

A Fully Synthesized Injection Locked Ring Oscillator  
Based on a Pulse Injection Locking Technique

by

Mingze Li

A thesis submitted to the Faculty of Graduate and Postdoctoral  
Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Electrical and Computer

Ottawa-Carleton Institute for Electrical and Computer Engineering  
Department of Electronics  
Carleton University  
Ottawa, Ontario

© 2017

Mingze Li

## **Abstract**

This thesis proposes a novel, all synthesized, Injection Locked Ring Oscillator (ILRO). It employs a digitally tunable oscillator and a pulse injection locking technique. The frequency tuning range of the free running oscillator is from 210 MHz to 1.8 GHz with a 1.1 volt power supply. The tuning range from 1.0 to 1.8 GHz can be achieved with 215 tuning steps with a maximum step size of 11.2 MHz, that is well within the worst case 75 MHz (3rd sub-harmonic) and 32 MHz (9<sup>th</sup> sub-harmonic) locking range of the oscillator. The design occupies 127.5 um by 31.5 um of chip area and is implemented in TSMC's 65-nm CMOS technology. For 3rd harmonic injection locking, the ILRO's RMS jitter is 192.7 fs (1 KHz to 40 MHz) with a phase noise of -130.9 dBc/Hz at 1 MHz offset from the 1.62 GHz carrier while consuming 7.15 mW of power.

## Acknowledgements

Firstly, I would like to express my gratitude to my supervisor, Professor Ralph D. Mason for his encouragement and technical support in my hard time. Without his help, it is hard to me to finish all of the design. I appreciate the time of Professor Mason spending on my chip design and verification. Additionally, he also spends a lot of time to correct my immature academic writing. At the meantime, I learned a lot from himself about how to become an engineer with critical thinking and dignity personality.

The Authors would like to acknowledge the Canadian Microelectronics Corporation (CMC), Kingston, ON, Canada for providing fabricating access. Additionally, the authors would also like to thank Xing Zhou, Robert Vandusen, Gord Allan and Analog Devices Inc. (ADI) for providing testing support.

I appreciate my brother Xing Zhou who gives me a great help on chip design. Moreover, based on his tape-out experience, he helped me to address massive of stubborn problems. Additionally, I would like to express appreciation to my colleague Nahla who gives me a great help in chip testing. Moreover, I would like to thank my friends who give me great help and support during my stressful time.

I would like to appreciate the help and support from Mrs. Anna Lee and Mrs. Blazenka Power. I would also like to show my appreciation to Scott Bruce, Stephen Maclaurin and Nagui Mikhail for the technical support.

Last but not least, I would like to give my special thanks to my love of my entire life, my fiancée Shaoxin Fan, for all her support and understanding in three years. I would like to express my deep love to my parents Mrs. Yujuan Hu and Mr. KaiFeng Li for their

unlimited care and financial support. Based on the solid stone, I can pay all my attention to the intensive research. Some gratitude is beyond words to express. I love you.

# Table of Contents

<b>Abstract.....</b>	<b>ii</b>
<b>Acknowledgements .....</b>	<b>iii</b>
<b>Table of Contents .....</b>	<b>v</b>
<b>List of Tables .....</b>	<b>viii</b>
<b>List of Figures.....</b>	<b>ix</b>
<b>List of Abbreviations .....</b>	<b>xiii</b>
<b>1 Chapter: Introduction .....</b>	<b>1</b>
1.1 Chapter Overview.....	1
1.2 Motivation .....	1
1.3 Contribution.....	2
1.4 Thesis Organization.....	2
<b>2 Chapter: Background.....</b>	<b>4</b>
2.1 Introduction .....	4
2.2 Oscillator Application .....	4
2.3 Oscillator Specifications.....	5
2.3.1 Oscillator Phase Noise Analysis .....	5
2.3.2 Jitter.....	7
2.3.3 Methods for Minimizing Phase Noise and Jitter in Chips.....	9
2.4 LC Oscillator Basic Theory .....	10
2.5 Ring Oscillator Classifications .....	11
2.5.1 Single-Ended Ring Oscillator.....	13
2.5.2 Differential Structure .....	15
2.5.3 Pseudo Differential Structure.....	17
2.5.4 Phase Noise Analysis of Ring Oscillator .....	18

2.6	VCO/DCO Tuning Method based on the Ring Oscillator.....	21
2.6.1	Changing Load Capacitance Method.....	21
2.6.2	Delay Stages Method.....	22
2.6.3	Supply Voltage Controlled Method.....	22
2.7	Phase Locked Loops.....	23
2.7.1	PLL Transfer Function and Phase Noise Analysis.....	28
2.8	All Digital Phase Lock Loops.....	30
2.8.1	ADPLL Transfer Function.....	35
2.9	Basic Theory of Injection Locking.....	36
2.10	Chapter Conclusion.....	38
<b>3</b>	<b>Chapter: Design and Architecture.....</b>	<b>40</b>
3.1	Introduction.....	40
3.2	Pulse Injection Technique.....	40
3.3	Top Level Design.....	49
3.4	Digitally Controlled Ring Oscillator Design.....	50
3.5	Pulse Generator.....	54
3.6	Serial to Parallel Shift Register.....	55
3.7	Clock Tree Design.....	57
3.8	Oscillator Core Control Mechanism.....	58
3.9	Conclusion.....	60
<b>4</b>	<b>Chapter: ILRO Simulation Results.....</b>	<b>61</b>
4.1	Clock Tree Simulation Results.....	61
4.2	Pulse Generator Simulation Results.....	63
4.3	S/P Shift Register Simulation Result.....	65
4.4	ILRO Simulation Results.....	67
4.5	Conclusion.....	72

<b>5</b>	<b>Chapter: Implementation and Measurement Results .....</b>	<b>73</b>
5.1	Fabricated Microchip and Measurement Environment Setting .....	73
5.2	ILRO Measurement Results .....	77
5.3	Conclusion.....	85
<b>6</b>	<b>Chapter: Thesis Conclusion .....</b>	<b>87</b>
6.1	Accomplishments .....	87
6.2	Issues in the Design .....	87
6.3	Future Work.....	88
	<b>References .....</b>	<b>89</b>
	<b>Appendices.....</b>	<b>100</b>
	Top View Codes: .....	100
	Thermometer-Decoder Codes: .....	109
	Cock Tree Codes: .....	110
	Digitally Controlled Oscillator Codes: .....	113
	S/P Shift Register Codes: .....	125

## List of Tables

Table 1: Oscillator Specification List .....	5
Table 2: Categories of PLLs .....	31
Table 3: 3-bit Binary to Thermometer Decoder Truth Table .....	59
Table 4: Microchip Pins Arrangement.....	73
Table 5: Performance and Comparison.....	85

## List of Figures

Fig. 2.1 Output Spectrum of Ideal and Practical Oscillator [25] .....	6
Fig. 2.2 Phase Noise Spectrum [25].....	7
Fig. 2.3 Time Variarion Caused by Jitter.....	7
Fig. 2.4 Phase Noise to Jitter Demonstration.....	8
Fig. 2.5 Balanced LC based VCO.....	11
Fig. 2.6 Phase Shift in Barkhausen Criteria.....	12
Fig. 2.7 Linear Model of N Stages Ring Oscillator .....	12
Fig. 2.8 Single-ended Ring Oscillator.....	14
Fig. 2.9 Differential Ring Oscillator .....	15
Fig. 2.10 Pseudo Differential Pair.....	17
Fig. 2.11 The Practical Wave Form in Time and Frequency Domains .....	18
Fig. 2.12 VCO Using Varactors as A Tuning Method .....	21
Fig. 2.13 Ring Oscillator with Digitally Controlled Delay Stages .....	22
Fig. 2.14 Voltage Controlled Structure.....	23
Fig. 2.15 Output Signal when Voltage Control Changes.....	23
Fig. 2.16 PLL Basic Topology.....	24
Fig. 2.17 VCO Output Frequency vs. Control voltage .....	25
Fig. 2.18 Phase Detector Structure .....	26
Fig. 2.19 PD State Diagram .....	26
Fig. 2.20 Tristate Phase Detector and Charge Pump .....	27
Fig. 2.21 Basic Structure of a Loop Filter .....	27

Fig. 2.22 PLL Model in Frequency Domain.....	28
Fig. 2.23 Digital Design Flow .....	31
Fig. 2.24 TDC Based ADPLL Topology .....	32
Fig. 2.25 Vernier-delay-line Based TDC.....	33
Fig. 2.26 TDC-less ADPLL Topology .....	33
Fig. 2.27 Bang-bang Frequency and Phase Detector.....	34
Fig. 2.28 MASH-3 $\Sigma\Delta$ Modulator Topology .....	35
Fig. 2.29 S-domain Mode of a Type I ADPLL.....	35
Fig. 2.30 (a) Conceptual oscillator. (b) Frequency Shift by Injection Current.....	37
Fig. 2.31 Phase Difference between Input and Output .....	37
Fig. 3.1 CMOS Buffer .....	41
Fig. 3.2 Sinusoidal Wave Shaping into Ideal Square Wave .....	41
Fig. 3.3 Fourier Decomposition of Ideal Square Wave .....	42
Fig. 3.4 1GHz Square Wave after Discrete Fourier Transform in Cadence .....	43
Fig. 3.5 Non-ideal Square with Finite Rise/Fall Time.....	43
Fig. 3.6 The Relation Between Duty Cycle and Power Degradation .....	44
Fig. 3.7 3 <sup>rd</sup> Harmonic Power as a Function of Injection Signal Duty Cycle and Rise/Fall Time .....	46
Fig. 3.8 Free Running Oscillator Wave Form with Jitter Accumulation.....	46
Fig. 3.9 Pulse Injection Method in Time Domain .....	47
Fig. 3.10 Active High Tristate Inverter with Output Buffer .....	48
Fig. 3.11 Proposed Injection Locked Ring Oscillator of the High-Frequency Oscillator	49
Fig. 3.12 Three Stages Ring Oscillator.....	50

Fig. 3.13 Oscillator Array of High Frequency Output.....	51
Fig. 3.14 Operational Matrix .....	52
Fig. 3.15 Maximum Load Capacitance Array .....	53
Fig. 3.16 The Second stage of Frequency Tuning .....	54
Fig. 3.17 Pulse generator .....	54
Fig. 3.18 4-bit Serial In Parallel Out (S/P) Shifter Register .....	55
Fig. 3.19 4-bit Shift Register Timing Table.....	56
Fig. 3.20 Symmetric H-tree and X-tree Clock Distribution Method .....	57
Fig. 3.21 Clock Distribution Tree .....	58
Fig. 3.22 Frequency Control Words and Enabled Components .....	59
Fig. 4.1 Test Bench of Clock Tree .....	62
Fig. 4.2 The Simulation Result of Clock Tree Outputs .....	63
Fig. 4.3 The Zoomed In Pulse Width with Different Supply Voltages .....	64
Fig. 4.4 The Relationship Between Supply Voltages and Pulse Width.....	65
Fig. 4.5 Test Bench for S/P Shift Register.....	66
Fig. 4.6 S/P Shift Register Simulation Result.....	67
Fig. 4.7 Test Bench for Frequency Tuning, Injection Locking, and Locking Bandwidth	68
Fig. 4.8 Frequency Tuning with Different Supply Voltages .....	68
Fig. 4.9 Locking Bandwidth and Step Size of the 3 <sup>rd</sup> harmonic Injection Locking .....	69
Fig. 4.10 3 <sup>rd</sup> Sub-harmonic Injection locking Simulation in Time Domain .....	70
Fig. 4.11 9 <sup>th</sup> Sub-harmonic Injection locking Simulation in Time Domain .....	71
Fig. 4.12 Simulated Phase Noise at 1.62 GHz.....	72
Fig. 5.1 Microphotograph of the Fabricated ILRO.....	75

Fig. 5.2 4-Layer Customized PCB Layout.....	76
Fig. 5.3 ILRO Test Bench.....	77
Fig. 5.4 Frequency Tuning with the Different Supply Voltages.....	78
Fig. 5.5 Frequency Tuning Simulation and Measurement Results.....	79
Fig. 5.6 Frequency Tuning Curve and Tuning Step Size.....	79
Fig. 5.7 The Power Consumption of Free Running Oscillator with Different Supply Voltages .....	80
Fig. 5.8 Locking Bandwidth of Different Injection Harmonics and Tuning Step Size as a Function of Frequency .....	81
Fig. 5.9 Injection Locked ILRO Output Spectrum with 80MHz Bandwidth .....	82
Fig. 5.10 Phase Noise Simulation and Measurement Result at 1.62GHz.....	83
Fig. 5.11 Measured Phase Noise at 1.62GHz .....	83
Fig. 5.12 Comparison of Phase Noise with Digital Switching Noise On and OFF .....	84

## List of Abbreviations

ADC	Analog to Digital Converter
ADILRO	All Digital Injection Locked Ring Oscillator
ADPLL	All Digital Phase Locked Loop
BB	Base Band
C2C	Cycle to Cycle
CFP	Ceramic Flat Package
CP	Charge Pump
CPU	Central Processing Unit
DCO	Digitally Controlled Oscillator
DRAM	Dynamic Random Access Memory
FCW	Frequency Control Word
FPGA	Field-Programmable Gate Array
GPIB	General Purpose Interface Bus
IF	Intermediate Frequency
ILRO	Injection Locked Ring Oscillator
IOTs	Internet of Things
I/O	Input/Output
ISF	Impulse Sensitivity Function
LDO	Low Drop Output
LP	Loop Filter
LPF	Low Pass Filter

LSB	Least Significant Bit
MSB	Most Significant Bit
NMOS	N-channel Metal Oxide Semiconductor
PCB	Printed Circuit Board
PD	Phase Detector
PLL	Phase Locked Loop
PN	Phase Noise
PP	Peak to Peak
P&R	Place and Routed
PVT	Process Voltage Temperature
Q	Quality Factor
RF	Radio Frequency
RMS	Root Mean Square
RO	Ring Oscillator
SMA	Sub-Miniature version A
SOC	System on Chip
S/P	Serial to Parallel
SW	SWitching
TDC	Time to Digital Converter
UHF	Ultra High Frequency
VCO	Voltage Controlled Oscillator
VPWL	Voltage Piece Wise Linear

# **1 Chapter: Introduction**

## **1.1 Chapter Overview**

In the following sections, the motivation behind the All Synthesized Digital Controlled Injection Locked Ring Oscillator (ADIRLO) will be discussed, as well as the technical contributions and organization of this thesis.

## **1.2 Motivation**

With the rapid growth of the wireless communication industry, research related to communication circuits and architectures has received a great deal of attention [1] - [9]. Analog phase lock loops have been widely used in applications such as clock recovery, clock distribution, and frequency synthesis. However, they have a number of drawbacks, such as large chip area, high power consumption, and high supply voltage, which conflict with System-on-chip (SOC) design. To overcome these limitations, the charge-pump PLLs, commonly known as traditional analog Phase Locked Loops (PLLs), have been replaced with All-Digital Phase Locked Loops (ADPLLs) in many applications. ADPLLs have the following advantages: easier integration with other digital intensive circuits, better scalability with deep-submicron techniques, and easier programming [10] - [14]. Additionally, the market of Internet of Things (IOTs) is growing rapidly and to match the IOTs requirements the next generation micro-devices should have characters of high-speed, low power consumption and occupy small chip area. Therefore, using ADPLLs to replace the traditional analog PLLs are the optimized choice in the future transvers topologies.

A major component of ADPLLs is the Digitally Controlled Oscillator (DCO). Some important parameters of the DCO are frequency tuning range, step size or resolution, phase noise (frequency domain), jitter (time domain), spurs level, and power consumption. Most DCOs are designed by using an LC tank with tunable capacitor banks or a tunable ring oscillator. The LC oscillator employs the spiral inductor as its resonant component, which occupies large chip area. This thesis proposes an efficient way to address these issues. All of the functional blocks of the ILRO are synthesized from TSMC's built-in digital libraries, which provide high robustness against process variations.

### **1.3 Contribution**

The major research contribution of this thesis is the implementation of a fully synthesized injection-locked ring oscillator with small chip area, low power consumption, a large frequency tuning range, large locking bandwidth, and excellent phase noise performance. The design is fully synthesized and therefore can be migrated to any deep-submicron process by simply synthesizing the circuit in the target process using standard cell libraries. The locking bandwidth, power consumption, chip area and resolution will be improved. Also, the pulse injection locking technique can address the poor phase noise performance of ring oscillator.

### **1.4 Thesis Organization**

The thesis consists of six chapters. Chapter 1 gives a brief introduction to the thesis motivation and the contributions of the proposed work. Chapter 2 highlights general specifications and working theories of different kinds of oscillators. Basic knowledge about

the injection locking method and a general discussion of the Analog PLLs and All Digital PLLs are also presented in this Chapter. Chapter 3 reviews the pulse injection technique, and proposes the system level architecture of the ILRO and its individual functional blocks. Chapter 4 focuses on the simulation results, such as frequency tuning range, step size, and phase noise performance (including free running and injection locked). All of simulation results are based on the extracted view. Measurement environment settings, and results are provided in Chapter 5. Finally, Chapter 6 provides the conclusion of the current work and future research ideas based on this thesis

## **2 Chapter: Background**

### **2.1 Introduction**

This chapter presents several types of ring oscillators, mainstream All Digital Phase Locked Loops (ADPLLs) topologies, and the basic theory of the injection locking method. The applications of oscillators are illustrated in section 2.2 and section 2.3 lists the oscillator specifications and numerical analyses about phase noise and jitter. The basic topology of LC based oscillators and different kinds of ring oscillators are discussed in section 2.4 and section 2.5, respectively. Section 2.6 reviews frequency-tuning methods of Digitally Controlled Oscillators (DCOs) and Voltage Controlled Oscillators (VCOs). The architecture and important functional blocks of Phase Locked Loops (PLLs) and ADPLLs are illustrated in Sections 2.7 and 2.8, as well as, comparisons between these two topologies. Finally, section 2.9 provides a mathematical analysis of the injection locking method.

### **2.2 Oscillator Application**

Oscillators are utilized in a wide range of applications [15] [16] such as the reference clock for digital circuits or as the source of the Local Oscillator (LO) in a mixer. A mixer, as found in a wireless receiver, uses the oscillator's waveform as the reference frequency to mix down the input Radio Frequency (RF) signal to an Intermediate Frequency (IF) or BaseBand (BB) frequency [17]. In wireless applications, VCOs and DCOs are essential components in analog PLLs and ADPLLs.

## 2.3 Oscillator Specifications

The following specifications provide key parameters to evaluate oscillator performance: output frequency, frequency tuning range, tuning step size, free running phase noise, power consumption, and chip area. Table 1 shows oscillator specification parameters.

**Table 1: Oscillator Specification List**

Specifications	Unit
Output Frequency	GHz or MHz
Tuning Step Size	MHz or KHz
Phase noise	dBc/Hz
Supply Voltage	V
Chip area	$\mu\text{m}^2$

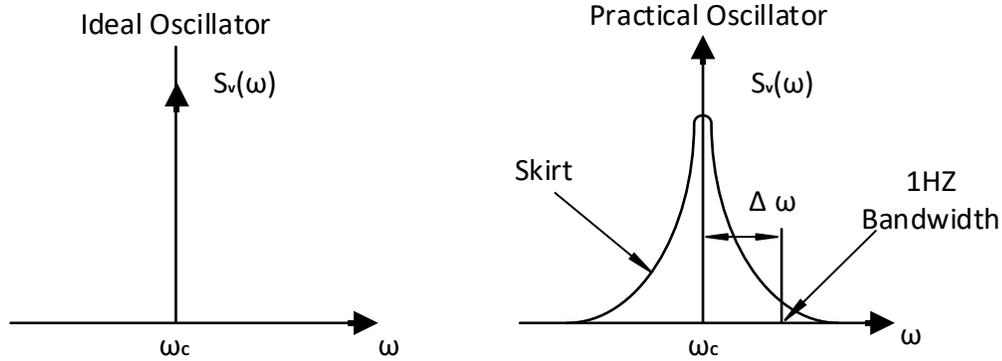
This thesis focuses on operating within the Ultra High Frequency UHF (from 300-3000MHz) band, which can provide low power consumption, smaller chip area, and high stability.

The free running phase noise characteristics of ring oscillator structure based DCOs have inferior performance compared to LC based oscillators [18] [19] [20]. Nevertheless, injection locking technique can be used to mitigate the poorer noise performance of ring oscillator based DCOs [21] - [24].

### 2.3.1 Oscillator Phase Noise Analysis

For an ideal oscillator operating at frequency  $\omega_c$ , all its power is concentrated in a single frequency  $\omega_c$  [25], as shown in Fig. 2.1. However, in practical oscillators, the spectrum spreads into nearby frequencies. This “skirt” is referred to as Phase Noise (PN)

and can cause interference in adjacent channels. The output signal of an ideal oscillator can be defined as  $v(t) = A \cos(\omega_c t + \Phi)$ , where  $A$  is the amplitude,  $\omega_c$  is the angular frequency, and  $\Phi$  is an arbitrary, but a fixed phase reference. Under this assumption, the power spectrum of the signal at a frequency of  $\omega_c$  is  $S_v(\omega) = \frac{A^2}{2} \delta(\omega - \omega_c)$  [25].

