decoupling capacitors that are employed for various supplies used in the chip. To minimize the effect of the bonding wire, several pads, and therefore multiple bonding wires, are assigned to the critical supplies such as the DCO (8 pads to VDD_DCO, 6 pads to GND_DCO) and the digital circuitry (4 pads for VDD_Digital and 3 pads to GND_Digital). Fig. 105 shows the detailed view of the supply connection from the pads to the decoupling capacitors. Due to large thick-metal spacing rules, hundreds of thin-metal slices are used to deliver the supply voltages to the decoupling capacitors, which also consist of routing grids that deliver the supply to the circuits which are placed in the middle of the chip.

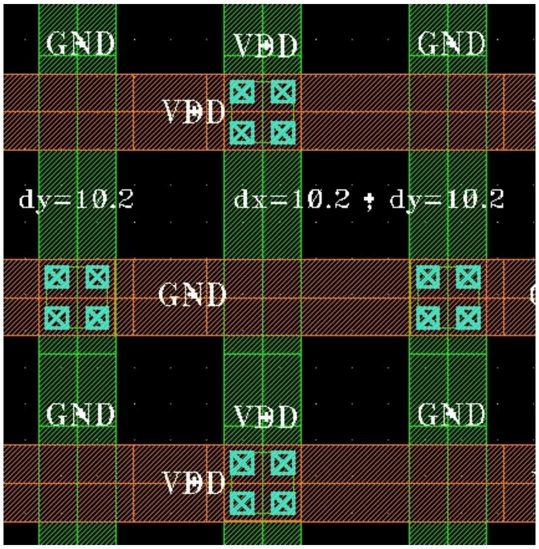


Fig. 102. Supply grid with horizontal metal 9 and vertical metal 8 layers

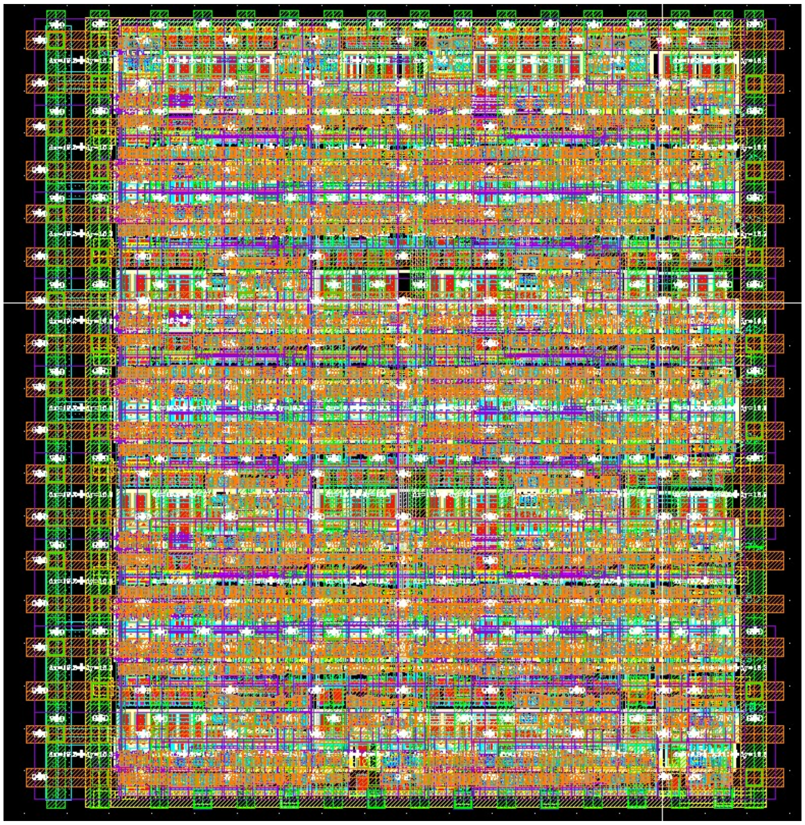


Fig. 103. Digital loop filter layout with power supply grid (89 μm x 94 μm)

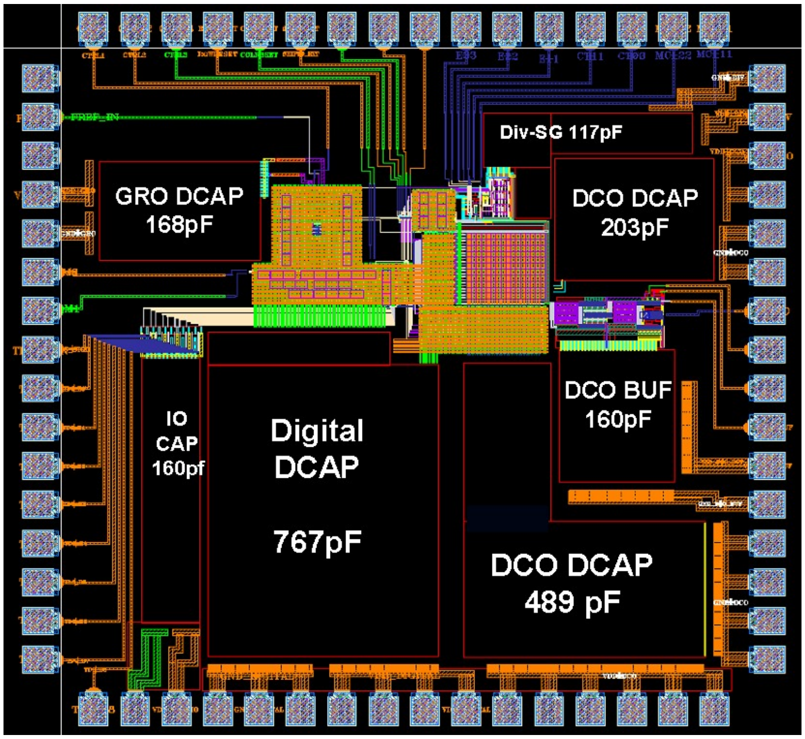


Fig. 104. Layout of the ADPLL chip with pads and decoupling capacitors

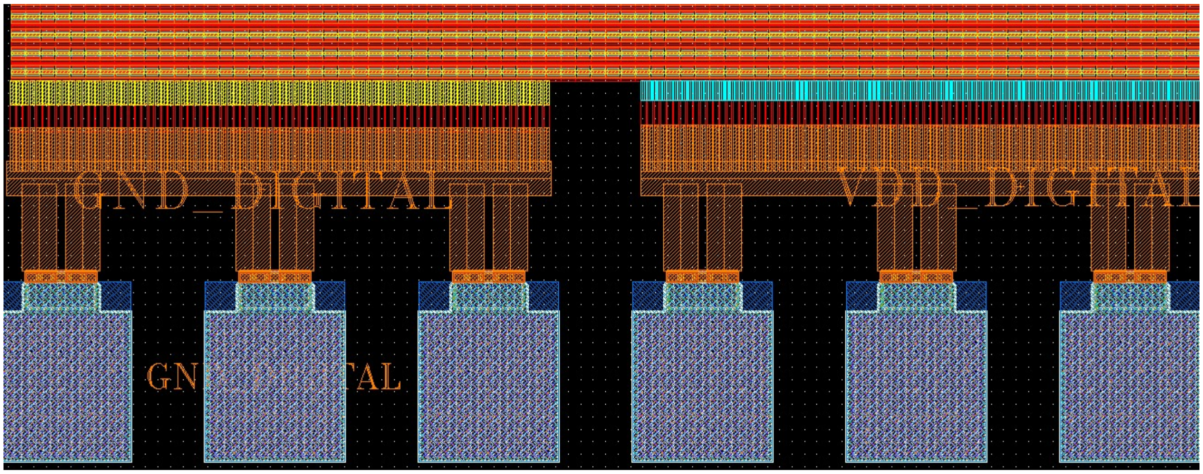


Fig. 105. Detailed view of power supply routing from pads

3. DCO Layout

While a standard cell library was created for the ADPLL, the circuits operating at critical RF frequency such as the DCO and frequency dividers, require custom design of their cells. Due to their noise-sensitive and high-speed nature, the routing of noisy digital signals over critical blocks such as the DCO is avoided. As discussed in Sections B and C, the DCO consists of many inverter units that constitute a three-stage ring oscillator. In total, the DCO has 998 units.

The coarse matrix that is controlled by the smart row/column shifter has 24x32 units which has control logic for row and column enable signals in addition to the oscillator inverter stages. To generate such a large matrix in layout, a self-routing unit-cell approach is followed [64], [65] similar to the standard cell approach that was used for the digital logic of the ADPLL. However, due to its high frequency nature, the DCO unit cell features high levels of metals, since no upper hierarchy routing will be performed over the sensitive DCO cells.

The DCO is a three stage ring oscillator with three output phases A, B and C as shown in Fig. 65. Then, three unit cell layouts are created for the DCO. One whose input is phase A and output is phase B, the second has its input at phase B and output at phase C and the third has its input and output at phases C and A, respectively.

Fig. 106 shows the layout of a single DCO unit cell (input at phase A and output at phase B). Note that the three phases of the DCO, A, B and C are routed in metal 6. The column select control signal is routed vertically in metal 3 while the WHOLEROW_ON and ROW_SEL signals are routed horizontally in metal 4. The supply rails of the DCO are routed in five metal layers stacked on top of each other (metal 1 to metal 5). Note that metal 3 layer in the supply rails are not continuous

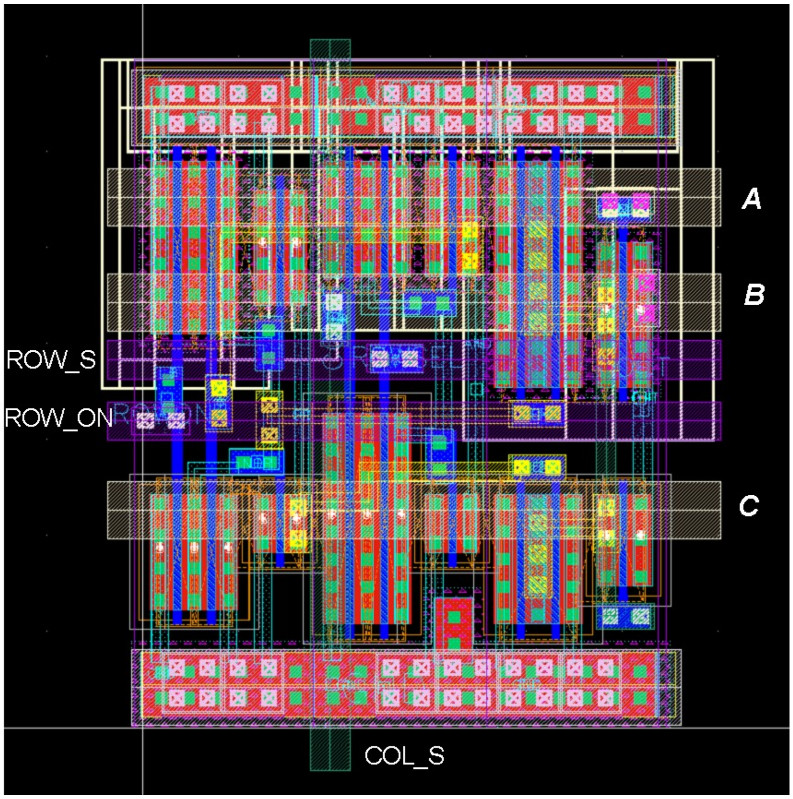


Fig. 106. Layout of a single DCO row/column matrix unit

due to the vertical column select signal that is also routed in metal 3.