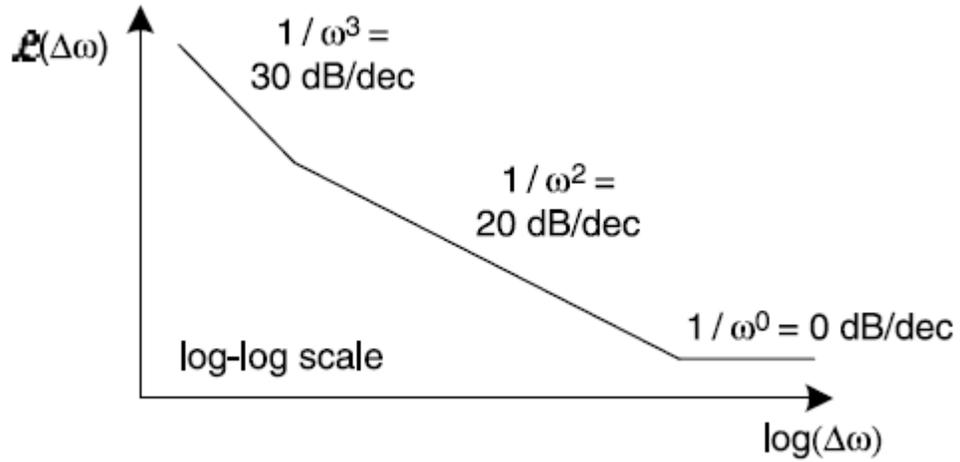


**Fig. 2.1 Output Spectrum of Ideal and Practical Oscillator [25]**

As shown in Figure 2.1, The PN can be characterized by considering a 1-Hz unit bandwidth at an offset of  $\Delta\omega$  from the carrier. The signal power within this 1 Hz band is divided by the carrier power, which gives the single sided spectral noise density of the oscillator in a unit of dBc/Hz. Equation 2.1 shows the single sided spectral noise density calculation:

$$\mathcal{L}\{\omega\} = 10 \log_{10} \frac{\text{(noise power in 1Hz bandwidth at frequency } \omega_c + \Delta\omega\text{)}}{\text{carrier power}} \quad (2.1)$$

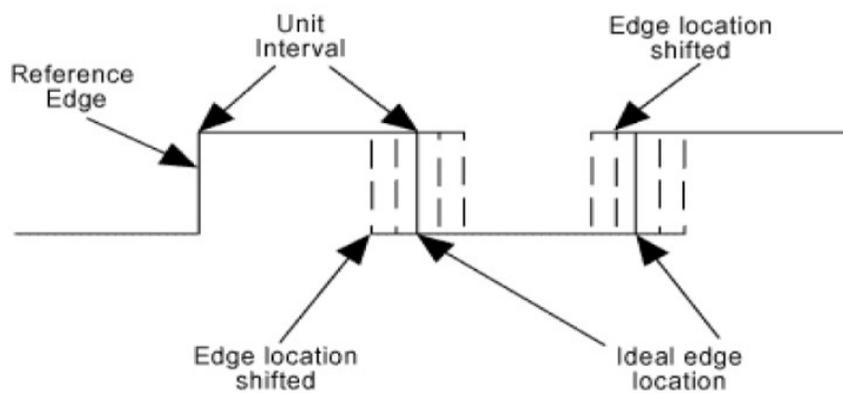
Figure 2.2 shows the phase noise spectrum of an oscillator on a log-log plot. The region  $1/\omega^2$  on Fig. 2.2 is referred to as the thermal noise region, which is caused by White Noise. The flicker (i.e.  $1/f$ ) noise of electronic devices is also substantial at lower offset frequencies. Flicker noise gets up-converted to the  $1/\omega^3$  region. The  $1/\omega^0$  region is the external thermal electronic noise added to the oscillator [25].



**Fig. 2.2 Phase Noise Spectrum [25]**

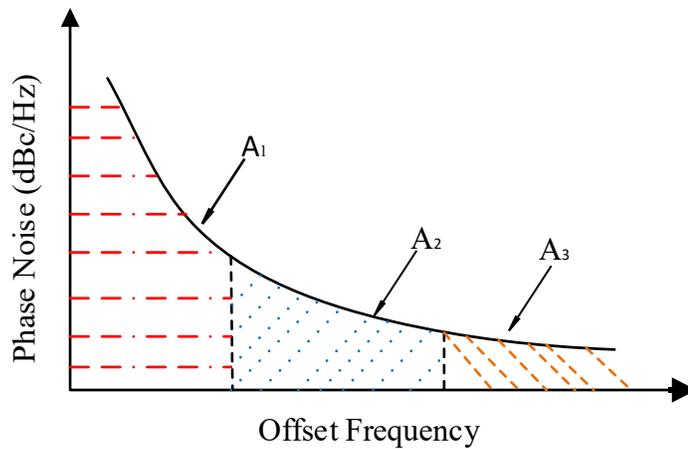
### 2.3.2 Jitter

Jitter is the timing variations of a set of signal edges from their ideal values. Thermal noise, power supply variations, device noise, and disturbance from adjacent circuits are the main contributing factors to jitter [26] [27] [28]. Fig. 2.3 shows a basic plot of jitter. There are several types of jitter, such as period jitter and cycle-to-cycle period jitter.



**Fig. 2.3 Time Variarion Caused by Jitter**

- Period jitter is the deviation in cycle time of a clock signal with respect to the ideal period over a number of randomly selected cycles. The standard deviation and the peak-to-peak value are referred to as the Root Mean Square (RMS) value and the peak-to-peak period jitter, respectively.
- Cycle to cycle (C2C) jitter is defined in JEDEC Standard 65B [27] as the variation in cycle time of a signal between adjacent cycles, over a random sample of adjacent cycle pairs. C2C jitter is typically reported as a peak value in pS, which defines the maximum deviation between the rising edges of any two consecutive clocks.



**Fig. 2.4 Phase Noise to Jitter Demonstration**

As shown in Fig. 2.4, the phase noise curve is broken into a number of individual areas ( $A_1$ ,  $A_2$ , and  $A_3$ ) and the offset frequency range is typically from 1KHZ to 40MHZ [20]-[24]. Phase noise can be transferred to RMS jitter using Equations 2.2, 2.3, and 2.4 [27]:

$$A = 10 \log_{10}(A_1 + A_2 + A_3) \quad (2.2)$$

$$\text{RMS phase jitter}_{(radians)} = \sqrt{2 * 10^{A/10}} \quad (2.3)$$

$$\text{RMS phase jitter}_{(seconds)} = \frac{\sqrt{2 * 10^{\frac{A}{10}}}}{2\pi f_0} \quad (2.4)$$

Where  $A_1$ ,  $A_2$ , and  $A_3$  represent individual power ratios. ‘A’ represents the integration of phase noise power in a unit of dBc, and  $f_0$  is oscillator frequency.

### 2.3.3 Methods for Minimizing Phase Noise and Jitter in Chips

At the chip level design, jitter and phase noise can be reduced using various methods [27]:

- Routing considerations

For time-sensitive signal paths, routes must be kept as short as possible, and crossing of digital tracks should be avoided.

- Buffer size considerations

If a buffer is used in clock distribution between clock-sensitive blocks, choosing the proper size of the buffer is necessary. The buffer should have enough drive strength to provide fast rising/falling edge time to avoid high phase noise.

- Ground and power supply considerations

Substrate and ground noise are also key contributors to jitter. In a clock-sensitive circuit, the ground bounce vibrates from tens of millivolts to several volts. To reduce the ground bounce effect, as many of the voltage supply pins and ground pins as possible should be placed on chip.

- Blocking digital systems and analog devices

It is good practice to separate power supplies for digital circuits and sensitive analog components. Power supplies for digital circuitry, particularly high-drive outputs, are highly susceptible to noise pickup, and can contribute greatly to jitter if used for timing circuitry. It might be advantageous to use a supply filter for circuits such as PLLs to further reduce the effects of supply noise. Additionally, substrate moats are also useful for isolating digital from analog circuits.

## 2.4 LC Oscillator Basic Theory

Wireless communications require outstanding phase noise performance and relative low power consumption, especially in mobile devices, hence, the LC based Voltage Controlled Oscillator should be taken into consideration [29] - [32]. The basic topology of balanced NMOS LC based VCO is shown in Fig. 2.5. The close loop gain should be greater or equal to unity magnitude with no imaginary component. In reality, there exists losses among varactors, transistors, and inductors. Additionally, practical on-chip inductors are formed into spiral inductors with low quality factor that dominates the losses of the VCO tank. However, these inductors occupy large chip area [31] - [35].

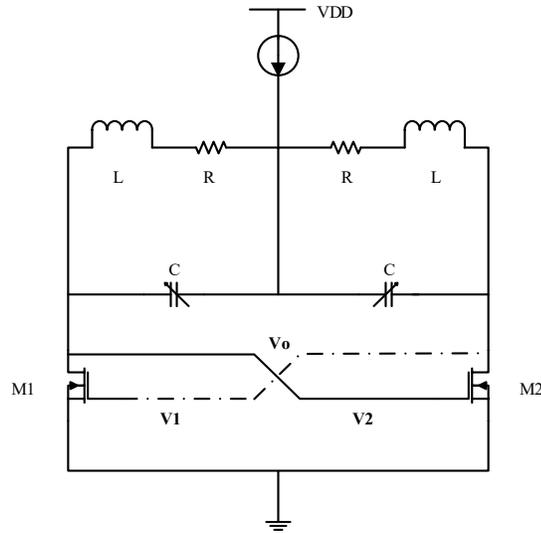
The quality factor  $Q_L$  of the inductor is given by:

$$Q_L = \frac{\omega_o L}{R} \quad (2.5)$$

Where  $\omega_o$  is the operating frequency of a VCO,  $L$  is the value of the inductance, and  $R$  is the inductor's equivalent series resistance. The oscillating frequency of the balanced VCO

with ideal varactors and transistors is expressed as  $\omega_o = \frac{1}{\sqrt{2LC}} \sqrt{1 - \frac{R^2 C}{L}}$  and the  $g_m$  of each

transistor must be greater than  $\frac{RC}{L}$  for oscillating to occur. The design details for LC based VCOs and phase noise performance analysis are discussed in [29] - [38].

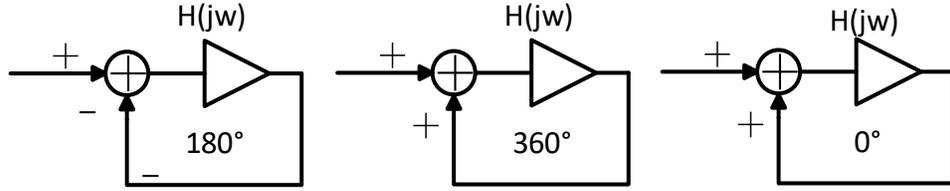


**Fig. 2.5 Balanced LC based VCO**

## 2.5 Ring Oscillator Classifications

Ring Oscillators (ROs) have many advantages, such as small die area, low power consumption, and wide frequency tuning range. ROs are also easily integrated in Very Large Scale Integration (VLSI), and have been widely used in Central Processing Units (CPU), Dynamic Random Access Memory (DRAM), clock generation, system synchronization, oversampling Analog to Digital (A/D) converters, and wired transceivers ( i.e. Gigabit Ethernet). Recently, ROs have also been used in wireless communication systems [39] - [44]. The following sections illustrate different types of ROs, parameter specifications, and noise analysis of ROs.

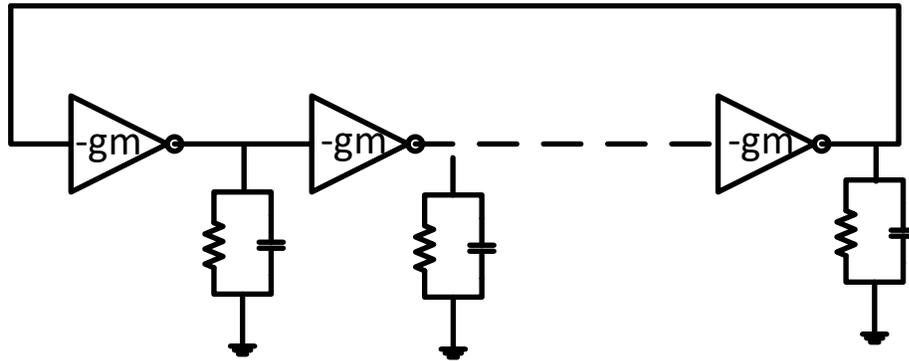
To satisfy the oscillation condition, also called Barkhausen Criteria, a ring oscillator must provide a phase shift of  $2\pi$  and have unity gain at the oscillation frequency, as shown in Fig. 2.6 [45]. The phase shift of each delay cell should be  $\pi/N$ , where N is the number of delay stages.



**Fig. 2.6 Phase Shift in Barkhausen Criteria**

The linear model, as shown in Fig. 2.7, gives a comprehensive way to determine the output frequency of a RO. Each delay stage is identical and has the same parameters. Therefore, all stages have the same gain, expressed in Equation 2.6:

$$A_1(j\omega) = A_2(j\omega) = A_3(j\omega) = A_N(j\omega) = \frac{-g_m R}{1 + j\omega RC} \quad (2.6)$$



**Fig. 2.7 Linear Model of N Stages Ring Oscillator**

Based on the Barkhausen Criteria, the total gain and phase shifting of the RO is given by:

$$|A_1(j\omega) \cdot A_2(j\omega) \cdot A_3(j\omega) \cdot \dots \cdot A_N(j\omega)| = 1 \quad (2.7)$$

$$\angle A(j\omega) = \theta = \arctan \omega RC = \frac{2k\pi}{N} \quad (2.8)$$

The frequency of oscillation is expressed as:

$$\omega_0 = \frac{\tan \theta}{RC} \quad (2.9)$$

By design, each delay stage provides the same delay,  $t_d$ . The signal goes through each of the N delay stages once to provide the first phase shift in a time of  $N \cdot t_d$ . The signal then must go through each stage a second time to obtain the remaining phase shift, resulting in a total period of  $2Nt_d$ . Hence, the oscillation frequency is:

$$f_0 = \frac{1}{2Nt_d} \quad (2.10)$$

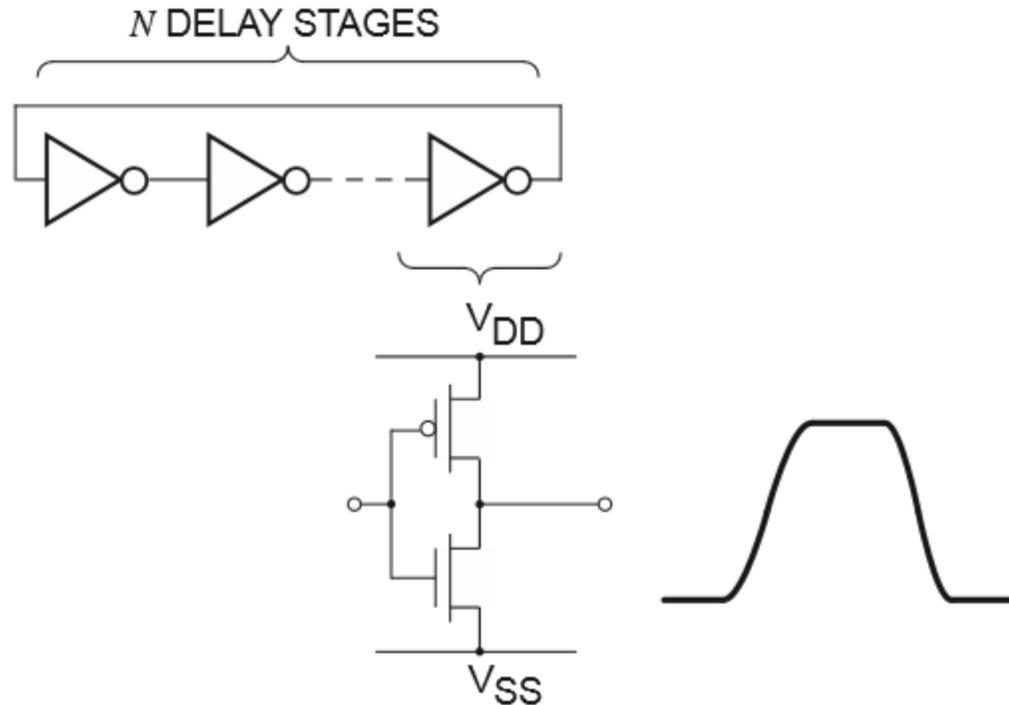
It is a challenge to get an accurate delay for each stage due to nonlinearities, component parasitics, and other unpredictable issues. Based on the theory presented in [46], the aforementioned equations can be modified as:

$$f_0 = \frac{I_{ss}}{2NV_{sw}C} \quad (2.11)$$

Where  $I_{ss}$  is the tail current used in a delay stage,  $V_{sw}$  is the Peak-to-Peak (PP) amplitude of the voltage waveform, and  $C$  is the load capacitance of the delay stage. As a result, the delay per stage is defined as the total change in the differential output voltage at the midpoint of the transition,  $I_{ss}$ , divided by the load capacitance to give the differential slew rate,  $I_{ss}/C$ , resulting in a delay per stage of  $CV_{sw}/I_{ss}$ .

### 2.5.1 Single-Ended Ring Oscillator

Fig. 2.8 [47] shows an example of a ring oscillator with single-ended output. The odd number of stages is required to satisfy the oscillating condition (Barkhausen Condition) [45] (the minimum number of delay stages is three).



**Fig. 2.8 Single-ended Ring Oscillator**

Advantages of Single-ended RO include:

- Power efficiency

The delay stage only draws power when there exists a signal transition, in contrast to true differential stages, which require a bias current that is always flowing whether or not the signal is transitioning.

- Signal amplitude

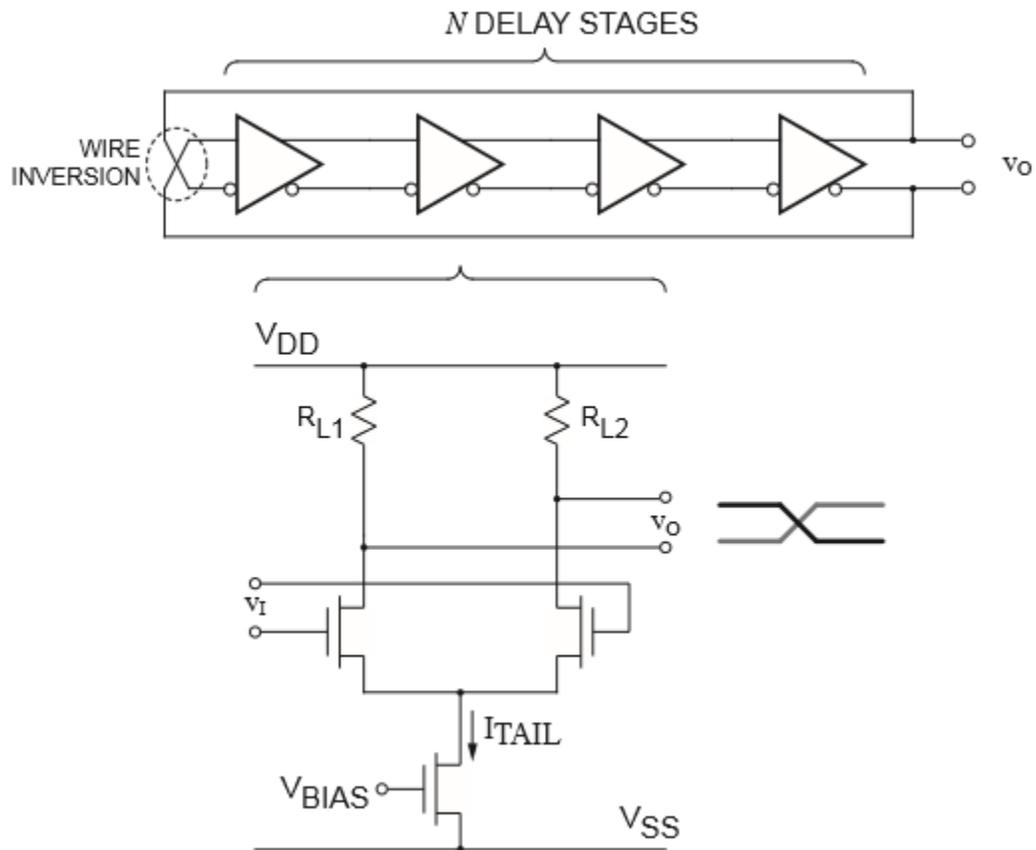
This delay stage is capable of a full rail-to-rail signal swing. Large signal amplitude is associated with lower jitter [47].

Disadvantages of this approach includes:

- The odd number of stages, which is undesirable in some cases, such as when in-phase and quadrature outputs are desired, which requires an even number of stages.

## 2.5.2 Differential Structure

Fig. 2.9 shows the basic structure of a differential ring oscillator. In the circuit structure, two resistors  $R_{L1}$  and  $R_{L2}$  and a biasing transistor (in the bottom) are used for keeping the differential pairs working in the active region.



**Fig. 2.9 Differential Ring Oscillator**

Advantages of the differential structure include:

- Number of stages

Since both phases of the signal are available, both odd and even number of delay stages can satisfy the oscillation condition.

- Interference rejection

The differential structure offers good common mode rejection.

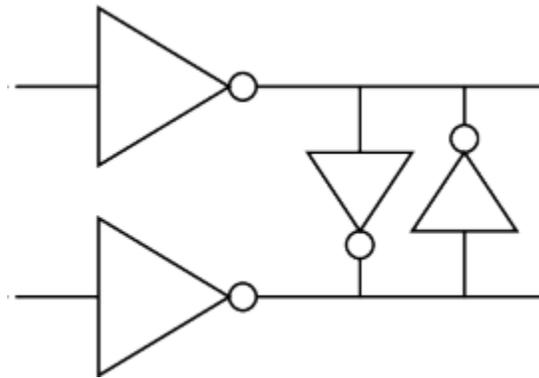
Disadvantages of the differential structure include:

- Low signal swing

The output signal amplitude must be much lower than the supply voltage to keep all devices in the active region of operation, and thus preserve the common mode rejection properties of the differential pair [47]. Also small signal swing will lead the circuit to be more sensitive to jitter.

### 2.5.3 Pseudo Differential Structure

Pseudo differential elements are designed using a differential pair with a latch. The latch can be implemented using cross-coupled inverters, as shown in Fig 2.10. This structure provides rail-to-rail swing and a 180° phase shift.



**Fig. 2.10 Pseudo Differential Pair**

Advantages of the pseudo differential pair include [47]:

- Full signal swing

This structure provides full swing from VDD to VSS.

- Supply/substrate interference rejection

With symmetric layout and good matching between the two sides of the pseudo differential circuit, there can be a reduction in amplitude coupling interference.

Disadvantages of the pseudo differential pair include:

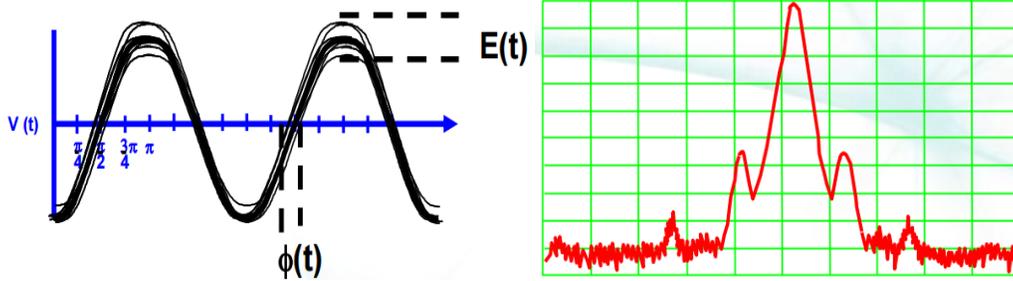
- There is no rejection of delay modulation. Each side of the pseudo differential circuit is affected by delay modulation due to supply voltage variation.

## 2.5.4 Phase Noise Analysis of Ring Oscillator

For the ideal oscillator, without any external or internal noise interference, the output signal can be expressed as  $V_{out}(t) = A \sin(\omega t + \phi)$ , where  $A$  and  $\phi$  represent fixed amplitude and phase, respectively. However, in practical oscillators, this equation should be modified by noise:

$$V_{out}(t) = [A + E'(t)] \cdot f[\omega t + \phi + \phi'(t)] \quad (2.12)$$

Where  $E'(t)$  and  $\phi'(t)$  represent random amplitude and phase changes due to noise, as shown in Fig. 2.11.



**Fig. 2.11 The Practical Wave Form in Time and Frequency Domains**

A general equation for phase noise can be derived using a single-ended ring oscillator as an example. If an impulse current, which is generated by a current source, with area of  $\Delta q$  injects into one of the ring oscillator's nodes at time  $\tau$ , it will cause an immediate change in voltage, given by the following equation:

$$\Delta V = \frac{\Delta q}{C_{node}} \quad (2.13)$$

Under this assumption, the extra current injection will produce a phase shift. The change in phase is proportional to the injected charge, and is given by the following equation [48]:

$$\Delta \phi = \Gamma(\omega_0 t) \cdot \Delta V / V_{swing} = \Gamma(\omega_0 t) \cdot \Delta q / q_{swing} \quad (2.14)$$

In Equation 2.14,  $\Gamma(\omega_0 t)$  is a dimensionless function with a period of  $2\pi$ . Moreover,  $\Gamma(x)$  represents the sensitivity of every point on the waveform to a perturbation.  $\Gamma(x)$  is also referred to as the Impulse Sensitivity Function (ISF).

According to previous work [48], if the impulse current is injected into the circuit at transition intervals, it will have a maximum impact on the phase shift. If the impulse is applied at the peak of the voltage across the capacitor, there will be no phase shift and only amplitude change. Based on the previous work [48], we can get the following impulse response:

$$h_\phi(t, \tau) = \frac{\Gamma(\omega_0 \tau)}{q_{max}} u(t - \tau) \quad (2.15)$$

Based on Equation 2.14, if the noise current is  $i(t) = I_n \cos[(n\omega_0 + \Delta\omega)t]$ , the general  $\phi(t)$  is given by:

$$\phi(t) \approx \frac{I_n c_n \sin(\Delta\omega t)}{2q_{max} \Delta\omega} \quad (2.16)$$

The Single Sideband (SSB) noise spectral density is expressed as:

$$\mathcal{E}\{\Delta\omega\} = 10 \log \left( \frac{\overline{i^2} \sum_{n=1}^{\infty} C_n^2}{8q_{max}^2 \Delta\omega^2} \right) \quad (2.17)$$

Where  $\overline{i^2} / \Delta f$  is the input noise current with a white power spectral density and it is based

on Parseval's equation,  $\sum_{n=0}^{\infty} c_n^2 = \frac{1}{\pi} \int_0^{2\pi} |\Gamma(x)|^2 dx = 2\Gamma_{rms}^2$

As a result, phase noise is given by:

$$\mathcal{E}\{\Delta\omega\} = 10 \log \left( \frac{\overline{i^2} \Gamma_{rms}^2}{4 \cdot \Delta\omega^2 \cdot q_{max}} \right) \quad (2.18)$$

The aforementioned Equations 2.12 through 2.18 illustrate the derivation process of general phase noise. To calculate the phase noise of a single-ended ring oscillator, we need to know the RMS value of the ISF. Based on simulation results from previous work, [48] and assuming the signal's rising and falling time are equal, the RMS value of the ISF,  $\Gamma_{rms}^2$ , can be expressed as:

$$\Gamma_{rms}^2 = \frac{1}{2\pi} \int_0^{2\pi} \Gamma^2(x) dx = \frac{2}{3\pi} \left( \frac{1}{f'_{max}} \right)^3 \quad (2.19)$$

Where  $f'_{max}$  is the maximum slope of the normalized waveform.

In order to turn the previous equation into the normalized period, it must be multiplied by  $2N$ . Hence, we can approximate the RMS value of the ISF as follows:

$$\Gamma_{rms} = \sqrt{\frac{2\pi^2}{3\eta^3} \frac{1}{N^{1.5}}} \quad (2.20)$$

Based on Equation 2.20, it is seen that the ISF is dimensionless, and independent from noise, frequency, and amplitude.

According to previous research, [48], the noise spectral density of the drain current in a CMOS device is expressed as:

$$\frac{\overline{i^2}}{\Delta f} = 4kT\gamma\mu C_{ox} \frac{W}{L} \Delta V \quad (2.21)$$

Where  $\mu$  is mobility,  $C_{ox}$  is gate-oxide capacitance,  $\gamma$  is the attenuation coefficient of the electron wave function in the oxide, and  $W$  and  $L$  are the channel width and length of the transistor [46] [48], respectively.

Under the assumption of equal-length of PMOS and NMOS, we can replace  $\mu$  and  $W$  with  $\mu_{eff}$  and  $W_{eff}$ , where  $W_{eff} = W_n + W_p$  and  $\mu_{eff} = \frac{\mu_n W_n + \mu_p W_p}{W_n + W_p}$ .

Therefore, we can deduce a new equation for the oscillation frequency:

$$f_o = \frac{\mu_{\text{eff}} W_{\text{eff}} C_{\text{ox}} \Delta V^2}{8 \eta N L q_{\text{max}}} \quad (2.22)$$

We combine Equations 2.16, 2.17, 2.20, 2.21, and 2.22 together to encompass all necessary expressions. The single-ended ring oscillator phase noise equation is given by:

$$\mathcal{E}\{\Delta\omega\} = \frac{8}{3\eta} \frac{kT}{P} \frac{V_{DD}}{V_{\text{char}}} \frac{f_o^2}{\Delta f} \quad (2.23)$$

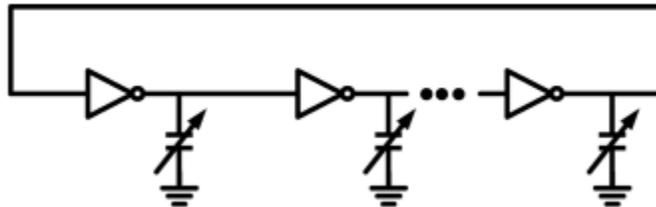
Where  $V_{\text{char}} = \frac{\Delta V}{\gamma}$  and P is the total power dissipation in the ring oscillator.

## 2.6 VCO/DCO Tuning Method based on the Ring Oscillator

Changing supply voltage, load capacitance, the number of delay stages, and the biasing current are the common ways to change the output frequency of ring oscillators. This section roughly characterizes these three tuning methods.

### 2.6.1 Changing Load Capacitance Method

Fig. 2.12 shows VCO tuning using varactors [47] [49]. By changing the value of the load capacitance, the charging and discharging times will be altered, and the output frequency will be changed as well. This approach is not widely used in practical applications because the tuning range is narrow and varactors are sensitive to PVT variations [49].



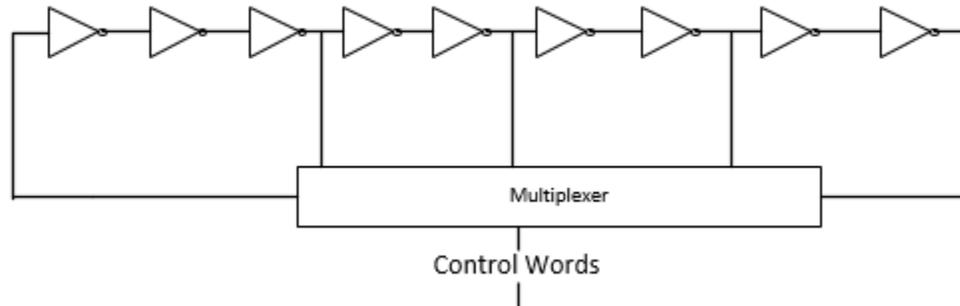
**Fig. 2.12 VCO Using Varactors as A Tuning Method**

### 2.6.2 Delay Stages Method

Fig. 2.13 shows a common method to change the output frequency using a Frequency Control Word (FCW) to vary the number of delay stages [47]. The primary oscillating components are the three inverters connected serially with a feedback loop, which gives the maximum output frequency. If a lower frequency output is required, more delay elements are added into the basic delay line through the MUX. This method provides a wide tuning range, and is a purely digitally controlled method, which is portable compared to the previous approach. The output frequency is given by:

$$f_{out} = \frac{1}{2(3T_d + T_{mux})} \quad (2.24)$$

Where  $T_d$  is the propagation delay of a single inverter and  $T_{MUX}$  is the delay time of the multiplexer.

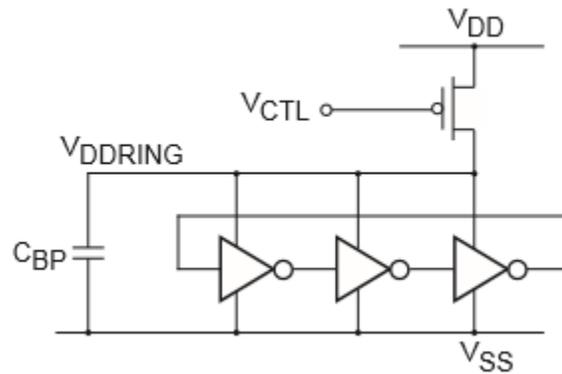


**Fig. 2.13 Ring Oscillator with Digitally Controlled Delay Stages**

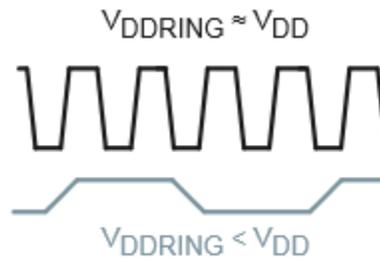
### 2.6.3 Supply Voltage Controlled Method

Frequency tuning can also be achieved by varying the voltage of the oscillator waveform itself, either by changing the supply voltage or the delay stage threshold [47]. However, output signal swing and jitter performance will be affected by the variation of

the control voltage. This is shown in Fig. 2.14 and Fig. 2.15. The system supply voltage  $V_{DD}$  will be reduced to a lower value  $V_{DDRING}$  due to the voltage drop source and drain terminal of the biasing P-channel MOSFET (PMOS). Additionally,  $V_{CTL}$  is the voltage control part for the VCO and a decoupling capacitor  $C_{BP}$  is used to decrease ripple and reduces coupling of supply and substrate variation to the  $V_{DDRING}$  voltage.



**Fig. 2.14 Voltage Controlled Structure**



**Fig. 2.15 Output Signal when Voltage Control Changes**

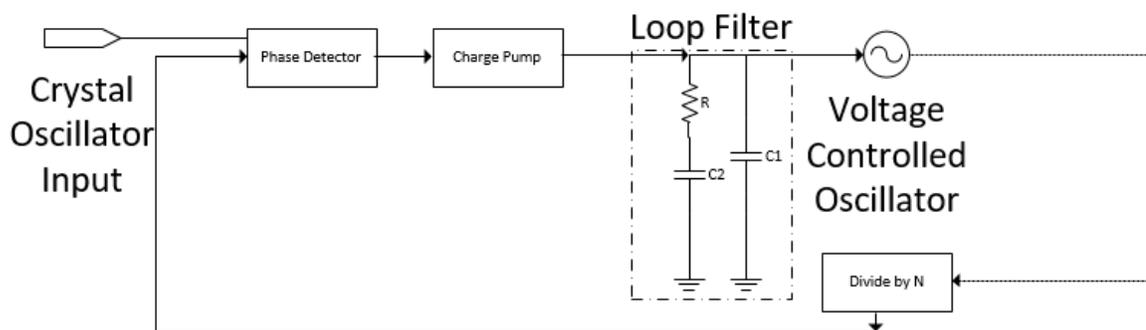
## 2.7 Phase Locked Loops

Phase Locked Loops (PLLs) are widely used in wireless communication systems [3, 10, 11, 13, 14, 21, 49]. PLLs can be used to demodulate a signal, recover a signal from a noisy communication channel, generate a stable frequency at multiples of an input frequency, or distribute precisely timed clock pulses in digital logic circuits such

as microprocessors. Since a single integrated circuit can provide a complete phase-locked-loop building block, the technique is widely used in modern electronic devices, with output frequencies from a fraction of a hertz up to many Gigahertz. As an important control system, a PLL generates an output signal whose phase is related to the phase of an input signal.

An analog PLL is broken into several blocks of operation: VCO, Divider, Phase Detector (PD), Charge Pump (CP), Loop Filter (LP), and a frequency reference, typically a crystal oscillator, as seen in Fig. 2.16. The VCO generates the desired output signal at the desired frequency, which is determined by the control Voltage,  $V_{vco}$ .  $V_{vco}$  is generated by the PD, CP, and LF, by comparing the current output signal to the reference frequency. By adjusting the divider, the output frequency can be changed.

The input signal will be compared to the VCO output by the PD. The PD will give the phase error and it can adjust the VCO output frequency to keep the phases matched. The PLL output frequency is N times higher than the input frequency (e.g. reference frequency) and N can either be an integer or a fractional number. Fig. 2.16 shows the basic topology of a PLL.

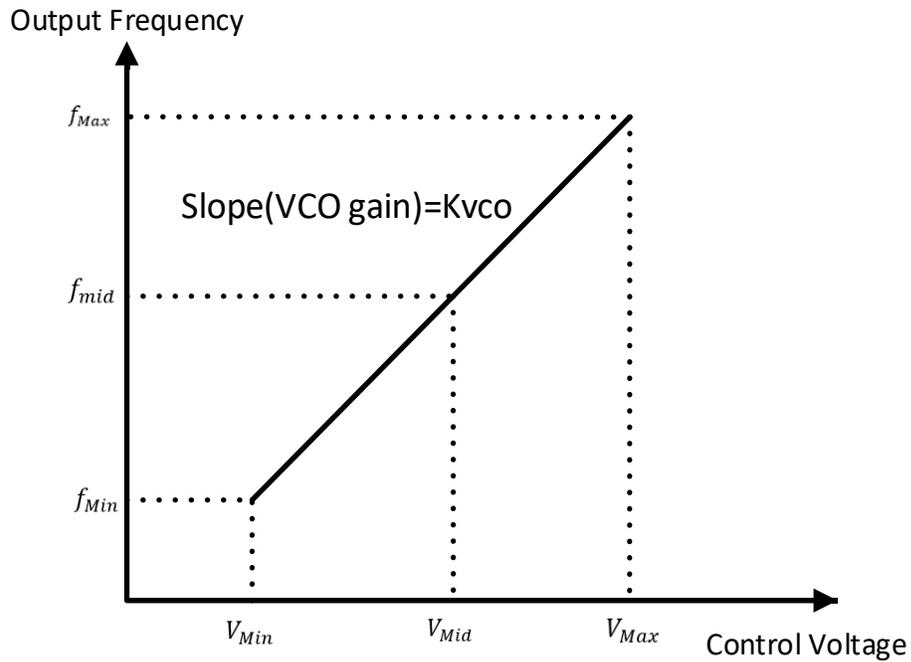


**Fig. 2.16 PLL Basic Topology**

- Voltage Controlled Oscillator

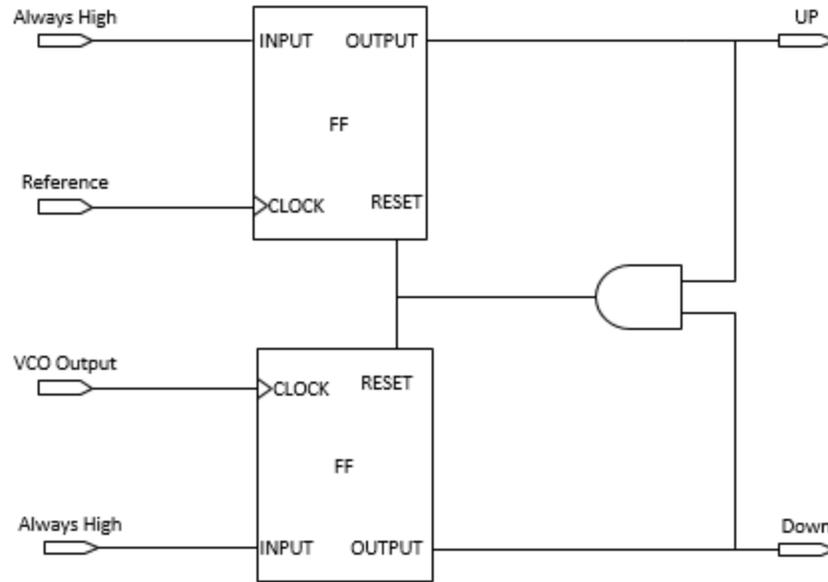
The output frequency of the VCO is related to the control voltage. Fig. 2.17 shows the relationship between the control voltage and the output frequency. Ideally, output frequency and control voltage are linearly correlated as follows:

$$f_{out} = f_{min} + K_{VCO}V_{control} \quad (2.25)$$



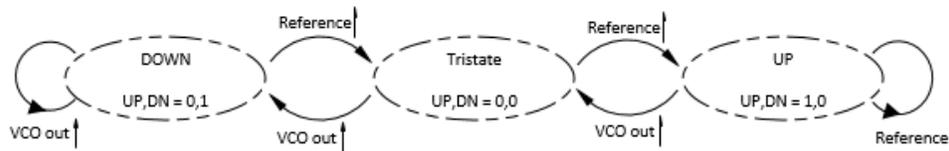
**Fig. 2.17 VCO Output Frequency vs. Control voltage**

The simple phase detector is shown in Fig. 2.18 generates the phase difference between two inputs, one input is the reference frequency, and the other input is the feedback signal from the VCO. The output voltage from the phase detector can be used to control the VCO to maintain a constant phase difference between the two inputs. If the phase of the reference signal is ahead of the VCO output phase, this will speed up the VCO output frequency. On the other hand, if the reference phase is lagging behind the VCO output phase, it will force the VCO output frequency down. These two conditions are called UP and DOWN, respectively. If there is no phase difference, the output will be zero (i.e. in-phase condition).