Fig. 107 shows a complete 3-stage ring unit, that consists of three of the units shown in Fig. 106, which are connected between the three phases to implement a simple three stage ring DCO. Note that many copies of the layout shown in Fig. 107 are repeated to obtain the row/column DCO matrix.

When placed in the matrix setting, the column and row control signals self-route, all of the cells are DRC-free when combined at their boundaries and the A,B,C high frequency oscillation nodes are self-routing as well. In the multiple rows of the DCO, every other row is horizontally flipped such that the power rails follow a VDD VDD - GND GND - VDD VDD pattern. Therefore, the width of the power rails in the final matrix are twice that of the width shown in Fig. 107.

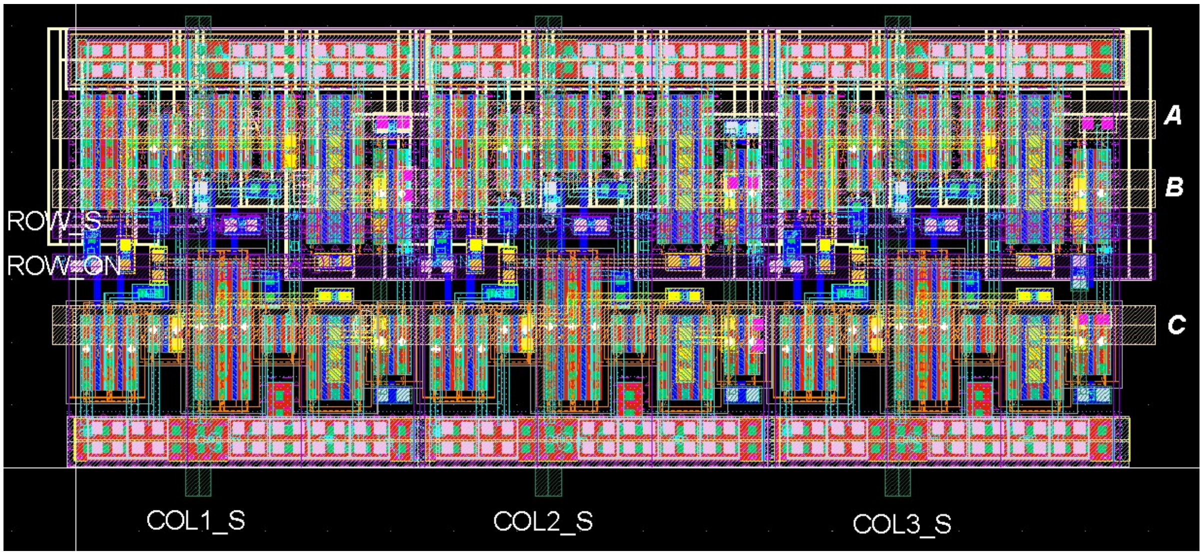


Fig. 107. Layout of the three stage DCO ring unit

To shield the DCO from the noise of the rest of the digital circuitry, the DCO has its own separate power supply grid and it is placed in its own guard ring.

CHAPTER VIII

CONCLUSIONS

A. Summary

In this dissertation, frequency synthesizers, their design concerns and building blocks for wireless and wireline applications, were discussed. Among the focus points was low power consumption, particularly through the reduction of the power consumption of the high speed frequency dividers in frequency synthesizers for wireless systems. Another focus point was the analysis and implementation of all digital PLLs to implement clock generators in wireline systems. The conclusions of this dissertation are summarized below.

For a wireless transceiver application, the implementation of a ZigBee frequency synthesizer with TSPC frequency dividers, with a focus on low power consumption, was presented. It was observed that TSPC divider power consumption is lower than its CML alternative, but is still high due to large driving-buffer power.

DCVSL based delay cells have been analyzed for RF frequency-divider and ring oscillator applications. We have presented a closed-form delay model for DCVSL inverters that demonstrates 8% worst case accuracy for various transistor sizing ratios and for two different technologies (0.13μm and 0.18μm CMOS). The inherent speed bottleneck of DCVSL structures that cause $\tau_{PLH} > \tau_{PHL}$ have been addressed, and a solution (DCVSL-R) that reduces τ_{PLH} and the total propagation delay of the circuit, offered.

The proposed speed-enhanced DCVSL-R circuits have implemented the RF dual-modulus prescaler of a low-power frequency synthesizer that satisfies ZigBee specifications, in 0.18μm technology. The proposed dual-modulus prescaler of this synthesizer consumes the lowest power (0.8mW), 40% less, among similar dividers in literature, that employ different circuit techniques such as CML and TSPC. The RF buffer that drives the DMP consumes only 0.27mW due to the low clock input capacitance of the DCVSL-R circuit. The proposed circuit proves to be a good candidate to replace existing RF frequency-divider circuits. It reduces the power consumption of the infamously power-hungry frequency dividers of frequency synthesizers while providing a low-cost, differential, low-input-capacitance and high-speed solution.

Two ring-oscillator-based VCOs employing DCVSL based cells and proposed DCVSL-R cells have been implemented, and measured results have been compared. At the same operating frequency of 2.4GHz for the same phase noise, the proposed DCVSL-R based oscillator consumes 30% less power than its standard counterpart. When compared to other state-of-the-art ring oscillators, the proposed oscillator performs with a good figure of merit, and small area.

In this dissertation, a wide-band all digital PLL that can serve as a multi-protocol PLL in wireline applications, was proposed. A new loop architecture that minimizes overall loop complexity was presented where DACs and thermometer converters are avoided and the digital nature of the TDC output is exploited to implement coarse and fine control paths while avoiding lock-detection and explicit dual loop architectures. A variable loop gain enables stability over wide range of operating frequencies.

A proposed digital bidirectional smart shifter, which presents a simple method to perform two-dimensional row/column matrix shifting at a single loop update cycle, controls the coarse DCO frequency. A GRO based digital TDC digitizes the phase error between reference and divider outputs and also serves as the frequency detector for the wide-range loop. A ring oscillator based DCO not only achieves a very wide tuning range but also maintains a good phase noise and a performance figure of merit (that entails noise/frequency/power trade off) throughout the entire operating range. Carry-skip adder/subtractors with Manchester Carry Chains are employed in the digital processing circuitries of the loop, to deliver simple design with good delay performance to operate at the loop frequency.

The ADPLL was implemented in a 90nm CMOS technology. It has a measured operating range from 2.5GHz to 7.3GHz for 1V nominal DCO supply voltage or from 1.95GHz to 6.15GHz for a low-voltage operation of 0.9V DCO supply while using only 0.7V supply for the digital circuitry in both cases. The ADPLL is purely digital without any R/L/C components and is therefore easily scalable for migration into smaller technologies and synthesizable with hardware description languages in future prototypes.

B. Contribution and Impact

The contributions of this dissertation are summarized as follows:

- Discussion on the performance and power consumption of frequency synthesizers and demonstration of the significance of the frequency divider power in the total power consumption of a frequency synthesizer.

- Analysis and discussion of frequency dividers and their power and performance cost in PLLs.

- Proposing DCVSL logic family as a candidate for use in the high-frequency dividers of a PLL to reduce power consumption while maintaining high speed capability.

- An analysis of delay in DCVSL circuits and a closed-form model to estimate the value of propagation delay in DCVSL circuits.

- Proposal of an improved circuit (DCVSL-R) with advantages of symmetric τ_{PLH} and τ_{PHL}, lower total delay, and small clock capacitance.
- The first use of a DCVSL style design (DCVSL-R) for high frequency dividers of a synthesizer in literature, to improve PLL performance (reduce the power consumption of the dividers and their driving buffers and therefore reduce the total PLL power significantly, and to provide symmetrical loading to VCO).
- Replacement of DCVSL delay cells in ring oscillators, with DCVSL-R cells to achieve power reduction for a desired oscillation frequency, or speed improvement for a desired power budget, without sacrificing phase noise.
- Validation of the above points via measurements of two 0.18μm synthesizers and 0.13μm ring oscillators.
- A discussion on all digital PLLs, their loop analysis and a discussion on the current state-of-the-art ADPLLs.
- The system level discussion and building-block level design and layout information of a proposed wide-range all digital PLL that eliminates DACs, or non-scalable R/L/C components and therefore removes all analog design concerns.
- The demonstration of the operation of the proposed ADPLL, through the measurements of a prototype implemented in 90nm digital CMOS technology.

The topics covered in this dissertation impact a wide area of applications in integrated circuit design and in communication systems. Phase locked loops are an essential part of communication systems, for frequency synthesis and for clock generation. Power consumption of the PLL and its individual building blocks, is an important concern and a crucial factor in the design of wireless transceivers where battery life is a critical performance metric, as well as in wireline systems, which often employ several PLLs for the generation of various clock domains.

The impact of all digital PLLs in communication systems is significant due to the rapid reduction in technology feature sizes, that challenges analog designers while improving the performance of digital circuits. Keeping the PLL as an analog building block diminishes its compatibility with its digital-intense environment, while moving to an all digital design enables a new arena of design possibilities such as high level of programmability. In literature, the recent works on ADPLLs focus on narrow range solutions that target a single standard and employ DACs or passive R/L/C components. The proposed ADPLL on the other hand, achieves wide range and will serve as an all digital, multi-protocol, programmable and flexible solution.

To conclude, the proposed analysis, techniques and advancements in PLLs, (on system level and in building blocks such as frequency dividers and ring oscillators), have a direct impact on the performance improvement of wireless and wireline systems.

REFERENCES

[1] C. Lam and B. Razavi, “A 2.6-ghz/5.2-ghz frequency synthesizer in 0.4- mu;m cmos technology,” *IEEE J. Solid-State Circuits*, vol. 35, no. 5, pp. 788 –794, May 2000.

[2] C.-M. Hung and K. O, “A fully integrated 1.5-v 5.5-ghz cmos phase-locked loop,” *IEEE J. Solid-State Circuits*, vol. 37, no. 4, pp. 521 –525, Apr. 2002.