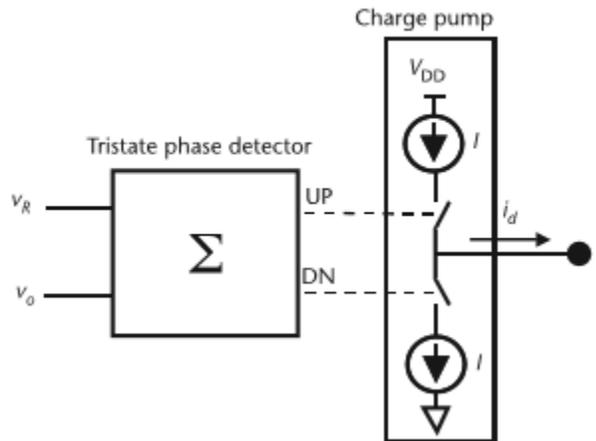
**Fig. 2.18 Phase Detector Structure**

The operation of the phase detector can be described using the state diagram in Fig. 2.19 [50].



**Fig. 2.19 PD State Diagram**

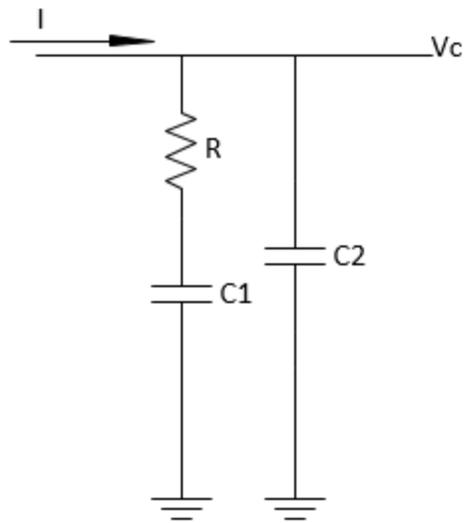
The two digital signals produced by a PD have to be converted back into an analog control signal at the input of the VCO, and the circuit most commonly used to do this is called a charge pump. A charge pump is made of two controllable current sources connected to a common output, as shown in Fig. 2.20. The outputs from the phase detector turn on one of the two current sources, which either charges or discharges capacitors attached to the VCO input [50].



**Fig. 2.20 Tristate Phase Detector and Charge Pump**

- Loop Filter

The VCO is a voltage-controlled component. However, the output of the charge pump is current. Therefore, a loop filter is employed in the circuit [50]. The basic structure of the loop filter is shown in Fig. 2.21.



**Fig. 2.21 Basic Structure of a Loop Filter**

The Low-Pass Filter (LPF) impedance can be expressed by the following equation:

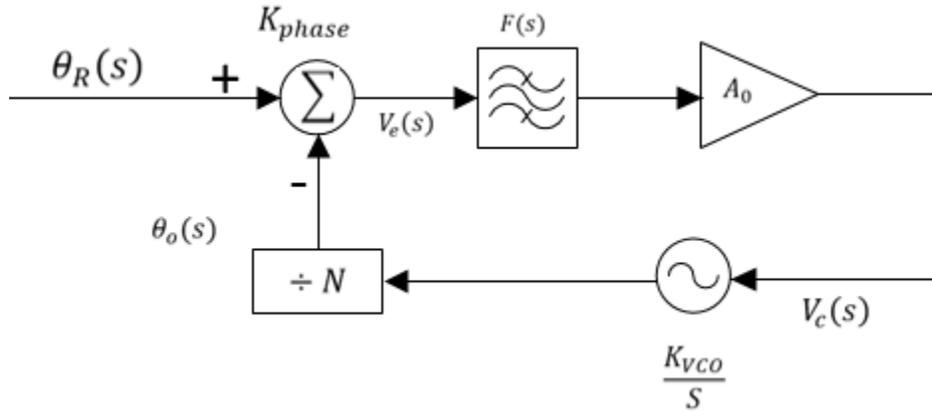
$$V_c = I * \frac{1+sC_1R}{s(C_1+C_2)(1+sC_2R)} \quad (2.26)$$

Where  $C_S = \frac{C_1 C_2}{C_1 + C_2}$ , and  $C_2$  is the high frequency pole.

- PLLs may include a divider between the oscillator and the feedback input to the phase detector to produce a frequency synthesizer. A programmable divider is particularly useful in radio transmitter applications, since a large number of transmission frequencies can be produced from a single stable and accurate, but expensive, quartz crystal–controlled reference oscillator.

### 2.7.1 PLL Transfer Function and Phase Noise Analysis

According to the PLL frequency model shown in Fig 2.22, the transfer function can be written as [50]:



**Fig. 2.22 PLL Model in Frequency Domain**

$$\frac{\theta_o}{\theta_R} = \frac{\omega_n^2 \left( \frac{2\xi}{\omega_n} s + 1 \right)}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2.27)$$

Where  $\omega_n = \sqrt{\frac{K_{VCO} C_1}{2\pi \cdot N}}$  is the natural frequency and  $\xi = \frac{R}{2} \sqrt{\frac{K_{VCO} C_1}{2\pi \cdot N}}$  is the damping constant.

Each component in the PLL system generates noise, such as VCO noise, reference signal noise, frequency divider noise, phase detector noise, charge pump noise, and loop filter

noise. The noise in the PLL system is generally referred to as phase noise, which is a measure of how much the output diverges from an ideal impulse function in the frequency domain [50]. The output signal of the synthesizer can be expressed as:

$$V_{out}(t) = V_o \cos[\omega_{LO}t + \varphi_n(t)] \quad (2.28)$$

Where  $\omega_{LO}t$  is the desired phase of the output,  $\varphi_n(t) = \varphi_p \sin(\omega_m t)$  represents random fluctuations in the phase of the output due to any of the noise sources,  $\varphi_p$  is the peak phase fluctuation, and  $\omega_m$  is the offset frequency from the carrier. Phase noise is reported relative to the carrier power as shown by the following equation:

$$\varphi_n^2(\Delta\omega) = \frac{Noise(\omega_{LO} + \Delta\omega)}{P_{carrier}(\omega_{LO})} \quad (2.29)$$

Noise transfer function in the loop can be broken down into two parts, one for the VCO and one for the rest of the components. The noise for the rest of the components is given by:

$$\frac{\varphi_{noise\ out}(s)}{\varphi_{noise\ I}(s)} = \frac{\frac{IK_{VCO}}{2\pi \cdot C_1} (1 + RC_1s)}{s^2 + \frac{IK_{VCO}}{2\pi \cdot N} RS + \frac{IK_{VCO}}{2\pi \cdot NC_1}} \quad (2.30)$$

The VCO noise can be expressed as [50]:

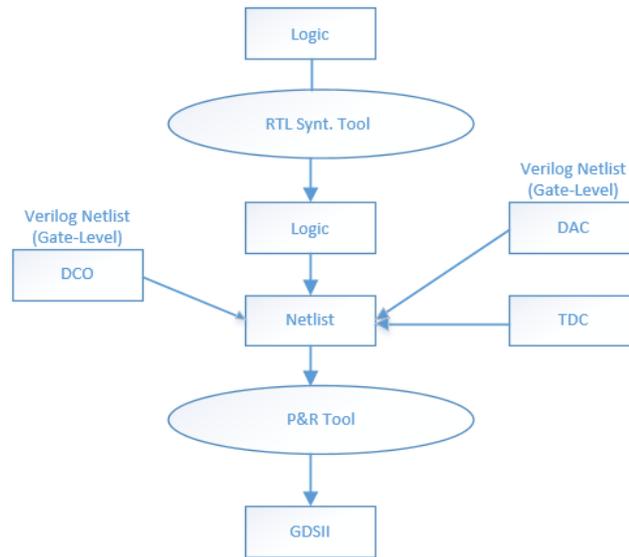
$$\frac{\varphi_{noise\ out}(s)}{\varphi_{noise\ I}(s)} = \frac{s^2}{s^2 + \frac{IK_{VCO}}{2\pi \cdot N} \cdot RS + \frac{IK_{VCO}}{2\pi \cdot NC_1}} \quad (2.31)$$

## 2.8 All Digital Phase Lock Loops

CMOS scaling into the nanometer region has resulted in improved timing accuracy, lower power consumption, and increased density of digital logic gates, as compared to analog circuits which suffers from supply voltage reducing and gate leakage increasing [14] [51]. Based on the advantages of technology improvement, the ADPLLs have been studied and utilized in various areas for a number of years [14] [21] [44] [49].

In mainstream system structures, there are two different topologies of ADPLLs: Time-to-Digital Converter (TDC) based ADPLLs [14], and Phase Frequency Detector (PFD) based, or Bang Bang (BB), ADPLLs [52]. Both topologies have advantages compared to traditional PLLs. ADPLLs tend to reduce chip area, power consumption, and provide higher compatibility, testability, and programmability to the whole system. Most of the functional blocks of ADPLLs can be designed based on a digital design flow as shown in Fig. 2.23 [49].

The design procedure for digital logic circuits is now highly sophisticated with synthesis, layout, and verification of the circuit being automated with design tools. To achieve excellent performance, layout in analog circuits needs to be designed meticulously, which is costly and time-consuming. On the other hand, most of the functional blocks of ADPLLs can be synthesized from standard cells and automatically Place and Routed (P&R) using synthesis tools [14] [49].



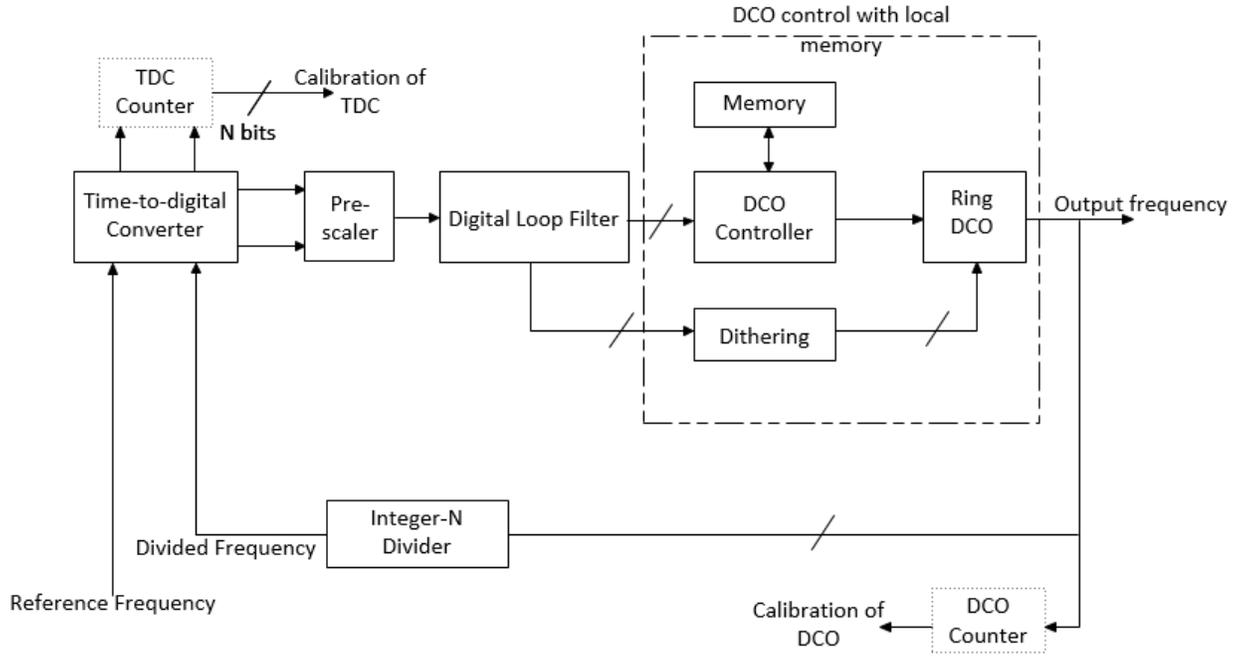
**Fig. 2.23 Digital Design Flow**

The basic ADPLL system includes a TDC, which performs the function of the PD, a digital filter to replace the analog filter, and a DCO to replace the VCO, as shown in Fig. 2.24. The various PLL categories are shown in Table 2 [53]:

**Table 2: Categories of PLLs**

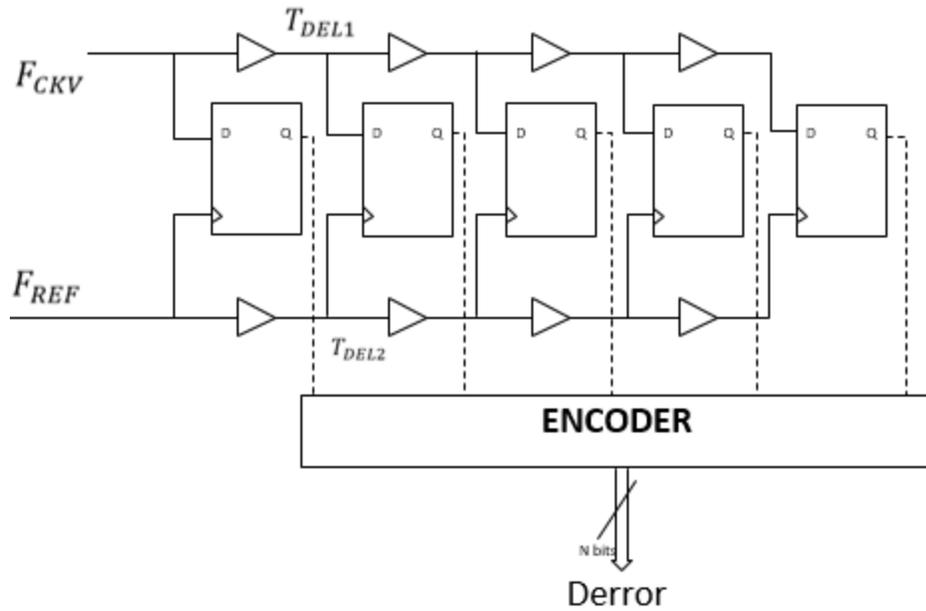
PLLs	Phase detector or comparator	Loop Filter	VCO/DCO
Analog PLL	Analog	Analog	Analog
Digital PLL	Digital	Analog	Analog
All Digital PLL	Digital	Digital	Digital

Fig. 2.24 shows the TDC based ADPLL topology [14].



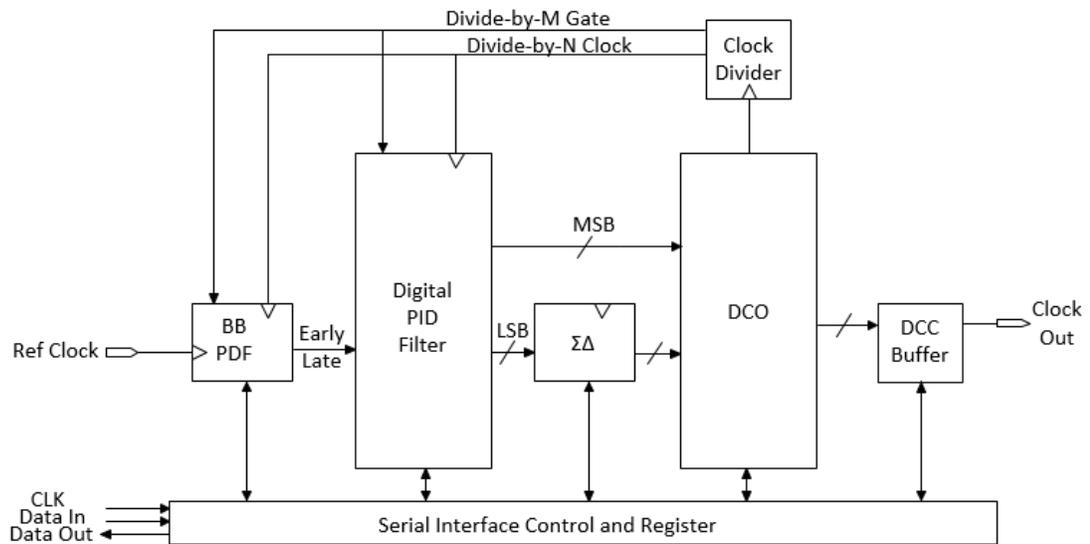
**Fig. 2.24 TDC Based ADPLL Topology**

As mentioned previously, all blocks in the TDC based structure are designed using standard cells. The basic structure of the TDC, shown in Fig. 2.25, is called the Vernier-delay-line based TDC, which has high resolution. The Vernier-delay-line utilizes two delay chains with delays  $T_{DEL1}$  and  $T_{DEL2}$ . The reference signal and divided clock from the DCO output propagate through the delay line, and the time difference between them is decreased by  $T_R = T_{DEL1} - T_{DEL2}$  after each stage [54].



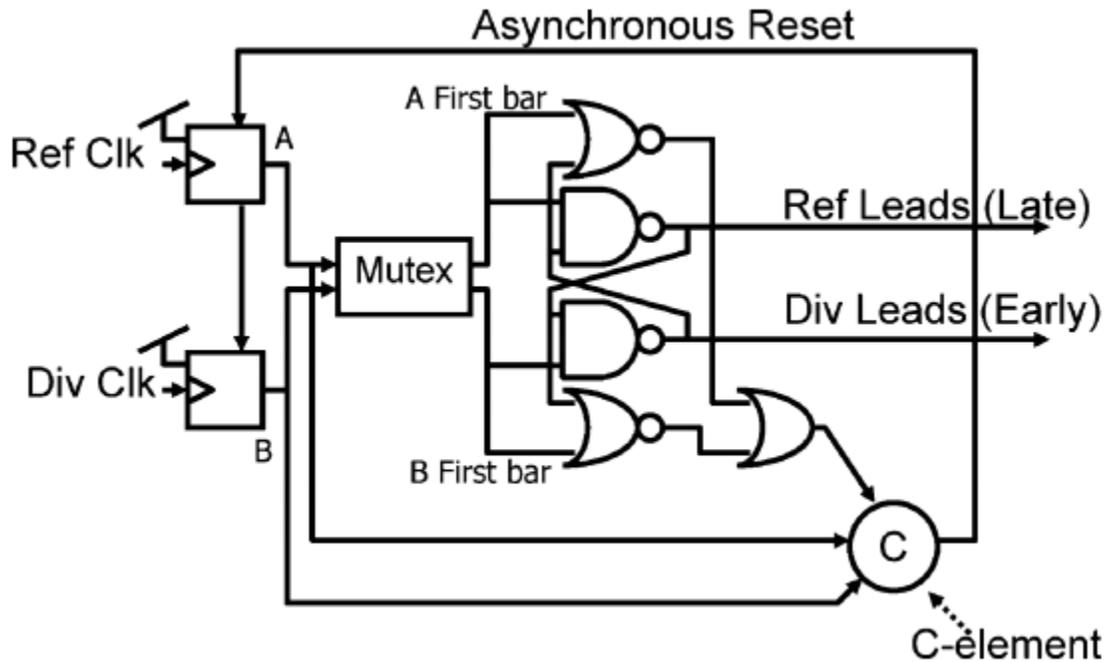
**Fig. 2.25 Vernier-delay-line Based TDC**

In the DCO design, the P&R scheme introduces variations which cause nonlinearity in frequency tuning. Fig. 2.26 shows the TDC-less ADPLL [52].



**Fig. 2.26 TDC-less ADPLL Topology**

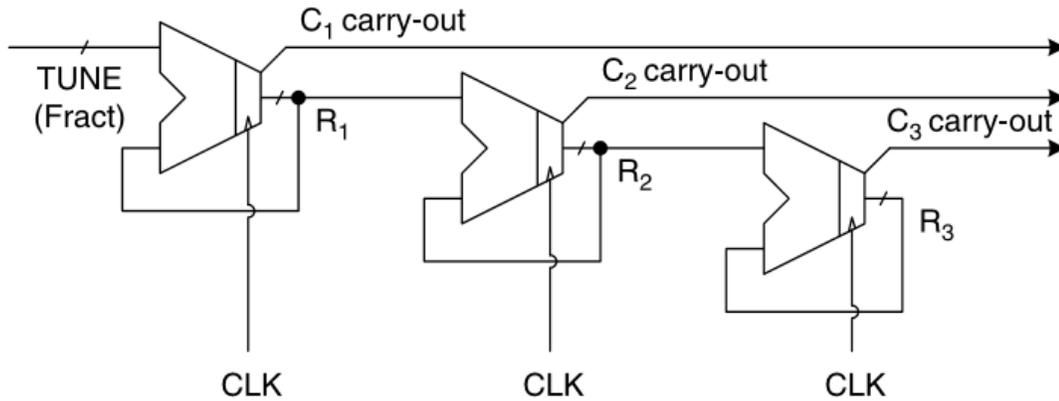
Fig. 2.27 shows the bang-bang frequency and phase detector.



**Fig. 2.27 Bang-bang Frequency and Phase Detector**

The two input latches are used to detect the arrival of an edge on the reference and feedback clocks, respectively. A mutual exclusion element determines which of the two edges arrived first, and stores the result in a set-reset flip-flop. A self-timed reset loop determines that all events have taken place, and generates a reset pulse that prepares the PFD for future edges of the reference and feedback clocks [52].

The structure of the digital Sigma Delta ( $\Sigma\Delta$ ) modulator is shown in Fig. 2.28 [55].

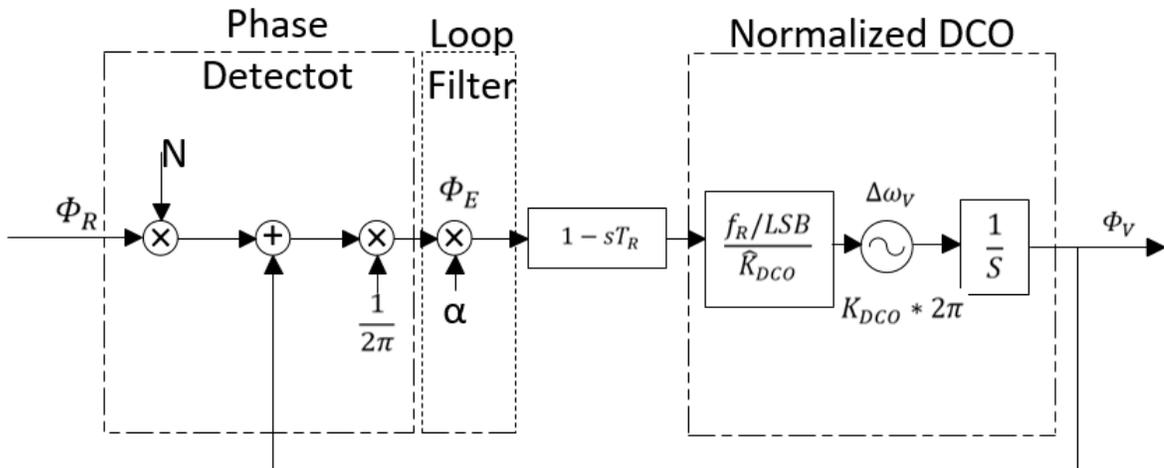


**Fig. 2.28 MASH-3  $\Sigma\Delta$  Modulator Topology**

The  $\Sigma\Delta$  modulator is used to encode the fractional frequency generated by the loop filter into dithering signals for the DCO, effectively increasing its frequency resolution.

### 2.8.1 ADPLL Transfer Function

Fig. 2.29 shows the linearized s-domain model of the Type I ADPLL [56], where the loop filter has been modified into a normalized gain stage, which has only one pole.



**Fig. 2.29 S-domain Mode of a Type I ADPLL**

The open loop transfer function  $H_{ol}(s)$  is [56]:

$$H_{ol}(s) = \frac{1}{2\pi} \alpha \frac{2\pi f_R K_{DCO}}{\bar{K}_{DCO} s} = \frac{\alpha f_R K_{DCO}}{s \bar{K}_{DCO}} \quad (2.32)$$

If the assumption of DCO gain is accurate, then Equation 2.36 can be simplified into:

$$H_{ol}(s) = \frac{\alpha f_R}{s} \quad (2.33)$$

The closed-loop transfer function can be expressed as:

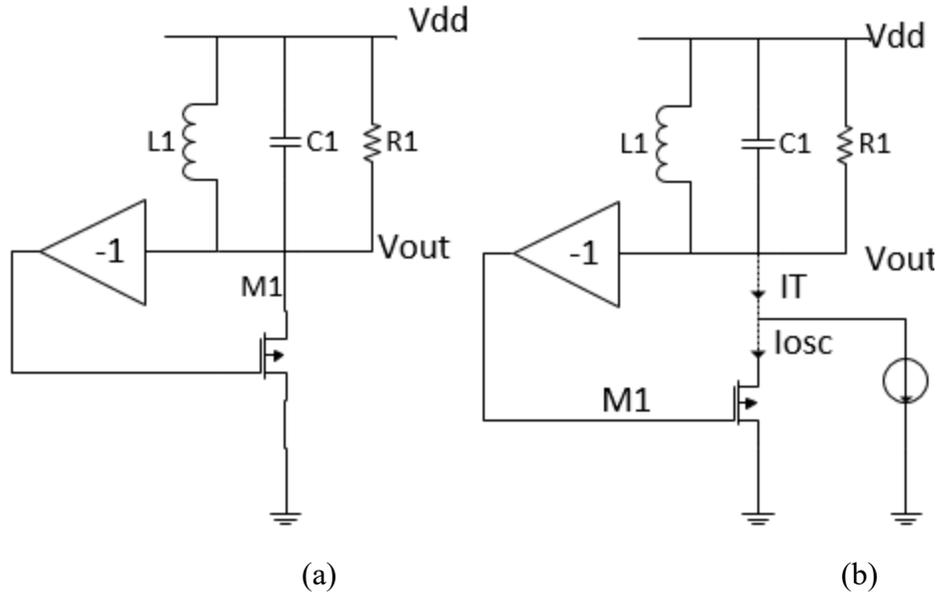
$$H_{cl} = \frac{NH_{ol}}{1+H_{ol}} = N = \frac{\frac{N\alpha f_R}{s}}{1+\frac{\alpha f_R}{s}} \quad (2.34)$$

All components in this section are assumed to be noiseless. In practice, the reference source (i.e. crystal oscillator), loop filter, phase detector, and DCO will add extra noise to the system. The higher order noise analysis can be found in [50] [56]. More design details about ADPLLs can be found in [56]

## 2.9 Basic Theory of Injection Locking

In the early 17<sup>th</sup> century, Christiaan Huygens found that the pendulums of two clocks on a wall moved in unison if the two clocks were close each other. The pendulums would eventually shift to the same frequency and would be 180<sup>o</sup> out of phase. This might be the first record of frequency synchronization or injection locking. Many years later, injection locking became useful in a number of applications, including frequency division, quadrature generation, and oscillators with finer phase separations [57]. The injection locking phenomena has been researched by Adler, Kurokawa, and many others. In a traditional LC oscillator, shown in Fig. 2.30(a), the resonant frequency is given by:

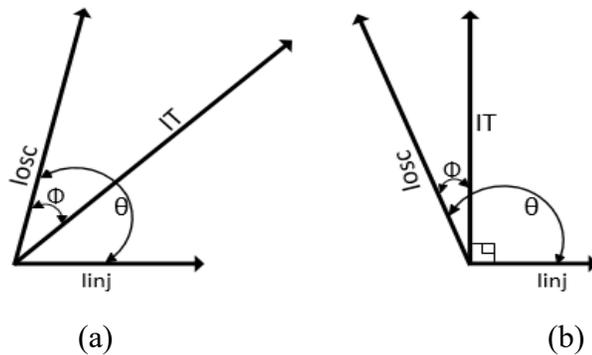
$$\omega_0 = 1/\sqrt{LC} \quad (2.35)$$



**Fig. 2.30 (a) Conceptual oscillator. (b) Frequency Shift by Injection Current**

If we generate  $\phi_0$  by adding a sinusoidal current to the drain current of M1 and if the amplitude and the frequency of  $I_{inj}$  are chosen properly, the circuit will oscillate at  $\omega_{inj}$  rather than  $\omega_0$  and injection locking occurs [57]. From the phasor diagram shown in Fig 2.34, the equation of locking range can be deduced as follows:

$$\sin\phi = \frac{I_{inj}}{I_T} \sin\theta \tag{2.36}$$



**Fig. 2.31 Phase Difference between Input and Output**

After expanding Equation 2.40 we get:

$$\sin\phi = \frac{I_{inj}\sin\theta}{\sqrt{I_{osc}^2 + I_{inj}^2 + 2I_{osc}I_{inj}\cos\theta}} \quad (2.37)$$

This equation can reach a maximum of:

$$\sin\phi_{0,max} = \frac{I_{inj}}{I_{osc}} \quad (2.38)$$

In Fig. 2.20(a), the phase difference between the input and the output reaches  $90^\circ + \phi_{0,max}$ , and forms the phase shift of LC resonance as shown below:

$$\tan\alpha \approx \frac{2Q}{w_0}(w_0 - w_{inj}) \quad (2.39)$$

Combining Equations 2.41 to 2.43, the new equation is given as:

$$\omega_0 - \omega_{inj} = \frac{\omega_0}{2Q} * \frac{I_{inj}}{I_{osc}} * \frac{1}{\sqrt{1 - \frac{I_{inj}^2}{I_{osc}^2}}} \quad (2.40)$$

If  $I_{inj} \ll I_{osc}$ ,  $\phi_0$  is very small, and  $\sin\phi_0 \approx \tan\phi_0$ , Equations 2.38 and 2.40 can be simplified into:

$$\sin\theta = \frac{2Q I_{osc}}{\omega_0 I_{inj}} (\omega_0 - \omega_{inj}) \quad (2.41)$$

The locking range is based on the injection level,  $I_{inj}$ . The general equation of locking range is given by:

$$w_L = \frac{w_0}{2Q} * \frac{I_{inj}}{I_{osc}} \quad (2.42)$$

Further details about injection locking can be found in [21] [58] - [63].

## 2.10 Chapter Conclusion

Compared to traditional analog PLLs, ADPLLs have many advantages. ADPLLs are highly integrable in digital systems, which occupy smaller chip area and consumes less power. ADPLL design is less susceptible to layout issues compared to analog design.

Traditional PLLs tend to have better phase noise performance than ADPLLs, but designers have tried to address this with elaborately designed fractional ADPLLs [59] - [63].

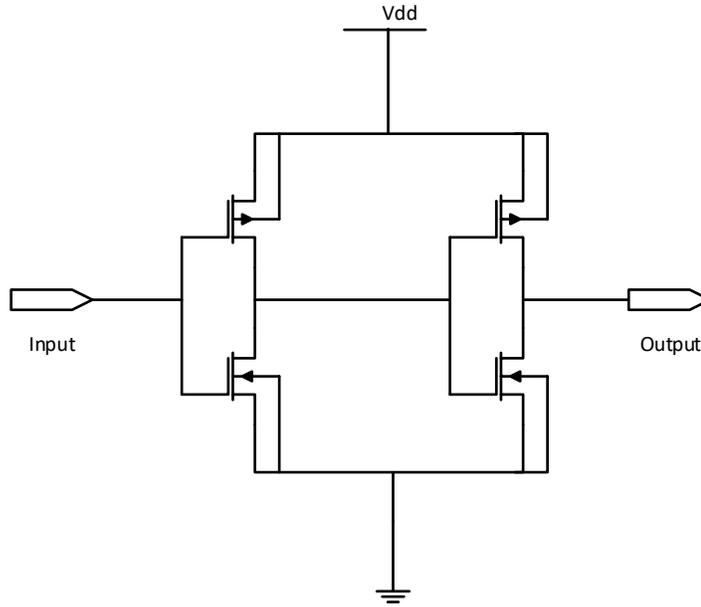
## **3 Chapter: Design and Architecture**

### **3.1 Introduction**

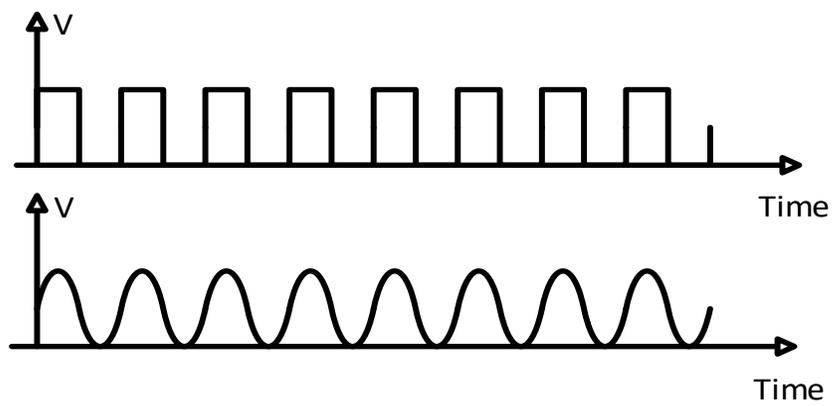
System level architecture, digital function blocks and the pulse injection technique are described in this chapter. Section 3.2 proposes the pulse injection locked methodology and the advantages of this technique. A novel system-level architecture is illustrated in section 3.3. The individual functional blocks are described in Sections 3.4 through 3.8, including oscillator structure, pulse generator design, Serial to Parallel (S/P) shift register design, clock tree design and control method.

### **3.2 Pulse Injection Technique**

Since the first observation of the injection locking phenomenon [58] – [63], it has been widely utilized in various areas, such as frequency synthesizers, clock distribution and clock recovery. Before discussing the ideal pulse, an ideal square wave is introduced first to demonstrate the relation between different sub-harmonics and output power. Typically, the crystal oscillator generates a low noise signal. However, this kind of oscillator is usually operating between several megahertz to hundreds of megahertz, which is hard to incorporate with mainstream wireless communication operating frequencies (from hundreds of megahertz to several gigahertz). In some cases, the output signal of a crystal oscillator is a sinusoidal wave which further complicates direct use for current design. In those cases, an input buffer (e.g. two inverters connected in series) can be deployed between the crystal oscillator and the pulse generator, as shown in Fig. 3.1. Hence a sinusoidal wave can be converted into a square wave, as shown in Fig. 3.2.

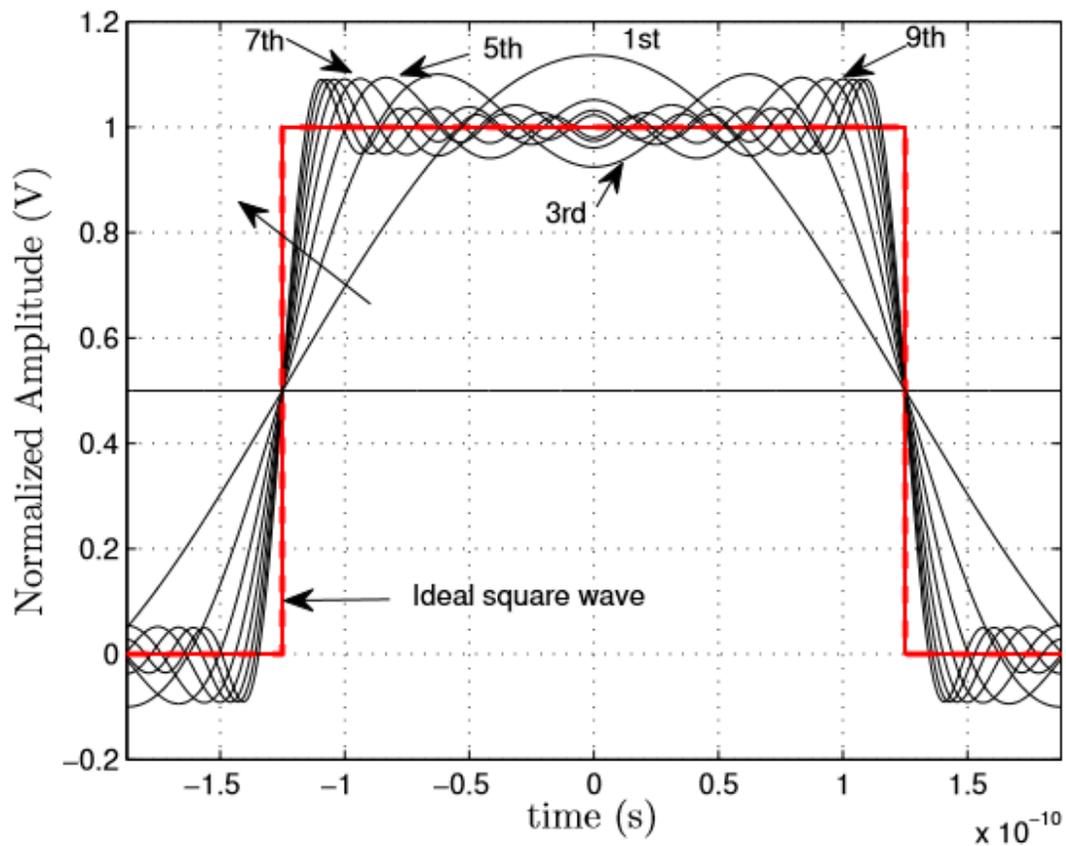


**Fig. 3.1 CMOS Buffer**



**Fig. 3.2 Sinusoidal Wave Shaping into Ideal Square Wave**

The Sub-harmonic injection-locked technique is an effective approach to address the issue of the low crystal oscillator frequency. As is well known, an ideal square wave contains higher order harmonic signals, as shown in Fig. 3.3 [64].

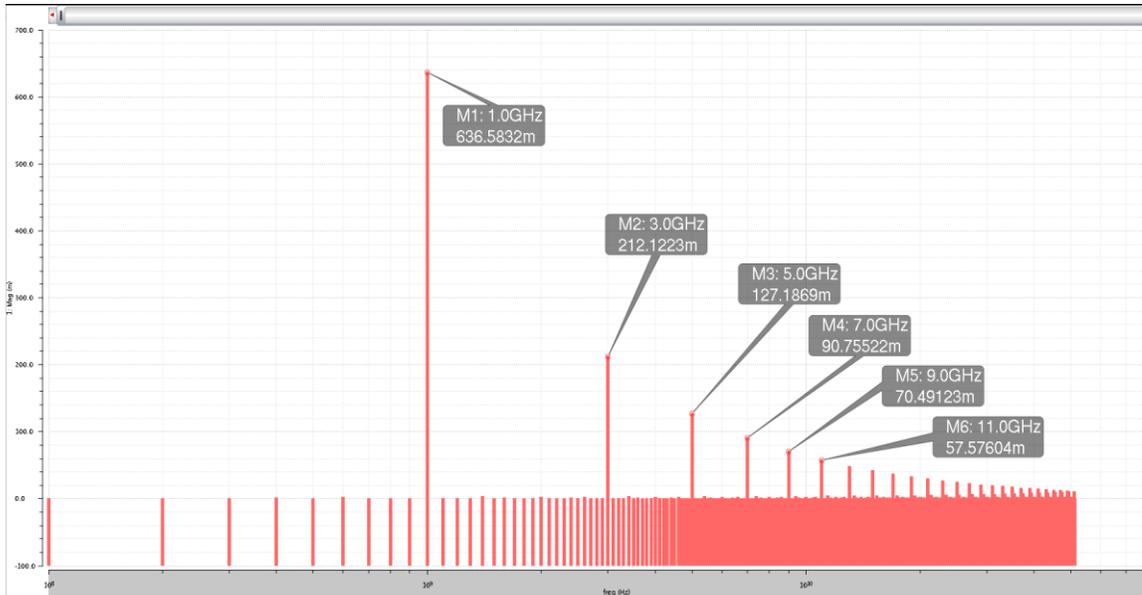


**Fig. 3.3 Fourier Decomposition of Ideal Square Wave**

Fig. 3.3 can be explained through Fourier series given by equation 3.1:

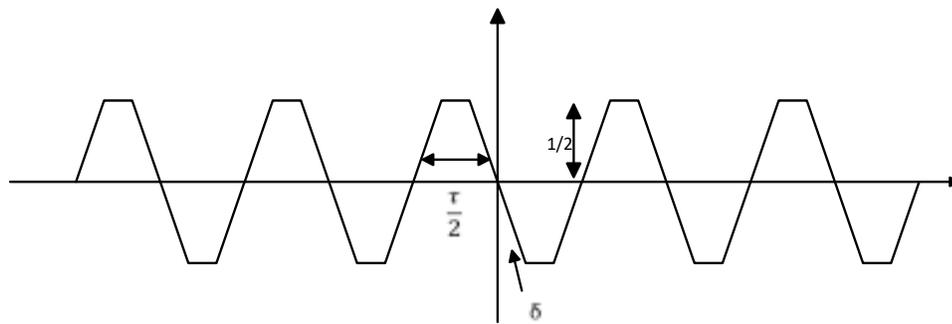
$$x(t) = \frac{1}{2} + \frac{2}{\pi} \left( \cos t - \frac{1}{3} \cos 3t + \frac{1}{5} \cos 5t - \frac{1}{7} \cos 7t + \frac{1}{9} \cos 9t \right) \quad (3.1)$$

Although an ideal square wave contains higher order harmonic signals, it suffers from power degradation when the harmonics order increases, as shown in Fig 3.4.



**Fig. 3.4 1GHz Square Wave after Discrete Fourier Transform in Cadence**

In reality, however, there is no ideal square wave, as shown in Fig 3.5, and non-ideal square waves have finite rise/fall times which affect the magnitude or the power of harmonic signals.