[3] H. Rategh, H. Samavati, and T. Lee, “A cmos frequency synthesizer with an injection-locked frequency divider for a 5-ghz wireless lan receiver,” *IEEE J. Solid-State Circuits*, vol. 35, no. 5, pp. 780 –787, May 2000.

[4] R. Srinivasan, D. Z. Turker, S. W. Park, and E. Sanchez-Sinencio, “A low-power frequency synthesizer with quadrature signal generation for 2.4 ghz zigbee transceiver applications,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, May 2007, pp. 429 –432.

[5] S. Pellerano, S. Levantino, C. Samori, and A. Lacaita, “A 13.5-mw 5-ghz frequency synthesizer with dynamic-logic frequency divider,” *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 378 – 383, Feb. 2004.

[6] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE 802.15.4 Standard, 2003.

[7] D. W. Kang and Y.-B. Kim, “Design of enhanced differential cascode voltage switch logic (edcvsl) circuits for high fan-in gate,” in *Proc. IEEE Ann. Int. ASIC/SOC Conference*, Sept. 2002, pp. 309 – 313.

[8] M. Elrabaa, “A new static differential cmos logic with superior low power performance,” in *Proc. IEEE Int. Conf. Electronics, Circuits and Systems (ICECS)3*, vol. 2, Dec. 2003, pp. 810 – 813.

[9] S. Safaric and K. Malaric, “Zigbee wireless standard,” in *48th International Symposium ELMAR focused on Multimedia Signal Processing and Communications*, Jun. 2006, pp. 259 –262.

[10] A. Emira, A. Valdes-Garcia, B. Xia, A. Mohieldin, A. Valero-Lopez, S. Moon, C. Xin, and E. Sanchez-Sinencio, “Chameleon: a dual-mode 802.11b/bluetooth receiver system design,” *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 53, no. 5, pp. 992 – 1003, May. 2006.

[11] A. Y. Valero-Lopez, “Design of frequency synthesizers for short-range wireless systems,” Ph.D. dissertation, Texas A&M University, College Station, Texas, 2004.

[12] A. Valdes-Garcia, C. Mishra, F. Bahmani, J. Silva-Martinez, and E. Sanchez-Sinencio, “An 11-band 3 ndash;10 ghz receiver in sige bicmos for multiband ofdm uwb communication,” *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 935 –948, Apr. 2007.

[13] L. Leung Lai Kan, D. Lau, S. Lou, A. Ng, R. Wang, G.-K. Wong, P. Wu, H. Zheng, V.-L. Cheung, and H. Luong, “A 1-v 86-mw-rx 53-mw-tx single-chip cmos transceiver for wlan ieee 802.11a,” *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 1986 –1998, Sep. 2007.

[14] B. Razavi, *RF Microelectronics.* Upper Sadle River, NJ: Prentice Hall, 1998.

[15] R. Srinivasan, “Design and implementation of a frequency synthesizer for an ieee

802.15.4/zigbee transceiver," MS in Electrical Engineering, Texas A&M University, College Station, Texas, May 2006.

[16] F. Hussien, "Ultra low power ieee 802.15.4/zigbee compliant transceiver," Ph.D. dissertation, Texas A&M University, College Station, Texas, 2009.

[17] S. T. Moon, "Design of high-performance frequency synthesizers in communication systems," Ph.D. dissertation, Texas A&M University, College Station, Texas, 2005.

[18] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits.* New York, NY: Cambridge University Press, 1998.

[19] B. Razavi, "A study of injection locking and pulling in oscillators," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1415 – 1424, Sep. 2004.

[20] K. Ogata, *Modern Control Engineering.* Upper Sadle River, NJ: Prentice Hall, 2002.

[21] K. Shu and E. Sánchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design.* New York, NY: Springer, 2005.

[22] B. Razavi, *Design of Integrated Circuits for Optical Communications.* New York, NY: McGraw-Hill, 2003.

[23] F. M. Gardner, *Phaselock Techniques.* Hoboken, NJ: John Wiley & Sons, 2005.

[24] F. Gardner, "Charge-pump phase-lock loops," *IEEE Trans. Communications*, vol. 28, no. 11, pp. 1849 – 1858, Nov. 1980.

[25] K. Shu, E. Sanchez-Sinencio, J. Silva-Martinez, and S. Embabi, "A 2.4-ghz monolithic fractional-n frequency synthesizer with robust phase-switching prescaler

and loop capacitance multiplier," *IEEE J. Solid-State Circuits,*, vol. 38, no. 6, pp. 866 – 874, Jun. 2003.

[26] M. Alioto and G.Palumbo, "Design strategies for source coupled logic gates," *IEEE Trans. Circuits and Systems I*, vol. 50, no. 5, pp. 640 –654, May 2003.

[27] U. Seckin and C.-K. K. Yang, "A comprehensive delay model for cmos cml circuits," *IEEE Trans. Circuits and Systems I*, vol. 55, no. 9, pp. 2608 –2618, Oct. 2008.

[28] S. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits.* New York, NY: McGraw Hill, 1999.

[29] J. Yuan and C. Svensson, "New single-clock cmos latches and flipflops with improved speed and power savings," *IEEE J. Solid-State Circuits*, vol. 32, no. 1, pp. 62 –69, Jan. 1997.

[30] Q. Huang and R. Rogenmoser, "Speed optimization of edge-triggered cmos circuits for gigahertz single-phase clocks," *IEEE J. Solid-State Circuits*, vol. 31, no. 3, pp. 456 –465, Mar. 1996.

[31] Y. B. Choi and X. Yuan, "A 3.5-mw 2.45-ghz frequency synthesizer in 0.18μm cmos," in *Proc. IEEE Int. Symp. Radio Frequency Integration Technology*, Dec. 2009, pp. 187 –190.

[32] L. Heller, W. Griffin, J. Davis, and N. Thoma, "Cascode voltage switch logic: a differential cmos logic family," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, vol. XXVII, Feb. 1984, pp. 16 – 17.

[33] L. Pfennings, W. Mol, J. Bastiaens, and J. van Dijk, "Differential split-level cmos logic for subnanosecond speeds," *IEEE J. Solid-State Circuits*, vol. 20, no. 5, pp.

1050 – 1055, Oct. 1985.

[34] P. Ng, P. Balsara, and D. Steiss, “Performance of cmos differential circuits,” *IEEE J. Solid-State Circuits*, vol. 31, no. 6, pp. 841 –846, Jun. 1996.

[35] K. Chu and D. Pulfrey, “A comparison of cmos circuit techniques: differential cascode voltage switch logic versus conventional logic,” *IEEE J. Solid-State Circuits*, vol. 22, no. 4, pp. 528 – 532, Aug. 1987.

[36] D. Somasekhar and K. Roy, “Differential current switch logic: a low power dcvs logic family,” *IEEE J. Solid-State Circuits*, vol. 31, no. 7, pp. 981 –991, Jul. 1996.

[37] T. Sakurai and A. Newton, “Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas,” *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584 –594, Apr. 1990.

[38] L. Bisdounis, S. Nikolaidis, and O. Loufopavlou, “Propagation delay and short-circuit power dissipation modeling of the cmos inverter,” *IEEE Trans. Circuits and Systems I*, vol. 45, no. 3, pp. 259 –270, Mar. 1998.

[39] A. Hamoui and N. Rumin, “An analytical model for current, delay, and power analysis of submicron cmos logic circuits,” *IEEE Trans. Circuits and Systems II*, vol. 47, no. 10, pp. 999 –1007, Oct. 2000.

[40] D. Auvergne, J. Daga, and M. Rezzoug, “Signal transition time effect on cmos delay evaluation,” *IEEE Trans. Circuits and Systems I*, vol. 47, no. 9, pp. 1362 – 1369, Sep. 2000.

[41] F. Ducati, M. Pifferi, and M. Borgarino, “Self-oscillation free 0.35μm si/sige bicmos x-band digital frequency divider,” *IEEE Microwave and Wireless Components Letters*, vol. 18, no. 7, pp. 473 –475, July 2008.

[42] I. S.-C. Lu, N. Weste, and S. Parameswaran, “A power-efficient 5.6-ghz process-compensated cmos frequency divider,” *IEEE Trans. Circuits and Systems II: Express Briefs,*, vol. 54, no. 4, pp. 323 –327, Apr. 2007.

[43] J. Lee and B. Kim, “A low-noise fast-lock phase-locked loop with adaptive bandwidth control,” *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1137 –1145, Aug. 2000.

[44] J. Maneatis and M. Horowitz, “Precise delay generation using coupled oscillators,” *IEEE J. Solid-State Circuits*, vol. 28, no. 12, pp. 1273 –1282, Dec. 1993.

[45] T. Wu, K. Mayaram, and U.-K. Moon, “An on-chip calibration technique for reducing supply voltage sensitivity in ring oscillators,” *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 775 –783, Apr. 2007.

[46] M. Tiebout, “Low-power low-phase-noise differentially tuned quadrature vco design in standard cmos,” *IEEE J. Solid-State Circuits*, vol. 36, no. 7, pp. 1018 –1024, July 2001.

[47] R. Navid, T. H. Lee, and R. W. Dutton, “Minimum achievable phase noise of rc oscillators,” *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 630 – 637, Mar. 2005.

[48] Z. Shu, K. L. Lee, and B. Leung, “A 2.4-ghz ring-oscillator-based cmos frequency synthesizer with a fractional divider dual-pll architecture,” *IEEE J. Solid-State Circuits*, vol. 39, no. 3, pp. 452 – 462, Mar. 2004.

[49] S. W. Park and E. Sanchez-Sinencio, “Rf oscillator based on a passive rc bandpass filter,” *IEEE J. Solid-State Circuits*, vol. 44, no. 11, pp. 3092 –3101, Nov. 2009.

[50] A. Elshazly and K. Sharaf, “2 ghz 1v sub-mw, fully integrated pll for clock recovery applications using self-skewing,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, May 2006, pp. 3213 – 3216.

[51] W. Rahajandraibe, L. Zaid, V. de Beaupre, and G. Bas, “Temperature compensated 2.45 ghz ring oscillator with double frequency control,” in *IEEE Radio Frequency Integrated Circuits (RFIC) Symp. Dig.*, June 2007, pp. 409 –412.

[52] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, “An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors,” *IEEE J. Solid-State Circuits*, vol. 30, no. 4, pp. 412 –422, Apr. 1995.

[53] C.-C. Chung and C.-Y. Lee, “An all-digital phase-locked loop for high-speed clock generation,” *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 347 – 351, Feb. 2003.

[54] T. Olsson and P. Nilsson, “A digitally controlled pll for soc applications,” *IEEE J. Solid-State Circuits*, vol. 39, no. 5, pp. 751 – 760, May. 2004.

[55] R. Staszewski, K. Muhammad, D. Leipold, C.-M. Hung, Y.-C. Ho, J. Wallberg, C. Fernando, K. Maggio, R. Staszewski, T. Jung, J. Koh, S. John, I. Y. Deng, V. Sarda, O. Moreira-Tamayo, V. Mayega, R. Katz, O. Friedman, O. Eliezer, E. de Obaldia, and P. Balsara, “All-digital tx frequency synthesizer and discrete-time receiver for bluetooth radio in 130-nm cmos,” *IEEE J. Solid-State Circuits*, vol. 39, no. 12, pp. 2278 – 2291, Dec. 2004.

[56] C.-M. Hsu, M. Straayer, and M. Perrott, “A low-noise wide-bw 3.6-ghz digital $\delta\sigma$ fractional-n frequency synthesizer with a noise-shaping time-to-digital converter and quantization noise cancellation,” *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2776 –2786, Dec. 2008.

[57] L. Xu, S. Lindfors, K. Stadius, and J. Ryynanen, “A 2.4-ghz low-power all-digital phase-locked loop,” *IEEE J. Solid-State Circuits*, vol. 45, no. 8, pp. 1513 –1521, Aug. 2010.

[58] R. Tonietto, E. Zuffetti, R. Castello, and I. Bietti, “A 3mhz bandwidth low noise rf all digital pll with 12ps resolution time to digital converter,” in *European Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2006, pp. 150 –153.

[59] H.-H. Chang, P.-Y. Wang, J.-H. Zhan, and B.-Y. Hsieh, “A fractional spur-free adpll with loop-gain calibration and phase-noise cancellation for gsm/gprs/edge,” in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2008, pp. 200 –606.

[60] E. Temporiti, C. Weltin-Wu, D. Baldi, R. Tonietto, and F. Svelto, “A 3 ghz fractional all-digital pll with a 1.8 mhz bandwidth implementing spur reduction techniques,” *IEEE J. Solid-State Circuits*, vol. 44, no. 3, pp. 824 –834, Mar. 2009.

[61] S.-Y. Yang and W.-Z. Chen, “A 7.1mw 10ghz all-digital frequency synthesizer with dynamically reconfigurable digital loop filter in 90nm cmos,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2009, pp. 90 –91,91a.

[62] N. Da Dalt, E. Thaller, P. Gregorius, and L. Gazsi, “A compact triple-band low-jitter digital lc pll with programmable coil in 130-nm cmos,” *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1482 – 1490, Jul. 2005.

[63] M. Lee, M. Heidari, and A. Abidi, “A low-noise wideband digital phase-locked loop based on a coarse fine time-to-digital converter with subpicosecond resolution,” *IEEE J. Solid-State Circuits*, vol. 44, no. 10, pp. 2808 –2816, Oct. 2009.

[64] A. Rylyakov, J. Tierno, D. Z. Turker, J.-O. Plouchart, H. Ainspan, and D. Friedman, “A modular all-digital pll architecture enabling both 1-to-2ghz and 24-to-

32ghz operation in 65nm cmos," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2008, pp. 516 –632.

[65] J. Tierno, A. Rylyakov, and D. Friedman, "A wide power supply range, wide tuning range, all static cmos all digital pll in 65 nm soi," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 42 –51, Jan. 2008.

[66] A. Rylyakov, J. Tierno, G. English, M. Sperling, and D. Friedman, "A wide tuning range (1 ghz-to-15 ghz) fractional-n all-digital pll in 45nm soi," in *IEEE Custom Integrated Circuits Conf. (CICC)*, Sep. 2008, pp. 431 –434.

[67] V. Kratyuk, P. Hanumolu, K. Ok, U.-K. Moon, and K. Mayaram, "A digital pll with a stochastic time-to-digital converter," *IEEE Trans. Circuits and Systems I*, vol. 56, no. 8, pp. 1612 –1621, Aug. 2009.

[68] V. Kratyuk, P. Hanumolu, K. Mayaram, and U.-K. Moon, "A 0.6ghz to 2ghz digital pll with wide tracking range," in *IEEE Custom Integrated Circuits Conf. (CICC)*, Sep. 2007, pp. 305 –308.

[69] V. Kratyuk, P. Hanumolu, U.-K. Moon, and K. Mayaram, "A design procedure for all-digital phase-locked loops based on a charge-pump phase-locked-loop analogy," *IEEE Trans. Circuits and Systems II*, vol. 54, no. 3, pp. 247 –251, Mar. 2007.

[70] R. B. Staszewski and P. T. Balsara, *All-Digital Frequency Synthesizer in Deep-Submicron CMOS.* Hoboken, NJ: John Wiley & Sons, 2006.

[71] R. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. Balsara, "1.3 v 20 ps time-to-digital converter for frequency synthesis in 90-nm cmos," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 53, no. 3, pp. 220 – 224, Mar. 2006.

[72] M. Straayer and M. Perrott, “A multi-path gated ring oscillator tdc with first-order noise shaping,” *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1089 –1098, Apr. 2009.

[73] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing.* Upper Sadle River, NJ: Prentice Hall, 2006.

[74] J. Hein and J. Scott, “z-domain model for discrete-time pll’s,” *IEEE Trans. Circuits and Systems*, vol. 35, no. 11, pp. 1393 –1400, Nov. 1988.

[75] A. Loke, R. Barnes, T. Wee, M. Oshima, C. Moore, R. Kennedy, and M. Gilsdorf, “A versatile 90-nm cmos charge-pump pll for serdes transmitter clocking,” *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1894 –1907, Aug. 2006.

[76] M. Ferriss and M. Flynn, “A 14 mw fractional-n pll modulator with a digital phase detector and frequency switching scheme,” *IEEE J. Solid-State Circuits*, vol. 43, no. 11, pp. 2464 –2471, Nov. 2008.

[77] C. Nagendra, M. Irwin, and R. Owens, “Area-time-power tradeoffs in parallel adders,” *IEEE Trans. Circuits and Systems II*, vol. 43, no. 10, pp. 689 –702, Oct. 1996.

[78] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design, A Systems Perspective.* Boston, MA: Addison Wesley, 1994.

[79] M. Lehman and N. Burla, “Skip techniques for high-speed carry-propagation in binary arithmetic units,” *IRE Trans. Electronic Computers*, vol. EC-10, no. 4, pp. 691 –698, Dec. 1961.

[80] M. M. Mano and C. R. Kime, *Logic and Computer Design Fundamentals.* Upper Sadle River, NJ: Pearson Prentice Hall, 2004.

[81] S. W. Park, "Oscillator architectures and enhanced frequency synthesizer," Ph.D. dissertation, Texas A&M University, College Station, Texas, 2009.

APPENDIX A

A DESIGN PROCEDURE FOR CHARGE PUMP BASED PLLS

The second order approximation of linear phase analysis of charge pump based PLLs was provided in Chapter II. Note that the charge-pump PLL that was demonstrated in Chapter II Fig. 2 that employs a loop filter shown in Fig. 3, is a third-order system. The closed loop transfer function of the third order PLL can be derived as:

$$H_{CL\_PLL}(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{(K_{LOOP}w_p \times N)(s+w_z)}{s^3 + w_p s^2 + (K_{LOOP}w_p)\, s + K_{LOOP}w_p w_z} \tag{A.1}$$

where

$$K_{LOOP} = \frac{K_{VCO}I_{CP}R}{2\pi N} \tag{A.2}$$

As shown in Chapter II, the third order system of (A.1) can be approximated as a second order system, with the assumption that w_p is placed much further than the loop bandwidth w_c and therefore the frequencies of interest. Then, the closed loop transfer function was derived as:

$$H_{CL\_PLL}(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{(K_{LOOP} \times N)(s+w_z)}{s^2 + (K_{LOOP})\, s + K_{LOOP}w_z} \tag{A.3}$$

which can also be represented as a standard second order system transfer function as follows:

$$H_{CL\_PLL}(s) = \frac{(2\xi w_n \times N)(s+w_z)}{s^2 + 2\xi w_n s + w_n^2} \tag{A.4}$$

Using the loop transfer function in (A.3) and (A.4), the important loop parameter expressions are derived and listed in Table IV. The important relations between the loop control parameters are given in Table V and the loop filter parameters are given in (2.5) and (2.7).

Based on these equations and expressions, a design procedure for the second order approximated third order PLLs can be determined. In this appendix, a procedure that shows how to determine the design parameters of a charge pump based PLL and its building blocks, will be provided.

Design Procedure

Step 1: Reference Frequency and Division Ratio

The choice of the reference frequency is very important in a frequency synthesizer. Most wireless standards support operation at various channels, spaced by a channel spacing frequency of F_{CH}. The PLL output frequency is determined by the feedback divider's division ratio and the reference frequency such that:

$$FOUT = FREF \times N \tag{A.5}$$

In integer-N based PLLs the reference frequency should be a common divisor of the channel spacing F_{SP} and the channel center frequency F_O with the maximum reference frequency being the greatest common divisor:

$$FREF_{MAX} = GCD(F_O, F_{SP}) \tag{A.6}$$

In frequency synthesizers designed for wireless standards, a pulse swallow divider is commonly used where the division ratio can be incremented in steps of 1 such as the divider used in the ZigBee synthesizer of Chapter II:

$$N = 481, 482, ..., 495, 496 \tag{A.7}$$

In this case, the common choice of reference frequency is the channel spacing frequency since the PLL output frequencies increment by one $FREF$. However, an additional divider can also be used as a prescaler (often a divide-by-2) circuit, before the pulse swallow divider to reduce the operating frequency of the dual-modulus prescaler of the pulse-swallow divider as shown in Fig. 108. Then the reference frequency is given by:

$$FREF = \frac{F_{SP}}{M} \tag{A.8}$$

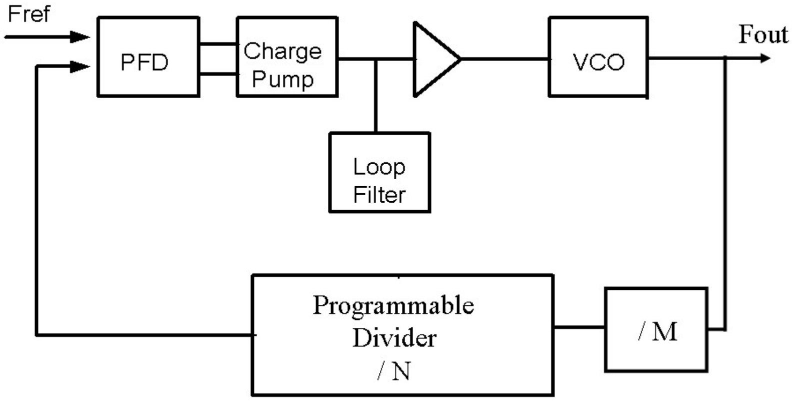


Fig. 108. Block diagram of a PLL with a prescaling divider before the pulse-swallow divider

Note that the continuous time linear analysis of PLLs, discussed in Chapter II, is based on an approximation since in reality PLLs are sampled systems. In frequency lock, which is assumed for the linear analysis of the loop, the PFD samples its inputs at the rate of the reference frequency. Then, in addition to the implementation of the frequency divider, the channel spacing and center frequency requirements of a wireless standard that are discussed above, the loop stability is also affected by the reference frequency.