**Fig. 3.5 Non-ideal Square with Finite Rise/Fall Time**

The general equation to describe a non-ideal square wave is given by equation 3.2 [63]:

$$F(j\omega) = \frac{1}{(j\omega)^2} H(j\omega) \quad (3.2)$$

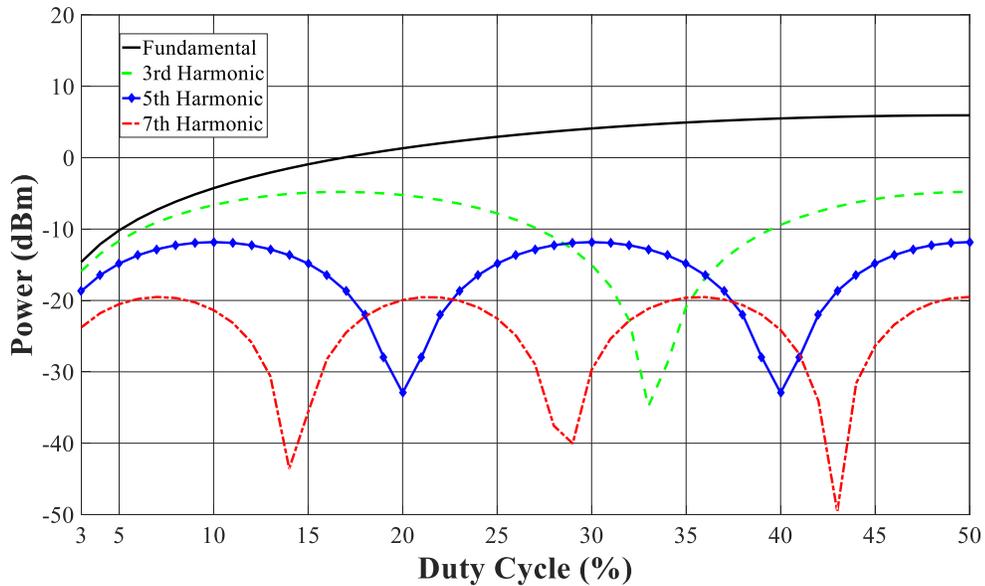
To get the result of  $F(j\omega)$ ,  $H(j\omega)$  can be calculated first with the assumption of  $\omega = n\omega_0$ :

$$H(j\omega) = \int_0^T \frac{d^2 f(t)}{dt^2} e^{-jn\omega_0 t} dt \quad (3.3)$$

According to complex deduction, the  $N^{\text{th}}$  harmonic coefficient of non-ideal square wave can be expressed as:

$$C_{n\_real} = \frac{\varepsilon}{T} \cos\left(-n\pi \frac{\delta+\varepsilon}{T}\right) \text{sinc}\left(n\pi \frac{\delta}{T}\right) \text{sinc}\left(n\pi \frac{\varepsilon}{T}\right) \quad (3.4)$$

Where  $\varepsilon$  is the effective duty cycle of the signal period, and  $\delta$  is the rise/fall time. Moreover, the pulse width or duty cycle also affects the injection signals' power. From [63], the amplitude has been normalized to 1v for simplification of analysis.



**Fig. 3.6 The Relation Between Duty Cycle and Power Degradation**

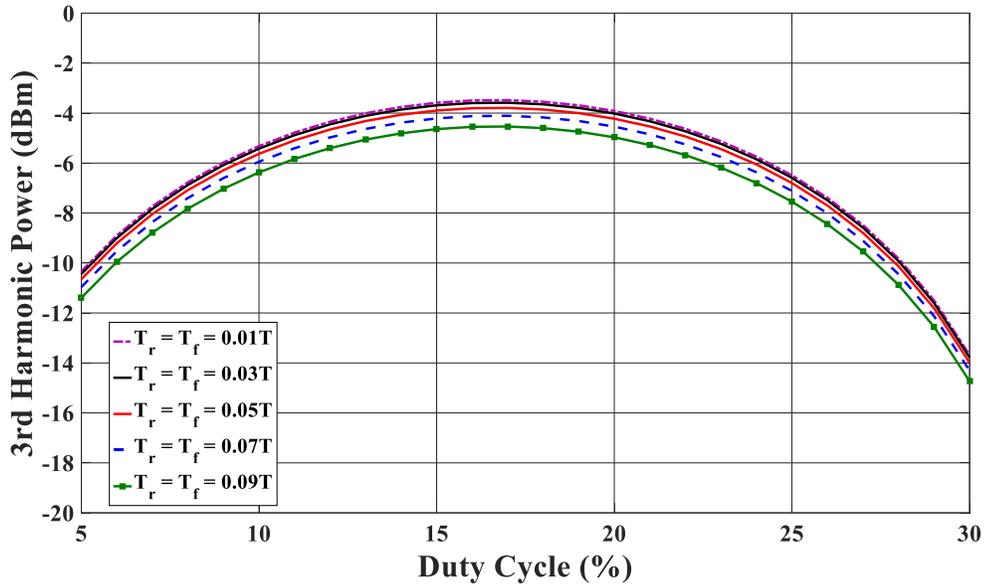
As shown in Fig. 3.6 [63], the relationship between duty cycle and power of different harmonics can be derived. The signal power deviation of the higher order harmonics, such as 7<sup>th</sup> harmonic is generally more sensitive to the pulse width compared to lower harmonics, as seen with the 3<sup>rd</sup> and 5<sup>th</sup> harmonics for instance. In Fig. 3.6, the maximum signal power changes according to different harmonics. Moreover, the power of the square wave signal which will be used as an injection signal for an injection locked

DCO and it will be discussed in the following part, affects the phase noise performance and the locking bandwidth of the locked signals [57]. As described by Equation 3.5:

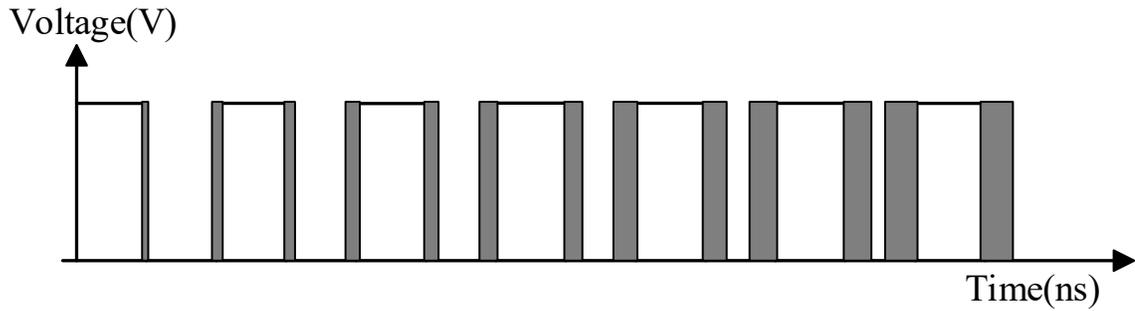
$$\Delta\omega = \frac{\omega_o I_{inj}}{2Q I_{osc} \sqrt{1 - \left(\frac{I_{inj}}{I_{osc}}\right)^2}} \quad (3.5)$$

Where  $\omega_o$  the free running frequency of the DCO, Q is the quality factor of the oscillator,  $I_{inj}$  and  $I_{osc}$  are the current of injected signal and free running oscillator respectively, and  $\Delta\omega$  is the locking bandwidth of the target harmonic.

Fig. 3.7 [63] shows the power of the 3<sup>rd</sup> harmonic with the relationship between duty cycle and rise/fall times. The power of the 3<sup>rd</sup> harmonic has a maximum value at 17% duty cycle and with a minimum rise/fall time (assumed to be 10% of the total period for this analysis). As a result, the proper duty cycle should be found to achieve the maximum harmonic injection power. More analysis on the relation between pulse width and harmonic power is described in [63]. Based on the relationship between the signal power, harmonics and duty cycle, the current design can be locked from the 2<sup>nd</sup> up to the 15<sup>th</sup> sub-harmonic at the expense of decreasing the locking bandwidth and phase noise performance with increasing sub-harmonic order (i.e. lower injection frequency). More details are shown in Chapter 5.

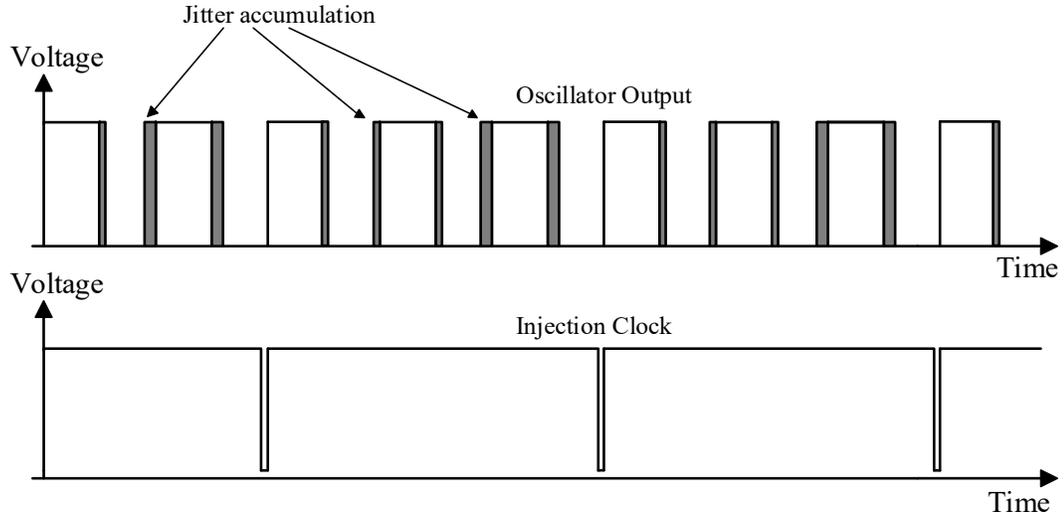


**Fig. 3.7 3<sup>rd</sup> Harmonic Power as a Function of Injection Signal Duty Cycle and Rise/Fall Time**



**Fig. 3.8 Free Running Oscillator Wave Form with Jitter Accumulation**

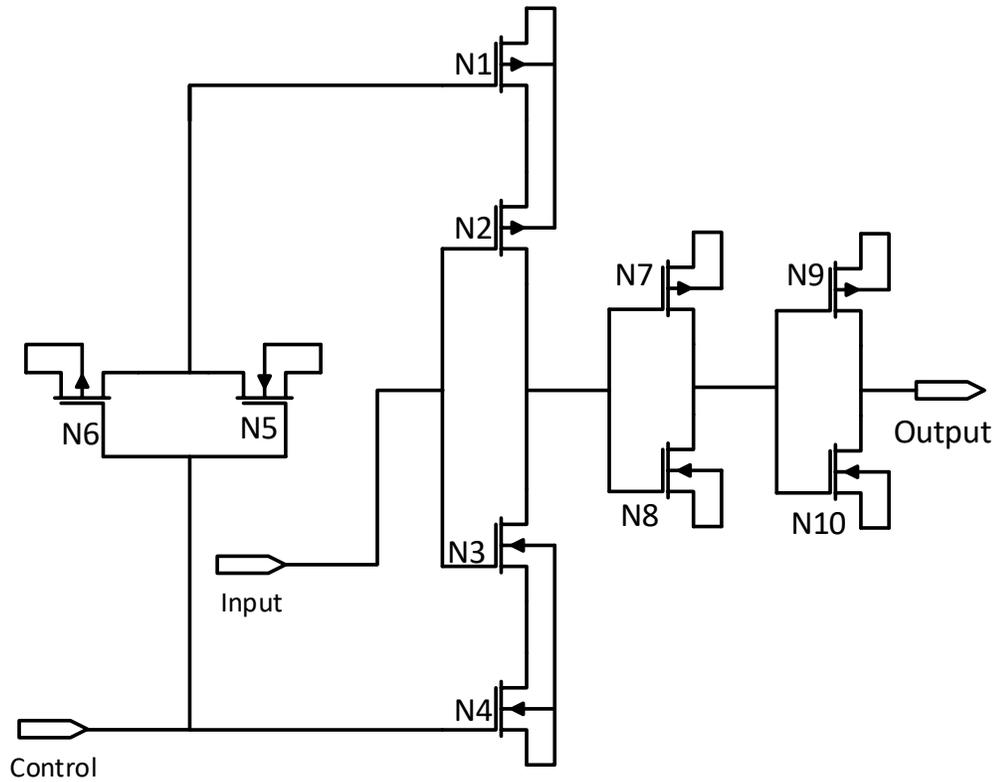
As mentioned in previous work [26], oscillator jitter accumulates due to the uncertainty in the earlier oscillator transitions affecting all the following transitions. Therefore, the signal quality degrades. As shown in Fig 3.8, jitter of the free running oscillator will accumulate at both edges of the oscillator output.



**Fig. 3.9 Pulse Injection Method in Time Domain**

The time domain illustration of the proposed pulse injection-locking method is shown in Fig. 3.9. When the output of the free running oscillator operates within the locking bandwidth, the output frequency of the free running oscillator will be locked to the corresponding sub-harmonic. In Fig. 3.9, the ring oscillator is locked to the 3<sup>rd</sup> sub-harmonic of the injection signal. For 3<sup>rd</sup> sub-harmonic injection locking, the injection clock has small duty cycle compared to the oscillator period. Moreover, the oscillating frequency of the free running oscillator is approximately three times higher than the injection clock. In the first and second clock cycles, random jitter is accumulated. During the third clock cycle, the reference pulse is injected into the free running oscillator which forces the free running oscillator to stop operating until the rising edge of the pulse injection clock. This greatly reduces the jitter accumulation such that the noisy signal of the DCO is corrected by the low noise injection edge [49]. However, random jitter will accumulate in the following cycles. Therefore, the reference pulse needs to be injected into the oscillator periodically.

In this work, the injection port is designed as an active high tristate inverter with output buffer, as shown in Fig 3.10.

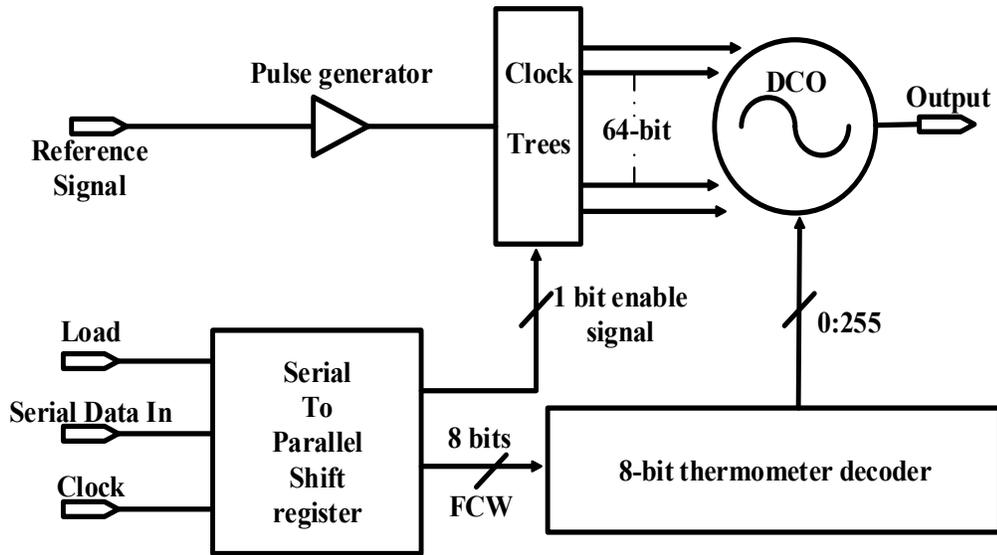


**Fig. 3.10 Active High Tristate Inverter with Output Buffer**

As shown in Fig. 3.10, the tristate inverter is composed of transistors N1-N6 and transistors N7-N10 form an output buffer. When the control signal goes high, the output will invert the input. If the control signal goes to low, N1 and N4 will be turned off, forcing the output to stop inverting the input.

### 3.3 Top Level Design

The top level design of the injection locked oscillator is described in this section, as shown in Fig. 3.11. More details about building blocks will be discussed in the following sections.



**Fig. 3.11 Proposed Injection Locked Ring Oscillator of the High-Frequency Oscillator**

The reference signal passes through the pulse generator to generate a low duty cycle injection clock (i.e. periodic narrow injection pulses). The clock tree uses a clock distribution method so that the injection clock arrives at the different injection ports of a DCO within 8pS of each other, based on simulation results. If the arrival time difference of injection signals is too large, the phase noise performance of the system and the locking band width will be affected. The proposed DCO is composed of a ring oscillator based on the design of [65]. The enable signal of each tristate inverter is connected to a thermometer decoder output whose input is the Frequency Control Word (FCW). The FCW comes from

a Serial to Parallel (S/P) shift register that can be loaded using external signals. The 8-bit input signal representing the FCW is decoded into a 256-bit thermometer code.

The output frequency out the ILRO is expressed as:

$$f_o = N * f_{ref} \quad (3.6)$$

Where  $f_o$  is the output frequency of ILRO,  $f_{ref}$  represents the operating frequency of the injection clock, and N is the order of the injection locked harmonic (e.g. N can be any integer number).

Based on the High-Frequency ILRO structure, a Low-Frequency ILRO was built in the proposed work that is using the same topology and injection locked method as the High-Frequency ILRO. However, it will not be discussed in detail due to its phase noise performance.

### 3.4 Digitally Controlled Ring Oscillator Design

The traditional ring oscillator has the fixed oscillating frequency  $f_{out} = \frac{1}{2*t*n}$ , where t represents the propagation delay for a single inverter and 'n' represents the number of inverters in the closed loop chain (in a single output structure, n must be an odd number). In the differential structure, n can be any positive number), as shown in Fig. 3.12.

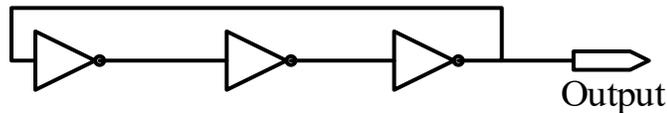
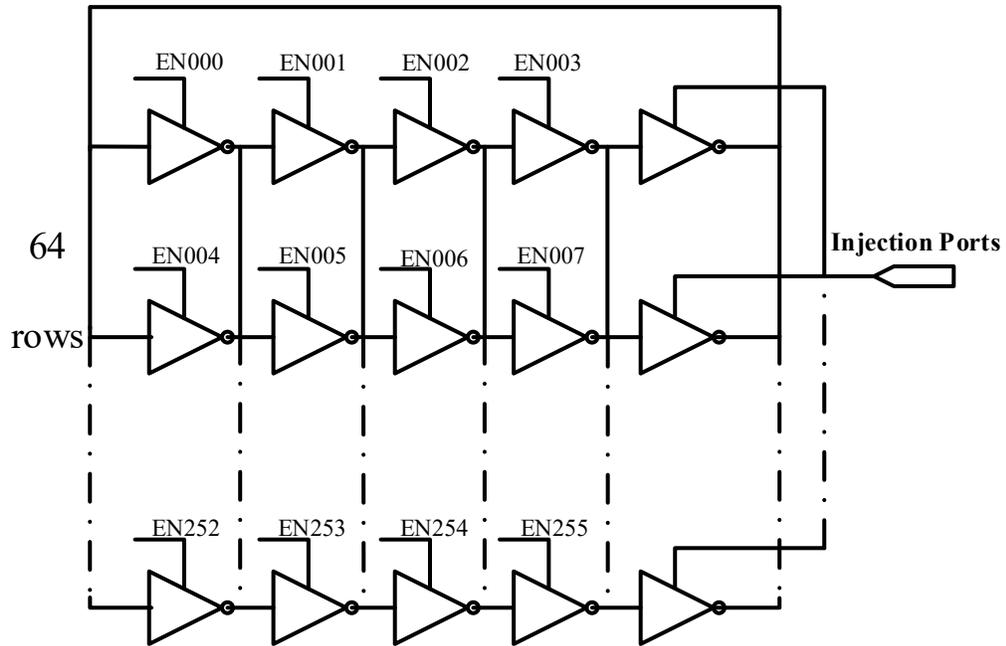


Fig. 3.12 Three Stages Ring Oscillator

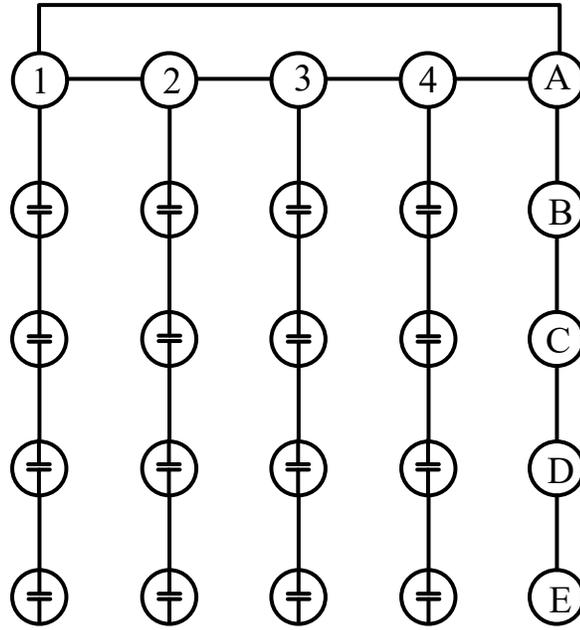
The proposed DCO is composed of a ring oscillator based on the design of [65], where the oscillator is a matrix of 5 columns and 64 rows of tristate inverters for the high frequency output oscillator, as shown in Fig. 3.13.