The continuous time approximation of PLLs and the stability analysis that is based on this, is valid if the closed-loop bandwidth of the PLL is much smaller than the loop sampling frequency. In [24], this stability limitation is analyzed in detail and it is demonstrated that to maintain loop stability:

$$w_n^2 < \frac{w_{REF}^2}{\pi(\pi + w_{REF}/w_z)} \tag{A.9}$$

Then, the choice of a small reference frequency also implies a limitation on the loop natural frequency, hence the closed-loop bandwidth.

Note that, as seen from A.5, the choice of the reference frequency and the feedback frequency division ratio are dependent on each other. The desired output frequency of the PLL is almost always the high priority and is an already-determined design parameter. Therefore, the implementation and power consumption of the frequency divider circuits might play a role in the selection of the division ratio, hence the reference frequency, especially for high frequency and low power designs.

Step 2: Loop Bandwidth

The stability limitation on the loop natural frequency, given by (A.9), can be expressed in terms of the loop bandwidth by using the loop parameter relations listed in Table V as shown below:

$$w_c < \frac{w_{REF}^2}{\pi(\pi w_z + w_{REF})} \tag{A.10}$$

Note that a commonly used rule-of-thumb [18] as the upper-limit for the loop bandwidth, set by Gardner's stability limit is:

$$w_c < \frac{w_{REF}}{10} \tag{A.11}$$

where w_{REF} is the reference radian frequency.

The loop bandwidth of the PLL is significant for not only the stability of the loop but also in terms of the loop settling time and PLL output noise. Wireless standards often define a PLL output accuracy and a settling time for the PLL to achieve frequency and phase lock in, when the output frequency is switched from one channel to another. Based on the second order approximation analysis of the loop, the settling time of the system is given by [11], [17]:

$$t_s = \frac{1}{\xi w_n} \ln\left(\frac{\Delta f}{a f_o \sqrt{1-\xi^2}}\right) \quad (\xi < 1) \tag{A.12}$$

$$t_s = \frac{1}{\xi w_n} \ln\left(\frac{\Delta f}{a f_o}\right) \quad (\xi = 1) \tag{A.13}$$

$$t_s = \frac{1}{w_n\left(\xi - \sqrt{\xi^2-1}\right)} \ln\left(\frac{\Delta f\left(\sqrt{\xi^2-1}+\xi\right)}{2a f_o \sqrt{\xi^2-1}}\right) \quad (\xi > 1) \tag{A.14}$$

where a is the output frequency accuracy at which the settling time should be measured, f_o is the output frequency and Δf is the frequency step that the output will cover. It is seen that the settling time behavior of the system depends on the damping of the system and different expressions are used for under ($\xi < 1$), critical ($\xi = 1$) and over ($\xi > 1$) damped systems.

Based on the loop parameter relations given in Table V, the expression ξw_n can be replaced with $w_c/2$. Then, it is seen that the settling time of the PLL is inversely proportional to its loop bandwidth. Therefore, for a frequency synthesizer designed to target a wireless standard, the settling time specification of the standard sets a lower limit to the loop bandwidth while Gardner's Stability limit sets an upper limit.

Another important performance metric that is affected by the loop bandwidth is the PLL output phase noise. The closed-loop PLL transfer function of (A.3) shows that the second order PLL acts as a low-pass filter, with a 3-dB bandwidth of w_c, to noise sources at its input. Then, a low-bandwidth means that the PLL will filter out more noise components from its input. However, a low-bandwidth also means that the PLL will have a large settling time as seen from the settling time expressions provided in (A.12) to (A.14).

Another important noise source in a PLL is the VCO. The discussions of DCO noise and the effect of the second order DPLL loop on it, in Chapter VI, can be applied to the VCO noise and the charge-pump based PLL. It was seen through the expressions in (6.20) and (6.21) that the PLL acts as a high-pass filter to the additive noise sources that are placed at the VCO output and as a band-pass filter to the noise sources that are added at the input of the VCO.

The loop bandwidth should be selected such that the phase noise contribution of the PLL input (crystal oscillator) and the VCO at the desired offset frequency, are similar. This is to prevent an unnecessary design overkill where the VCO might be optimized for excellent phase noise performance (often with the cost of power consumption) but the PLL input noise might dominate the output noise. In such a case for instance, a low bandwidth should be employed to further suppress the PLL input noise.

Most wireless standards place stringent phase noise requirements on the frequency synthesizer to satisfy interference suppression requirements. Therefore, LC tank based VCOs are commonly preferred in wireless frequency synthesizers over ring oscillators due to their superior phase noise performance. Very high accuracy crystal oscillators are also available at the low-end (a few MHz) reference frequencies of wireless synthesizers. Therefore, the loop bandwidth choice is often made based on the stability and settling time requirements. However, if the crystal oscillator accuracy is not as high as desired, a low bandwidth can be favored while a high bandwidth might be favored to reduce the power consumption of the VCO by assigning it a more relaxed phase noise budget.

Step 3: Damping Factor, Loop Zero and Natural Frequency

The second order approximation closed loop transfer function of the PLL shown in (A.3) was rewritten in the standard second order system transfer function form in (A.4). When the denominator of the transfer function is analyzed, it is seen that the placement of the closed loop poles depend on the damping factor. Table XXII summarizes the ranges of damping factor and the resulting pole locations [20] and their implications on the transient response of the PLL.

Table XXII. Damping factor and pole locations in a second order system

Damping Factor	Poles	Effect in Transient Response
$\xi > 1$	Both poles are real and negative	No ringing
$\xi = 1$	Both poles are real negative and equal $w_{p1} = w_{p2} = w_n$	No ringing
$0 < \xi < 1$	Complex conjugate poles with negative real parts	Ringing in the transient response
$\xi < 0$	Complex conjugate poles with positive real parts	Unstable system

Based on Table XXII, it is seen that to avoid ringing and to have a stable system, often, a critically damped system ($\xi = 1$) is favored. With the loop bandwidth (w_c) determined at *Step 2*, the natural frequency and the loop zero can both be determined from the expressions listed in Table V ($w_n = w_c/(2\xi)$, $w_z = w_n/(2\xi)$).

Step 4: Loop Filter Pole

As discussed in Chapter II, the charge-pump based PLL that employs the loop filter of Fig. 3 is a third-order type II system. But due to the placement of the filter pole w_p at frequencies that are further from the loop bandwidth, the system can be successfully approximated as a second order type II system. The loop relations given in Table VI and Table V are given for this second order approximation.

However, the placement of the loop filter pole is significant from a stability perspective and should be taken into account in characterizing the stability of the system. In the third order system of (A.1), the phase margin of the open loop gain is an important measure of the system stability and is given by [21]:

$$PM = \tan^{-1}\left(\frac{w_c}{w_z}\right) - \tan^{-1}\left(\frac{w_c}{w_p}\right) \tag{A.15}$$

To simplify the analysis of the third-order loop, the loop filter pole and zero can be placed at frequencies with a symmetric distance, α^2, to the loop bandwidth such that [17]:

$$w_z = \frac{w_c}{\alpha^2} \tag{A.16}$$

$$w_p = w_c\alpha^2 \tag{A.17}$$

Then, using the above zero placement and the loop expressions of Table V ($w_z = w_c/(4\xi^2)$), the damping factor can be expressed in terms of α:

$$\xi = \alpha/2 \tag{A.18}$$

To achieve a critically damped second order system behavior, the loop zero and pole placement factor α^2 can be chosen as 4. Then, for a desired loop bandwidth, the loop filter pole frequency can be determined. Note that for $\alpha^2 = 4$, the phase margin of the loop gain, based on (A.15), is 62 degrees, verifying the stability of the system.

Step 5: VCO Gain

After designing the loop control parameters as discussed in the previous steps, the building block design parameters such as the VCO gain, charge pump current, loop filter resistor and capacitor values, can be determined.

The VCO gain, K_{VCO}, is an important factor that not only determines the tuning range but also affects the loop gain, PLL output phase noise and spurious tones. The loop transfer function for noise sources at the VCO input is given for digital PLLs in Chapter VI in (6.21). This transfer function can be rewritten for charge-pump based PLLs as follows:

$$\frac{\phi_{out}}{V_{CTRL}} = \frac{K_{VCO}/s}{1 + K_{LOOP}\dfrac{(s+w_z)}{s^2}} = \frac{K_{VCO}s}{s^2 + K_{LOOP}s + K_{LOOP}w_z} \tag{A.19}$$

It is seen that large values of K_{VCO} imply that the effect of noise sources that appear at the VCO input (noise coupled from supply, ground or surrounding signals, loop filter output noise, etc.) will be significant at the PLL output. The amplitude of the spurious tone, at an offset frequency of f_m from the PLL output frequency f_o, is proportional to the VCO gain and the spurious tone offset frequency as follows:

$$A_{SP}(f_o + f_m) \propto \frac{K_{VCO}}{2\pi f_m} \tag{A.20}$$

A detailed analysis on the spurious tones of a PLL is given in [81].

A frequency synthesizer designed to meet the requirements of a wireless standard should cover all of the channel frequencies that the standard employs. Therefore, while the effect of the VCO gain on the output phase noise and spurious tones is significant, the design priority is to achieve a desired tuning range. Then the VCO gain is determined by:

$$K_{VCO} = 2\pi F_{tuning}/(V_{ctrl\_max} - V_{ctrl\_min}) \tag{A.21}$$

where F_{tuning} is the desired tuning range and the denominator is the dynamic range of the VCO control voltage. If the desired tuning range is very wide to deteriorate the output noise and spur performance, discrete tuning might be added to the VCO to enable/disable based on the target channel frequency. Another option to maintain small VCO gain is to increase the control voltage dynamic range. In the ZigBee synthesizer presented in Chapter II, special 3V supply transistors rather than the nominal 1.8V supply transistors of the implementation technology, are used in the PFD and charge pump to provide a larger control voltage dynamic range to accommodate a smaller VCO gain.

Step 6: Charge Pump Current and Loop Filter Components

Based on (A.2) and the loop bandwidth expression in Table IV ($w_c = K_{LOOP}$), the product of charge pump current and the loop filter resistor value is given by:

$$I_{CP} \times R = \frac{w_c 2\pi N}{K_{VCO}} \tag{A.22}$$

Since the loop bandwidth, divider ratio and the VCO gain are determined in the previous steps, the charge pump current and resistor value product is known. A small charge pump current increases the impact of current matching in the charge pump and therefore effects the PLL output spurs, hence a large current is desired. Another important concern is the size of the loop filter resistor and capacitor values. It should be taken into account that for a fully integrated solution, the loop filter component sizes should be realizable on-chip. Often, C_1 is too large for implementation. In that case, a capacitance multiplier [25] can be used to actively implement C_1 through a smaller passive capacitor.

Then, when the values of I_{CP} and R are determined, the value of C_1 is found from the zero frequency that was determined in *Step 3* while the value of C_2 is determined from the pole frequency that was found in *Step 4*.

Design Example

To demonstrate the design procedure discussed above, we can design a frequency synthesizer to meet the ZigBee requirements that were listed in Table II. The design parameters of the ZigBee frequency synthesizer that was implemented in Chapter II were listed in Table VI. Note that the implemented design aimed to generate quadrature output signals and therefore employed a VCO that operated at twice the channel frequency and was followed by a divide-by-2 circuit. In this example, let's assume that the synthesizer is expected to generate LO signals at the channel operating frequency and the quadrature signal generation within the synthesizer is not required. Therefore, VCO will be designed to operate at the channel frequency and the only dividers are in the feedback path.

Step 1: The reference frequency can be selected such that it is equal to the channel spacing of ZigBee:

$$FREF = 5MHz \tag{A.23}$$

Then, to obtain the desired output frequencies of 2.405GHz to 2.48GHz, with 5MHz of channel spacing, the feedback programmable divider ratios should be:

$$N = 481, 482, 483, ..., 495, 496 \tag{A.24}$$

Step 2: The upper and lower limits of the loop bandwidth are determined by the reference frequency and the settling time requirement, respectively. The output accuracy requirement for a ZigBee synthesizer is 40ppm. Then, for a critically damped loop, the lower limit of the loop bandwidth is determined by (A.13) and the upper limit is determined by (A.11) as follows:

$$2\pi \times 11kHz < w_c < 2\pi \times 500kHz \tag{A.25}$$

Considering the possible effects of parasitic poles on stability, the loop bandwidth can be selected conservatively as follows.

$$w_c = 2\pi \times 50kHz \tag{A.26}$$

Step 3: As noted in step 2, the loop can be designed as a critically damped loop:

$$\xi = 1 \tag{A.27}$$

Then, based on the value of the loop bandwidth and the loop parameter relations that are summarized in Table V, the natural frequency and loop zero frequency are determined as follows:

$$\begin{aligned} w_n &= 2\pi \times 25kHz \\ w_z &= 2\pi \times 12.5kHz \end{aligned} \tag{A.28}$$

Step 4: The loop zero and pole placement factor is chosen as $\alpha^2 = 4$, resulting in a phase margin of 62 degrees as discussed in the design procedure. Then, the loop pole is given by:

$$w_p = 2\pi \times 200kHz \tag{A.29}$$

Step 5: The desired tuning range to cover the 16 channel frequencies of the ZigBee standard is 75MHz. With additional margin for process variations, the output tuning range can be selected as 85MHz. If a 0.13μm technology is targeted with a nominal supply voltage of 1.2V, the control voltage dynamic range can be assumed as 1V. Then the VCO gain is:

$$K_{VCO} = 2\pi \times 85MHz/V \tag{A.30}$$

Step 6: Based on the above calculated design parameters we determine the relation between the charge pump current value and the loop filter resistor and capacitor values from (A.22):

$$I_{CP} \times R = 1.8V$$
$$I_{CP} \times R \times w_z = \frac{I_{CP}}{C_1} = 141kV/s \tag{A.31}$$

Then, taking the size of R and C_1 into account as well as possible matching problems in the charge pump, we choose the following parameters:

$$I_{CP} = 50\ \mu A$$
$$R = 36\ kohms$$
$$C_1 = 354\ fF$$
$$C_2 = 22\ fF \tag{A.32}$$

where C_2 is determined by the pole frequency of (A.29) once the value of R is determined. Since C_1 is a multiple of (16 times) $C2$, a capacitance multiplier can be used to multiply the capacitance of C_2 to implement C_1 on-chip.

Note that the above design procedure is mainly based on the second order approximation of the system. To check the behavior of the actual system (a third order system), we can place the design parameters to the third order system transfer function of (A.1) and analyze the frequency dependent and the transient behaviors.

To analyze the stability of the system, we can check the phase margin of the open loop gain. The open loop gain for the third order system is:

$$H_{OL} = K_{LOOP} w_p \frac{(s + w_z)}{s^2(s + wp)} \tag{A.33}$$

Note that the open loop gain for the second order approximation of the above system is:

$$H_{OL} = K_{LOOP} \frac{(s + w_z)}{s^2} \tag{A.34}$$

Fig. 109 and Fig. 110 show the Bode plots of the open loop gain for the third order system (A.33) and for the second order approximation (A.34), respectively.

Note that the phase margin calculation in Step 4 of the design procedure, given in (A.15), was done for the third order system, taking into account the loop filter pole w_p and the expected phase margin was 62 degrees. It is seen in Fig. 109 that the expected phase margin is accurate while the second order system's phase margin is more than that of the realistic third-order system since the loop filter pole is ignored. It is also seen that the gain bandwidth product (GBW) of the open loop gain, which was approximated to be equal to the closed loop 3dB bandwidth, is 50kHz as expected.

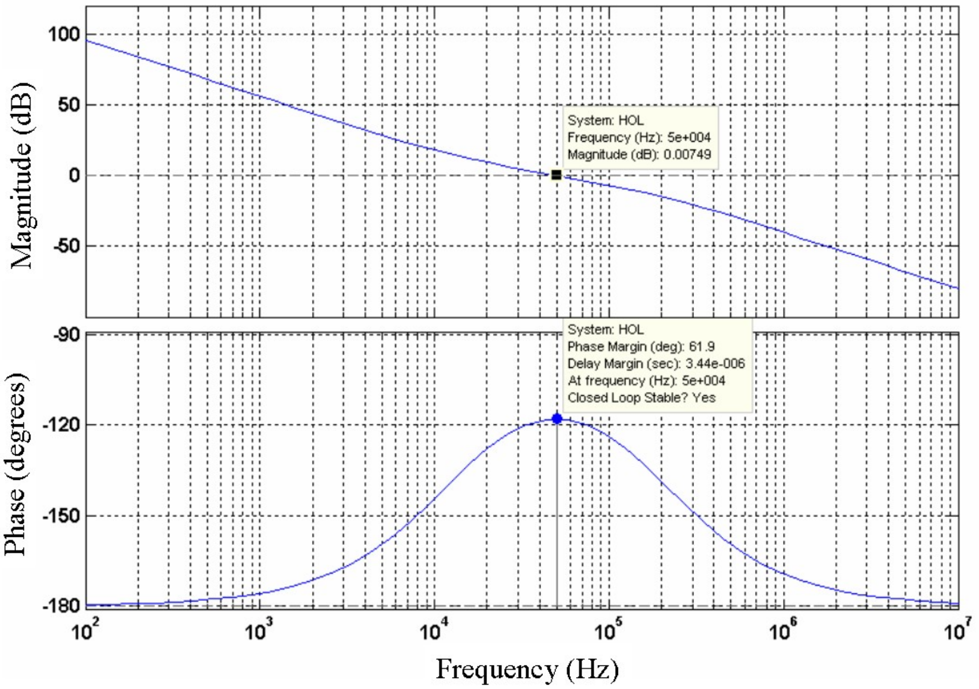


Fig. 109. Bode plot of the open loop gain of the third order PLL

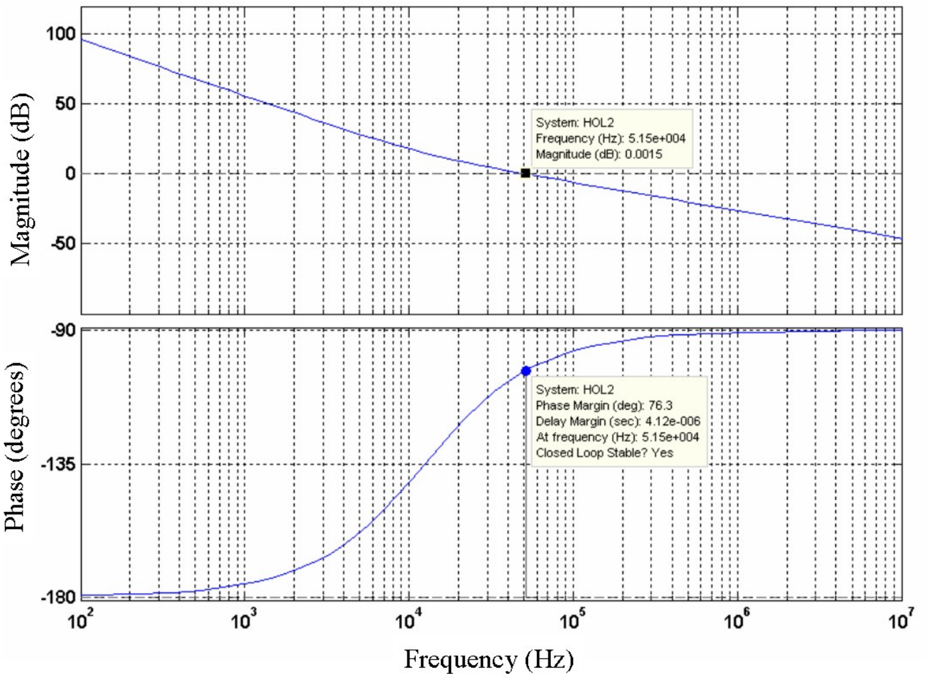


Fig. 110. Bode plot of the open loop gain of the second order approximation of the PLL

Fig. 111 and Fig. 112 show the frequency response of the closed loop transfer function for the realistic third order system (A.1) and for the second order approximation of the system (A.3), respectively. In the design steps, the expected approximate loop bandwidth was 50kHz. It is seen that when the second order approximation of the loop is plotted, the loop bandwidth is 61kHz. However, the real loop bandwidth, in the third order system, is 76kHz, higher than the targeted value. Therefore, as a last design step, it is important to plot the transfer functions that result from the design and iterate the design process to achieve the desired behavior.