**Fig. 3.13 Oscillator Array of High Frequency Output**

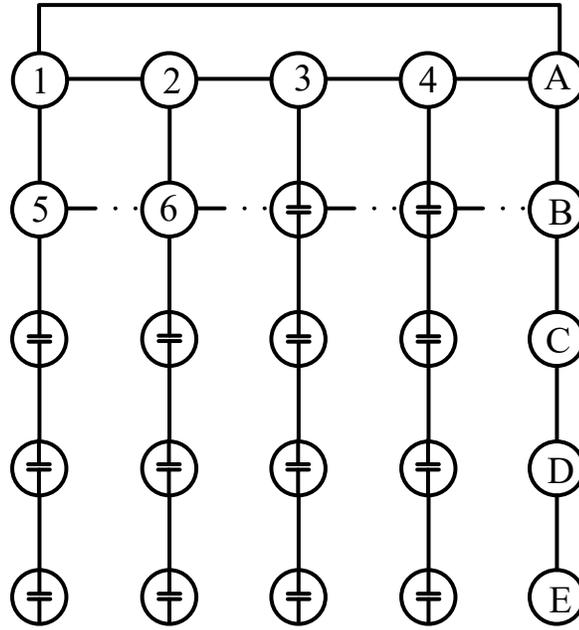
Each component of the oscillator array is a tristate inverter as shown in Fig. 3.10. It should be noted that the maximum output frequency is limited by the vendor supplied digital library. The proposed work utilizes the academic version of the library which does not include a tristate inverter standard cell. The tristate inverter function was accomplished using a tristate buffer in series with an inverter. Therefore, the number of delay cells and load capacitance in the oscillator matrix is significantly increased thereby reducing the output frequency. Changing the load capacitance of the tristate inverters can adjust the output frequency [47] [49]. For example, as shown in Fig.3.14.





**Fig. 3.15 Maximum Load Capacitance Array**

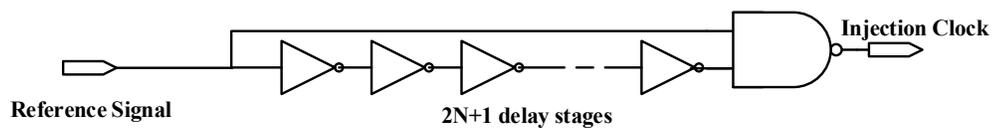
At the second tuning step, more tristate inverters are enabled, as shown in Fig 3.16. In the second row, frequency tuning units 5 and 6 are enabled while the first row remains enabled. At this time, the total load capacitance value remains approximately the same, however, the total drive strength of the DCO increases, hence the output frequency increases. At the highest tuning step, all tristate inverters are enabled, which leads to the peak drive strength and the maximum output frequency of the DCO.



**Fig. 3.16 The Second stage of Frequency Tuning**

### 3.5 Pulse Generator

A pulse generator block was designed for pulse injection-locked method. As mentioned before, the pulse width determines the power of the harmonic signals [63]. The pulse generator structure is shown in Fig. 3.17.

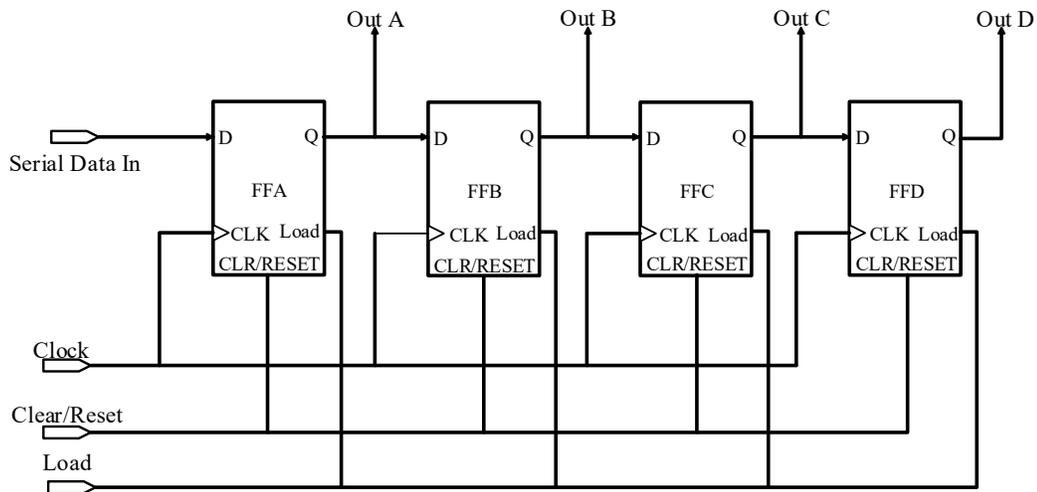


**Fig. 3.17 Pulse generator**

The pulse width of the injected pulse is controlled by a voltage-controlled delay line. The difference in delay along the inverter path and injection signal provides a narrow negative going pulse at the output of the NAND gate. The width of the pulse can be controlled by varying the number of delay stages or varying the supply voltage of the delay stages.

### 3.6 Serial to Parallel Shift Register

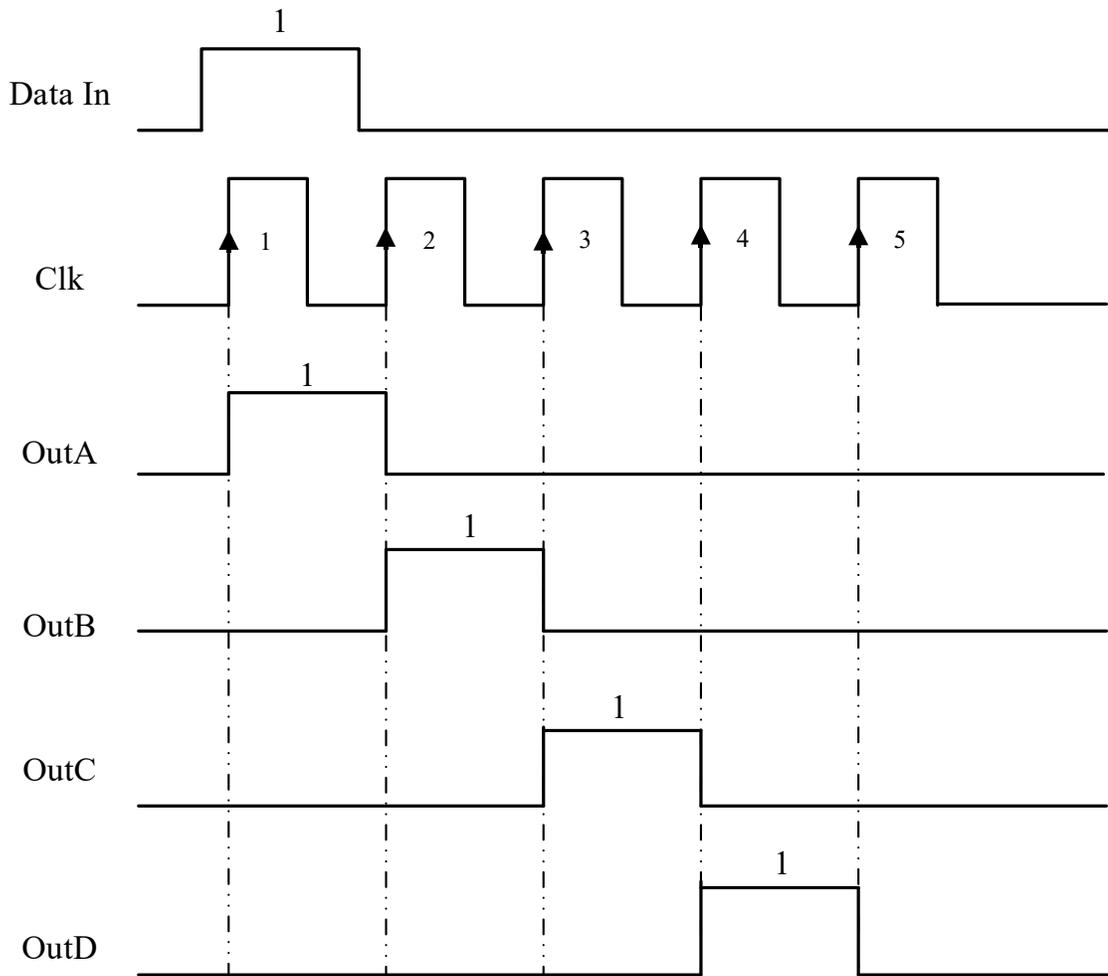
One of the important function blocks is Serial to Parallel (S/P) shift register. This digital block has 21 control bits for frequency tuning (18 bits), and output selection (3 bits). If each of FCW related ports is allocated an individual Input or Output (I/O) port on the chip, the number of chip pins would be doubled (24-pin package form into 40-pin package). Hence, a common S/P shift register was chosen to reduce the number of package pins. Fig 3.18 shows the architecture level of S/P shift register.



**Fig. 3.18 4-bit Serial In Parallel Out (S/P) Shifter Register**

The operation flow is depicted as follows. Under the initial conditions, all flip-flops (FFA to FFD) are in the reset state, and the outputs from Out A to Out D are "0", therefore no data is output. If a logic high passes through the input pin, then on the rising edge of the 1<sup>st</sup> clock the output A will be set to logic "1" with all other outputs remaining in the previous logic value "0". In the 2<sup>nd</sup> clock cycle, the input data remains as logic "1", then both output A and B have the output logic of "1", while outputs C and D remain constant at logic level "0". During the 3<sup>rd</sup> and 4<sup>th</sup> clock cycles, two "0" are sent to the input port sequentially. The output logic of A, B, C, and D will be "0-0-1-1" respectively. The two logic "1" is shifted

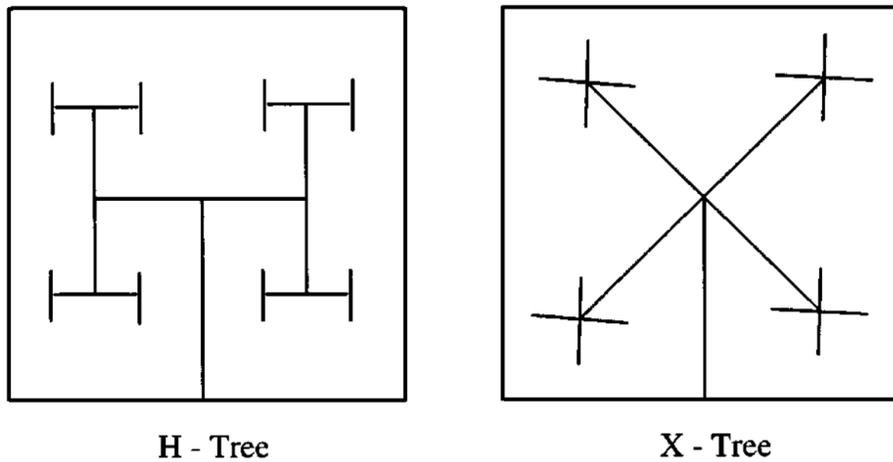
one place along the register to the right as it is now at Out C and Out D. At the same time the Load command is triggered, changing from logic "0" to "1". Then the data has been converted from a serial data input signal to a parallel data output. The truth table and following waveforms show the propagation of the logic "1" through the register from left to right as shown in the Fig 3.19. The proposed work requires 21 control bits, therefore, the 4 bit S/P shift register was modified to include 21 bits.



**Fig. 3.19 4-bit Shift Register Timing Table**

### 3.7 Clock Tree Design

In section 3.4, the digitally controlled oscillator core was discussed, which employs a large numbers of tristate inverters (i.e. the oscillator includes 5 columns by 64 rows). The oscillator also includes 64 injection ports. Injection signal arrival time at the different ports is a critical issue that can affect the injection-locking performance. Under ideal conditions, all injection signals should arrive at the injection port at the same time, with only a slight time difference on the scale of picoseconds. An appropriate clock distribution method can mitigate signal arrival time mismatch. Two similar approaches to satisfying the clock distribution are H-tree and X-tree designs, as shown in Fig 3.20 [66].



**Fig. 3.20 Symmetric H-tree and X-tree Clock Distribution Method**

However, X-tree and H-tree distribution methods are difficult to implement in this design because all of the components are Place and Routed (P&R) automatically, and all the injection ports are placed on the edge of the oscillator block which makes it difficult to form the shape of either an X tree or an H tree.

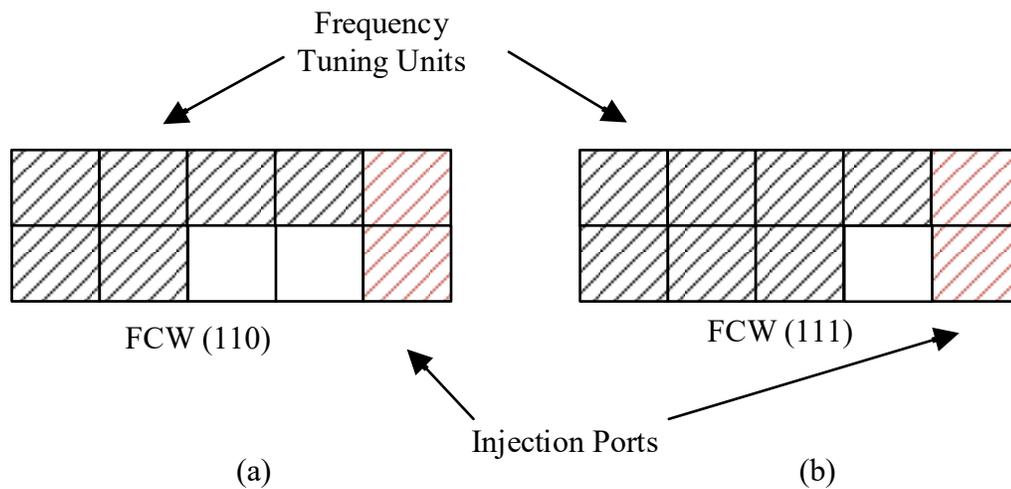
An alternate approach is using a symmetric buffer tree, as shown in Fig. 3.21.



**Table 3 3-bit Binary to Thermometer Decoder Truth Table**

Binary In	Thermometer Code Out
000	0000000
001	0000001
010	0000011
011	0000111
100	0001111
101	0011111
110	0111111
111	1111111

Fig. 3.22 shows the operating steps for this. Fig. 3.22(a) depicts the input binary control bits as 110 and Fig. 22(b) shows the control bits as 111. To control 256 components of oscillator, an 8-bit S/P shift register is used to import all data into the system, and it takes 8 clock cycles to complete importing data.



**Fig. 3.22 Frequency Control Words and Enabled Components**

### **3.9 Conclusion**

The pulse injection locking technique and top level design of the proposed work are discussed in this chapter. The relationship between pulse width and output power for the different harmonic components is discussed in numerical equations and simulation results. Moreover, the details of the building blocks are discussed in this chapter and operating principle of individual blocks and oscillator tuning method are depicted in this chapter as well.

## 4 Chapter: ILRO Simulation Results

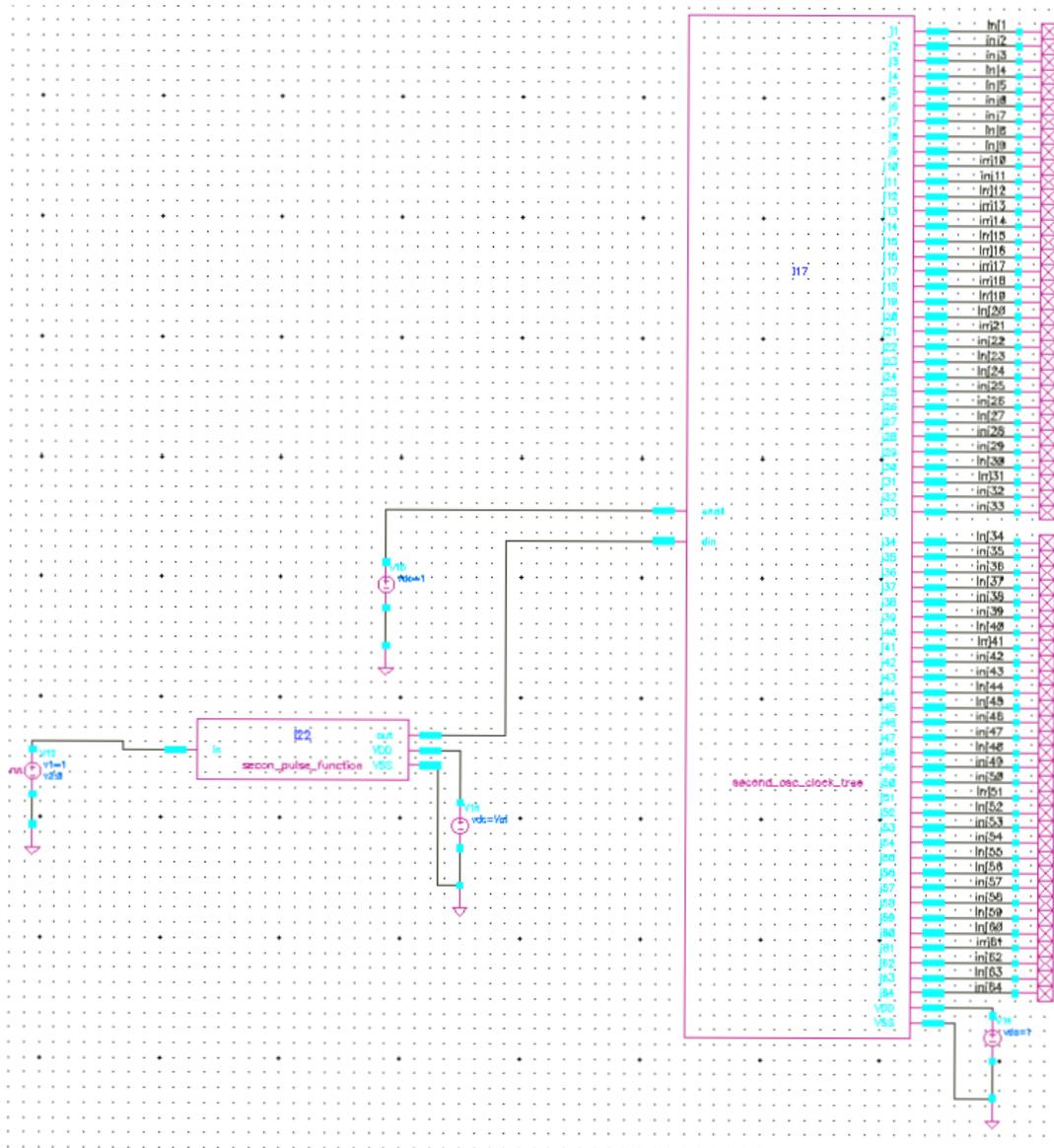
Chapter 4 focuses on the ILRO simulation results. The simulation results of the individual digital blocks are shown in sections 4.1 through 4.3. In section 4.4 the ILRO simulation results are illustrated in detail including, free running frequency tuning range and frequency variation with different supply voltages. Moreover, the locking bandwidth and the injection locking phase noise for different sub-harmonics are also shown in Section 4.4.

### 4.1 Clock Tree Simulation Results

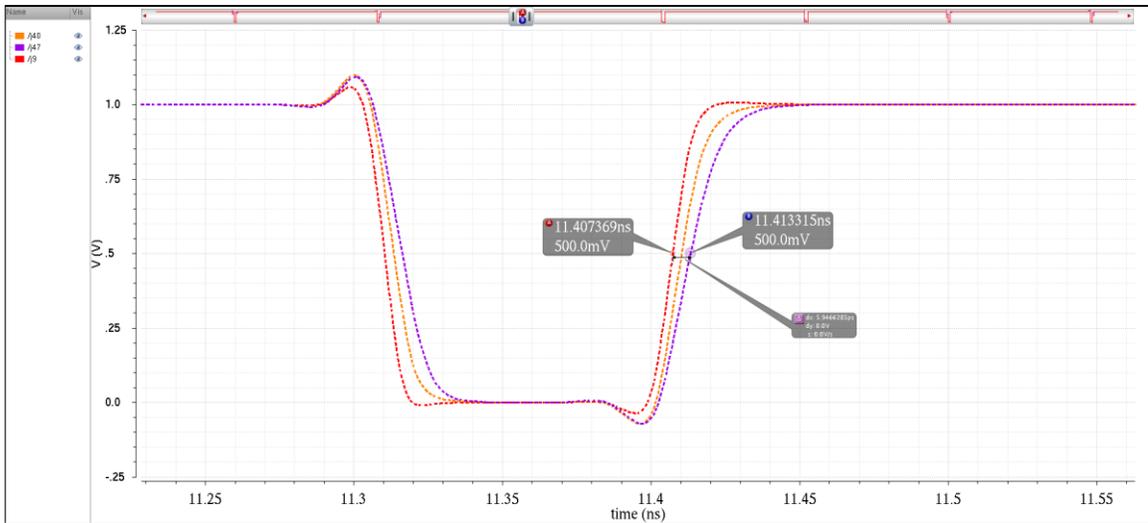
The test bench for the clock tree is shown in Fig.4.1. A clock signal with 50% duty cycle is generated by a signal source and then the clock signal input to a pulse generator that generates a narrow injection pulse. Finally, a one input 64 output clock tree function block generates 64 injection pulse signals with relatively small timing skew.