In this appendix, a design procedure was described for the second order approximated phase locked loops. The approximation simplifies the design procedure significantly, while providing a good estimation of the actual behavior of the third order system. The design procedure also provides guidelines on the selection of several critical loop parameters such as the loop bandwidth, reference frequency and VCO gain as well as the loop filter parameter design values. Plots of the third order closed loop and open loop transfer functions demonstrate that a design example, that follows the design procedure, demonstrates a stable loop and a close approximation to the targeted design performance.

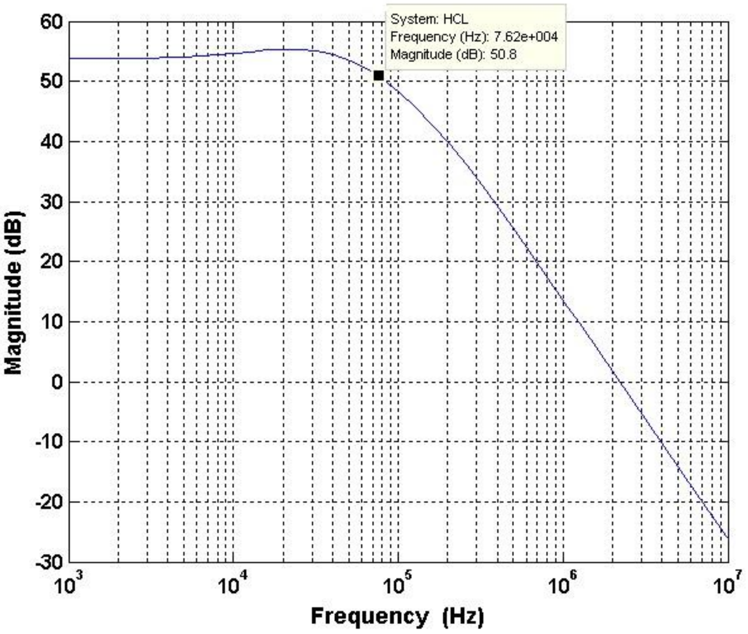


Fig. 111. Closed loop frequency response of the third order PLL

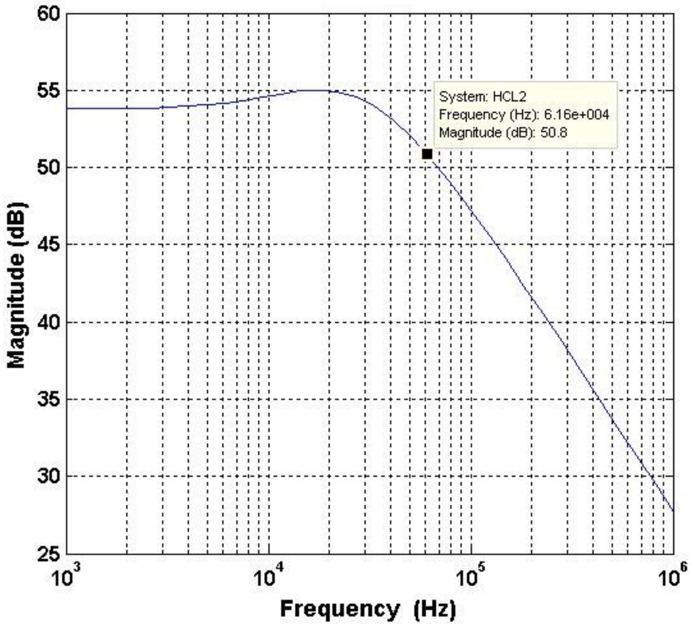


Fig. 112. Closed loop frequency response of the second order approximation of the PLL

APPENDIX B

A DESIGN PROCEDURE FOR ALL DIGITAL PLLS

A design procedure, similar to the one that was described in Appendix A, can be followed to design an all digital PLL. A conventional second order DPLL was discussed in Chapter VI, with important loop transfer functions and parameters summarized in Table XV and Table XVI.

Note that an ADPLL that employs the proportional-integral filter of Fig. 61, is a second order system, not a third order system as in the case of the charge pump based PLL of Chapter II. Then, the second order system design steps that are employed in Appendix A also apply to the DPLL and with better accuracy. Also note that the important loop parameter relations given in Table V of Chapter II, that were widely used in the design procedure discussed in Appendix A also apply to the DPLL.

The closed loop transfer function of the continuous time approximation of a second order type II DPLL was derived to be:

$$H_{CL\_DPLL}(s) = \frac{\phi_{out}}{\Delta\phi_{in}} = \frac{(K_{DLOOP} \times N)(s + w_z)}{s^2 + K_{DLOOP}s + K_{DLOOP}w_z} \tag{B.1}$$

where N is the feedback divider ratio the digital loop gain factor K_{DLOOP} is given by:

$$K_{DLOOP} = \left(\frac{Tref}{2\pi \times t_{res}}\right)\left(\frac{\alpha 2\pi f_{res}}{N2^F}\right) \tag{B.2}$$

Note that the ADPLL that was proposed in Chapter VII, can be analyzed with the same closed loop transfer function. For a wide range digital PLL, N takes on a wide range of values. To maintain the loop stability over the wide range of operation, a loop gain factor, A, was used in the proposed ADPLL of Fig. 62. Another system level change in the proposed system, when compared to the conventional ADPLL of Chapter VI, was the use of a coarse path at the output of the loop gain factor block, by separating the digital output word into its most significant and least significant bits. However, the most significant bits that are connected to the smart shifter, form a coarse path for frequency acquisition. As discussed in Chapter VII and demonstrated in the system level simulations of Section G, the coarse path settles before the loop is close to phase lock.

The linear phase analysis of PLLs assume that the PLL input phase difference is small. Then, it is safe to assume that the coarse path is settled for the linear phase domain analysis of the ADPLL that characterizes the loop stability and bandwidth. As a result, only the fine path of the ADPLL of Fig. 62 is considered to be active (the MSBs that are connected to the smart shifter are settled to their final value) and the loop behavior analysis of a conventional ADPLL that was shown in Fig. 60 will also apply to the ADPLL of Fig. 62.

The loop gain factor however, should include the variable loop gain block A, which can be set to 1 for a conventional narrow range ADPLL. The fractional bits were called F_2 in Fig. 62, to avoid confusion between the least significant bits F_1 that are connected to the loop filter. Then, the loop gain factor for the ADPLL of Fig. 62 is:

$$K_{DLOOP} = \left(\frac{Tref}{2\pi \times t_{res}}\right)\left(\frac{\alpha 2\pi f_{res}}{N2^{F_2}}\right) \tag{B.3}$$

The design steps for the ADPLL is similar to the steps that are provided in Appendix A. Therefore, in this appendix, the discussion will focus on the design procedure steps that are different for ADPLLs when compared to their charge-pump based counterpart. A design example for a wide range ADPLL will also be presented along with the design procedure. The design example will be the multi-protocol AD-

PLL that was implemented in Chapter VII, targeting a 5 GHz tuning range, designed and fabricated in 90nm CMOS technology.

Step 1: Reference Frequency and Division Ratio

While the choice of reference frequency is mainly done based on the channel frequencies and their spacing in a wireless PLL, for a wireline application, in absence of multiple channels, the choice of the reference frequency is determined by the concerns of the target loop bandwidth, implementation of the dividers, and for a multi-protocol or multi-rate application, the desired output frequencies.

Unlike a charge-pump based PLL, in an all digital PLL, the reference frequency also appears in the loop transfer function (B.3) and the loop zero expression as shown in (B.4) because the ADPLL is a sampled system with the sampling frequency often being equal to the reference frequency.

$$w_z \approx \frac{\beta}{\alpha} Fref \tag{B.4}$$

The choice of the reference frequency therefore is also significant from a loop behavior standpoint. It should be noted that the reference frequency appears in the loop gain factor K_{DLOOP} because of the TDC transfer function. The reference frequency together with the TDC resolution, determines the required number of TDC bits, hence complexity of the loop. If the TDC is employed to count the whole phase difference between the reference and the divider output as in [56], [72], then the TDC implementation complexity increases with a large reference period.

Note that in absence of equally spaced output channels, in wireline systems, a pulse-swallow divider is not needed, and the dividers can be implemented as cascaded division stages.

In this design, a reference frequency of 125MHz is selected, to be able to demonstrate various output frequencies (2GHz, 3GHz, 4GHz, 6GHz, 7GHz) through the use of simple cascaded divider stages that implement division ratios of:

$$N = 16, 24, 32, 48, 56 \tag{B.5}$$

Step 2: TDC resolution

The minimum resolution that can be obtained in the TDC is very much dependent on the implementation technology and the minimum gate delays as well as the implementation of the TDC. In 90nm technology, it was determined that the minimum sized inverter delays in schematic simulations are around 25ps-30ps. Note that not only this number increases in post-layout performance, but also pushing the absolute minimum gate delays means that with process variations, even if the supply voltage of TDC delay stages are modified to tune the resolution, the target resolution might not be achieved.

The GRO architecture that was used in this design and was proposed in [72], is capable of achieving a finer resolution than the minimum gate delay by overlapping multiple phases of a ring oscillator. By employing this structure, we can obtain a much finer resolution. In this design, we implemented a 20ps resolution (based on post-layout characterization of the TDC) to maintain a balance between the required number of bits and therefore complexity and the TDC quantization noise performance.

$$t_{res} = 20ps \tag{B.6}$$

Then, at 4GHz operation (N=32), the in-band noise of the ADPLL output, due to the TDC quantization will be $\pounds\_TDC_{PLL\_inband} = -97.7dBc/Hz$ based on (6.19).

Step 3: DCO resolution and Fractional Bits

The DCO frequency resolution, f_{res} should be determined by the desired tuning range and the phase noise expressions given in (6.22) and (6.23). For the design example, where a very wide, 5GHz, tuning range is required, the frequency resolution is determined to minimize the implementation complexity. Note that the DCO has three sections that are tuned, a binary weighted section that is controlled by the loop filter output MSBs, dithering portion that is controlled by a DSM, and a tunable row/column matrix, controlled by the smart shifter.

In the binary weighted portion of the DCO, the ratio of the MSB to the LSB of the binary weighted control units should be minimized to avoid settling time mismatches in the different bits of the binary weighted control and to maintain monotonicity in the DCO transfer function. Then, if a small number of bits will be assigned to the binary weighted portion of the DCO, we conclude that the DCO's target frequency tuning range should be covered in the smart shifter controlled row/column matrix.

In this design, a DCO resolution of 6.5MHz is chosen, which means that to cover the 5GHz tuning range, 768 units are needed in the DCO row/column matrix.

$$f_{res} = 6.5MHz \tag{B.7}$$

A delta-sigma modulator (DMS) is commonly employed in ADPLLs to dither the DCO control bits and improve the DCO resolution. Note that the DSM should be clocked at a frequency, higher than the reference frequency, and the effect of the dithering frequency on the DCO phase noise was shown in (6.23). In this design, since all of the divide ratios are multiples of 8, the last stage of the frequency dividers is a divide-by-8 circuit as shown in Fig. 62. The input of that divide-by-8 circuit is used as the dithering clock, resulting in:

$$Fdith = 8Fref \tag{B.8}$$

at frequency lock.

Since the DSM runs at a high frequency, it will consume higher power than regular low-frequency digital logic and therefore requires high performance adders.

To maintain a balance between the complexity of the DSM adders while achieving a fine resolution, the number of fractional bits to control the DSM is designed to be:

$$F_2 = 5 \tag{B.9}$$

Step 4: Loop Bandwidth

The upper limit for the loop bandwidth of the ADPLL comes from Gardner's stability limit (A.9) and (A.11). Note that in the ADPLL, in addition to the natural noise of the crystal oscillator that implements the reference frequency and the ring oscillator, TDC and DCO quantization noise also significantly contribute to the output phase noise, as discussed in Chapter VI Section C.

Note that the loop bandwidth is inversely proportional to the feedback divider ratio as shown in Table XVI. To have a constant loop bandwidth and maintain stability over the wide range of operation, a loop gain element, A, is introduced in the loop. To simplify the implementation of the multiplication, A is chosen to be powers of two such that the multiplication can be performed as shifting. The values of A in this design are:

$$A = 1, 2, 4 \tag{B.10}$$

Based on the state-of-the art ring oscillator performances listed in Table XIV, we can conclude that a well designed ring oscillator is expected to contribute around -93dBc/Hz phase noise at 1MHz offset. In this design, a wide bandwidth of 6.5MHz is implemented to suppress the ring oscillator phase noise throughout the wide bandwidth.

$$w_c = 2\pi \times 6.5MHz \tag{B.11}$$

Since 4GHz operation is a middle frequency in the wide tuning range, we design the loop bandwidth of 6.5MHz for the 4GHz operation.

If a $1/s^2$ phase noise behavior is assumed for the noise of the ring oscillator, then at the bandwidth of 6.5MHz we expect approximately -109dBc/Hz phase noise from the ring oscillator. The quantization noise at 6.5MHz offset frequency at the PLL output will be $\pounds\_DCO_Q = -121dBc/Hz$.

Step 5: Loop Filter Parameters

For a critically damped loop ($\xi = 1$), the loop zero frequency and natural frequency are determined as follows:

$$w_n = w_c/2 = 2\pi \times 3.25MHz \tag{B.12}$$

$$w_z = w_n/2 = 2\pi \times 1.625MHz \tag{B.13}$$

Once the zero frequency and reference frequency are determined, the ratio of the loop filter integration and proportional path constants are found from (6.9) as follows:

$$\frac{\alpha}{\beta} = \frac{Fref}{w_z} = 12.24 \tag{B.14}$$

The value of α is found from the loop gain factor K_{DLOOP} (6.13) (with N=32, 4GHz operation) as shown:

$$\alpha = 16; \tag{B.15}$$

Then, from (B.14), we find $\beta = 1.3$. However, if α and β are both powers of 2, then the implementation of the loop filter simplifies considerably since the multiplication can be implemented through shifters. Then, for β=1, the loop zero will become:

$$w_z = 2\pi \times 1.25MHz \tag{B.16}$$

and the damping factor is found by the loop bandwidth and the loop zero based on the expression provided in Table XVI:

$$\xi = 1.14 \tag{B.17}$$

The ADPLL system designed in this appendix is analyzed through the frequency response and step response of its closed loop transfer function. Fig. 113 demonstrates the step response of the designed ADPLL for the various operating frequencies, verifying the stability of the system. Fig 114 shows the closed loop frequency response for the various operating frequencies (2GHz-7GHz). The closed loop 3dB bandwidth is determined from these plots and are listed in Table XXIII.

Table XXIII. Closed loop bandwidth of the designed ADPLL based on its frequency response

ADPLL Frequency	Closed Loop Bandwidth
2 GHz (N=16)	14 MHz
3 GHz (N=24)	9.88 MHz
4 GHz (N=32)	7.72 MHz
6 GHz (N=48)	9.83 MHz
7 GHz (N=56)	8.54 MHz

In the design procedure, the closed loop bandwidth for 4GHz operation (N=32) was approximated through the open loop gain bandwidth product and designed as 6.5MHz. However, since this is an approximation, it is seen that the closed loop 3dB bandwidth of the feedback system is actually larger than the GBW of the open loop gain.

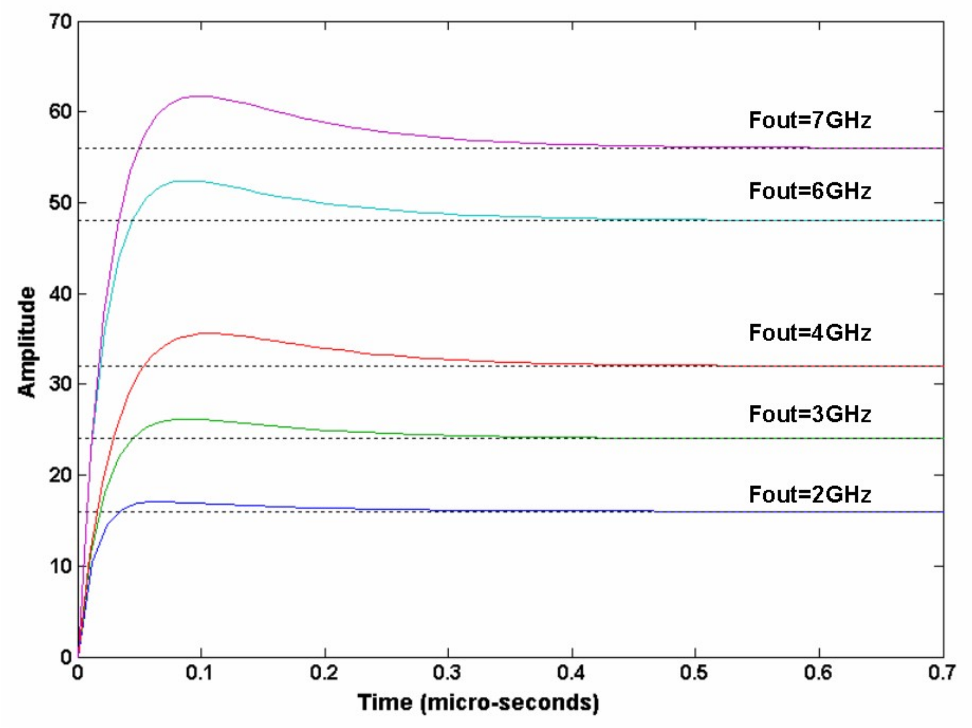


Fig. 113. Step response of the closed loop ADPLL

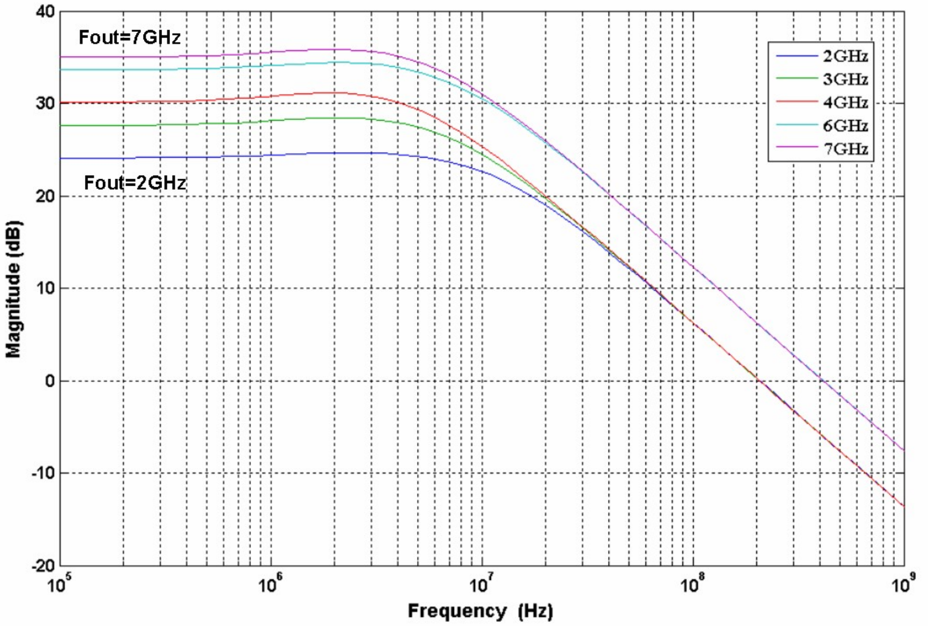


Fig. 114. Frequency response of the closed loop ADPLL

At 2GHz operation, the closed loop bandwidth is 14MHz, larger than one tenth of the reference frequency. However, the stability condition of (A.11) is actually an approximation of (A.9). The closed loop bandwidth values listed in Table XXIII all satisfy the stability condition of (A.9).

In this appendix, we provided a detailed design procedure for an ADPLL. The ADPLL, the design steps and design parameters of which is discussed in this Appendix, is implemented and fabricated in 90nm CMOS technology. Chapter VII discusses the implementation of this ADPLL in detail.

While the frequency response and the step response that are analyzed in this Appendix are valuable tools as a starting point in the design, an accurate time domain analysis of the system is needed taking nonidealities (additional delays, finite size of adders, etc.), the DSM dithering, and the coarse path of the system into account. Therefore, the transient behavior of the system is characterized through time domain simulations of an accurate time-domain model of the ADPLL and its building blocks in MATLAB Simulink environment in Chapter VII Section G, where the stability of the system is confirmed throughout the wide range of operating frequencies.

VITA

Didem Zeliha Türker was born in Antalya, Turkey. She received her B.S degree in Microelectronics Engineering from Sabanci University ,Istanbul, Turkey in 2003. In August 2003, she started working towards her Ph.D. at the Analog and Mixed Signal Design Center at Texas A&M University.

In summers of 2005 and 2006 she worked as a co-op researcher in the Analog and Mixed Signal Communications IC Design group of IBM T.J. Watson Research Center in York Town Heights, New York where she worked on equalization in high speed serial links and all digital PLLs for clocking applications.

During her doctoral program, Turker has worked on RFID systems, frequency synthesizers, high speed digital circuit techniques, frequency dividers, ring oscillators and all digital PLLs. Her research interests are analog and mixed signal circuit design and RF and high speed communication circuits.

She graduated with her Ph.D. from Texas A&M University in December, 2010 and can be reached through the Department of Electrical and Computer Engineering, 214 Zachry Engineering Center, Texas A&M University, College Station, TX 77843.