The simulation result for the clock tree outputs is shown in Fig. 4.2(a). The maximum timing skew of the clock tree outputs is approximately 6pS. The green curve represents the 47<sup>th</sup> output signal of the clock tree, and the red curve shows the 9<sup>th</sup> output signal of the clock tree. The time difference is measured from the 50% of the rising edges of the green curve and red curve, respectively. Ideally, the output signals of the clock tree should be overlapped without timing skew. However, at the layout level, all components are P&R automatically and is therefore limited by the P&R routing algorithms. For this design, no special timing constraints we're placed on the injection clocks, however, it should be possible to further reduce the timing skew with additional timing constraints at the expense of a slightly more complex design. Also, the timing would need to be modified

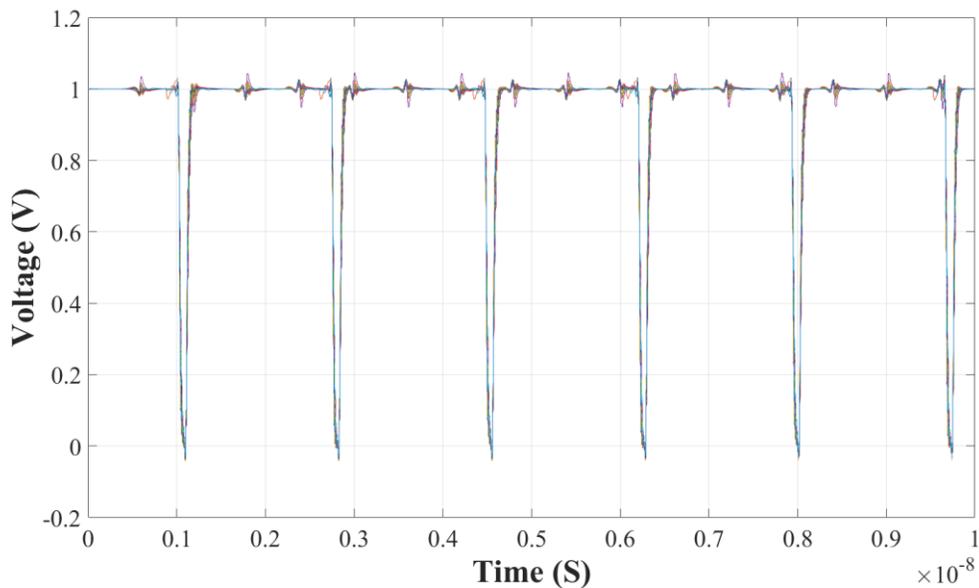
when porting the design to a new process technology. Additionally, Fig. 4.2(b) shows the all 64 fan-out signals. As can be seen, the duty cycle of the injection signal is greatly reduced compared to the 50% duty cycle input clock. The disturbances on the top of the signal are caused by the digital noise generated by the closely located DCO.



**Fig. 4.1 Test Bench of Clock Tree**



**Fig. 4.2(a). The Simulation Result of Clock Tree Outputs**



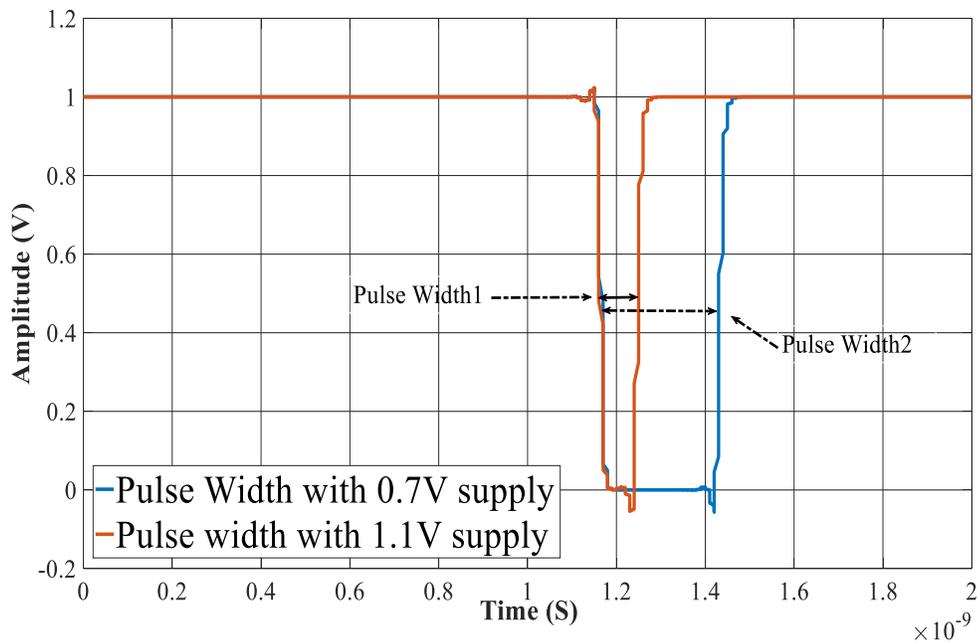
**Fig. 4.2(b). The Overall view of the Clock Tree Outputs**

## 4.2 Pulse Generator Simulation Results

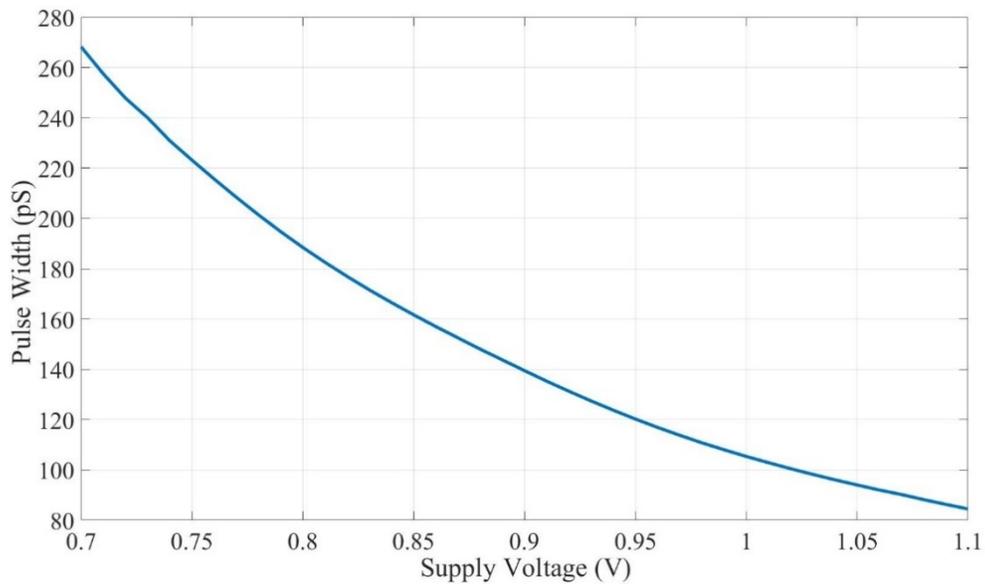
This section presents the simulation results for the pulse generator. As shown in Fig. 4.3, the red curve and the blue curve represent the output signal of the pulse generator with different supply voltages. At the 50% point of the rising and falling edge two

measuring points are placed and the pulse width is calculated from the time difference between these two points. With the maximum supply voltage (1.1V), the pulse width is approximately 85pS. With minimum supply voltage (0.7V) the pulse width is approximately 270pS.

By varying the supply voltage of the pulse generator, different pulse widths can be achieved and the relationship between pulse width and supply voltage is shown in Fig. 4.4. As mentioned previously [63], to achieve the optimal phase noise and locking bandwidth for different harmonics, the pulse width should cover the range from approximately 5% to 17% duty cycle depending on the order of output harmonic.



**Fig. 4.3 The Zoomed In Pulse Width with Different Supply Voltages**



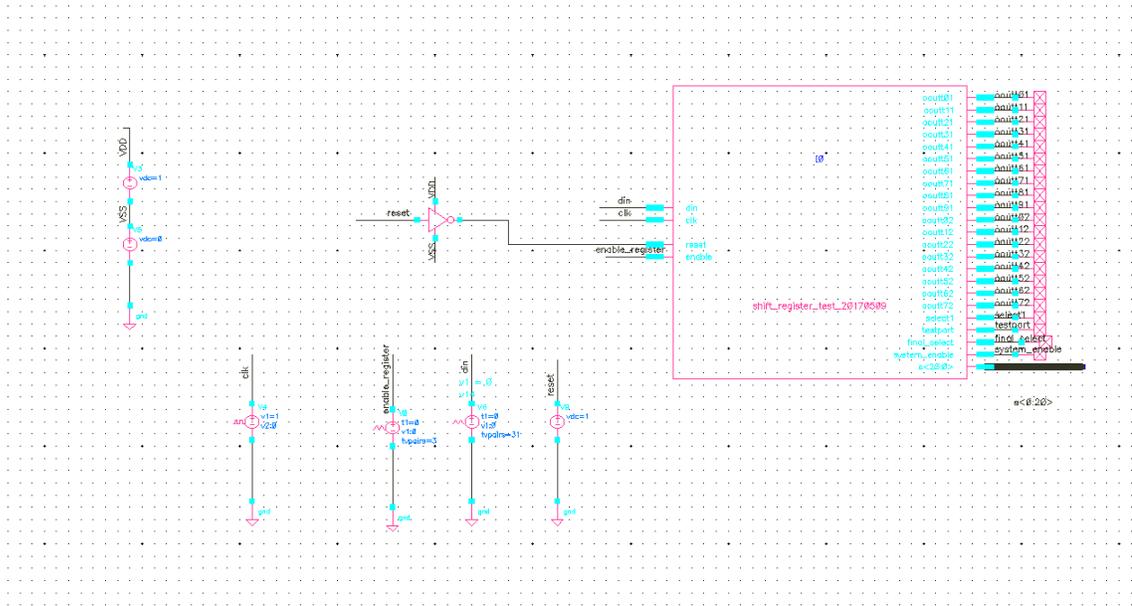
**Fig. 4.4 The Relationship Between Supply Voltages and Pulse Width**

### 4.3 S/P Shift Register Simulation Result

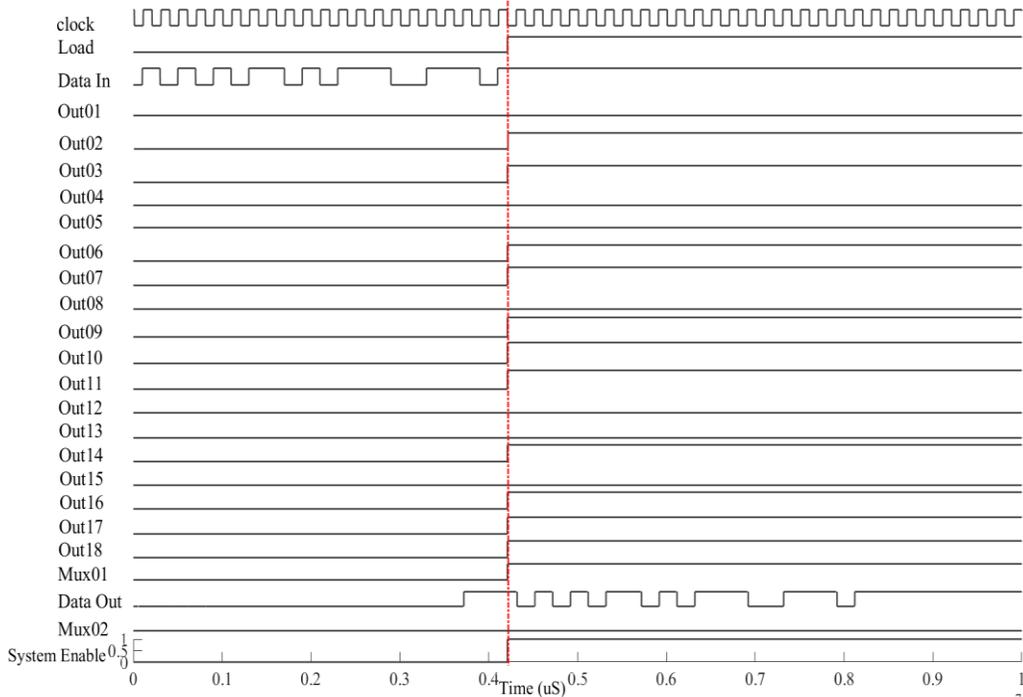
The test bench setting of the S/P shift register is shown in Fig. 4.7. The S/P shift register has four input ports: rest (reset), data\_in, clock, and enable. In addition, the reference clock is generated by a square wave generator and the rest control signal is generated by programmable signal sources in Cadence (i.e. ‘Voltage Piece Wise Linear (VPWL)’). The power supply of the shift register uses the same voltage as the oscillator core.

The simulation results of the S/P shift register are shown in Fig. 4.6. The clock is used for shifting the FCW into registers operating at 500MHz. In reality, the operating frequency of the clock on the physical test chip can be much lower. The ‘Data In’ contains a serial representation of the FCW. For the simulation results in Fig. 4.6 the data series is ‘10101011\_0101110011\_101’. The first 8 bits (starting from the left side of the series),

from Out01 to Out08, belong to the HF output oscillator FCW, and the following 10 bits, from Out09 to Out18, control the free running frequency of the Low-Frequency (LF) oscillator. The last 3 bits control the MUX1, MUX2 and system enable port respectively. The data stream will load into the system when the ‘Load’ signal goes high. The ‘Data Out’ is used to verify that the input data series is equal to the output data series. If the two data streams are identical, the system configuration control is operating appropriately.



**Fig. 4.5 Test Bench for S/P Shift Register**

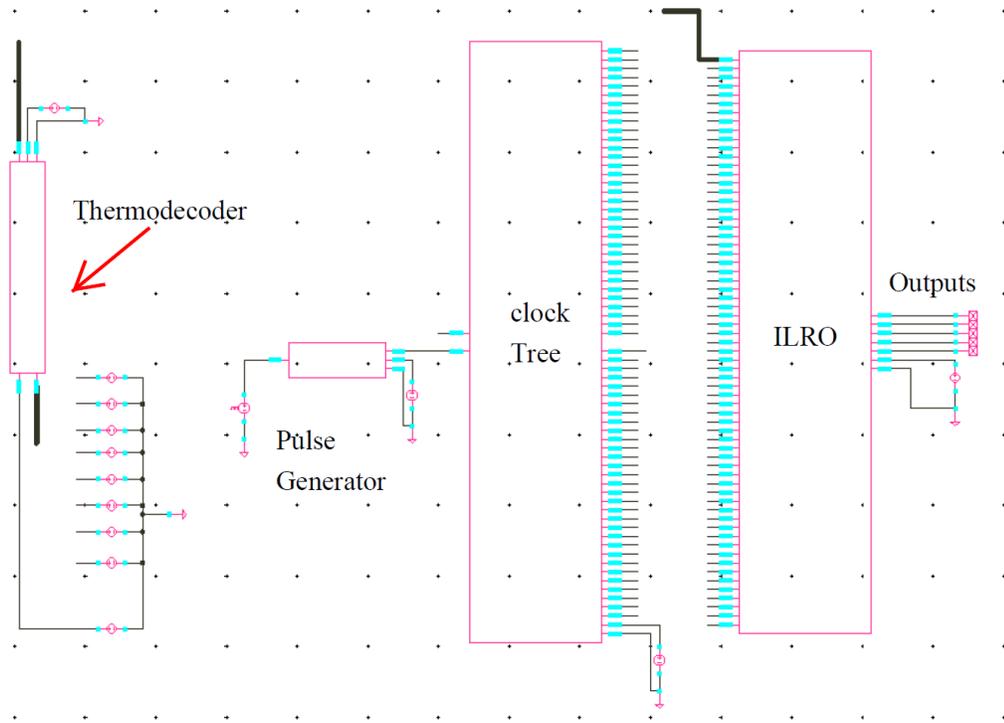


**Fig. 4.6 S/P Shift Register Simulation Result**

#### 4.4 ILRO Simulation Results

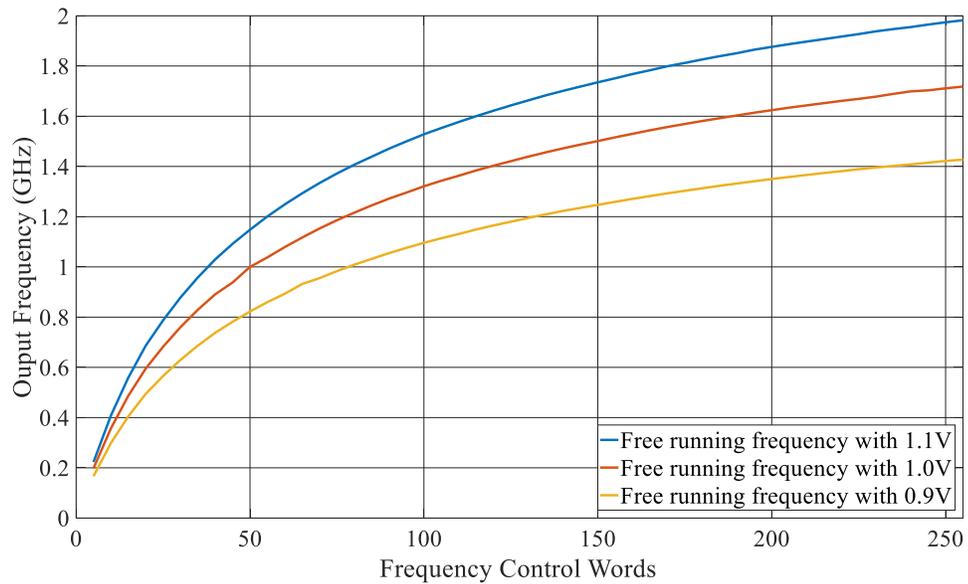
As shown in Fig. 4.7, this test bench is used for verifying the function of the ILRO. The thermometer decoder is used for coding a binary code into a thermometer code. The pulse generator and clock tree are combined together to provide the injection signals. The ILRO is the main core oscillator of the proposed design.

According to previous studies [47], varying the supply voltage of the oscillator core leads to a change in the output frequency due to variations in the charging and discharging time of the oscillator's parasitic load capacitances. As shown in Fig. 4.8, the output frequency of the ILRO increases with increased supply voltages as expected. At the maximum FCW (255) and supply voltage (1.1V), the operating frequency of ILRO is 1.95GHz. At the minimum supply voltage (0.9V) the operating frequency is 1.41GHz.



**Fig. 4.7 Test Bench for Frequency Tuning, Injection Locking, and Locking**

### Bandwidth

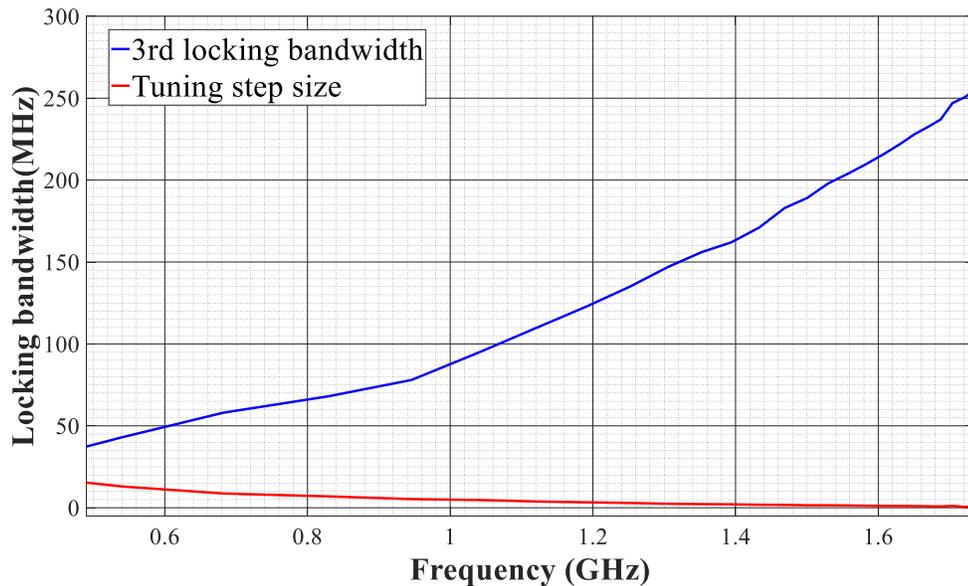


**Fig. 4.8 Frequency Tuning with Different Supply Voltages**

The locking bandwidth and tuning step sizes simulation results are shown in Fig. 4.9. As the frequency decreases the locking bandwidth decreases. In order to provide accurate tuning of the ILRO a locking bandwidth of at least three times of the tuning step size is required [63]. The tuning step size is 1.1 MHz at 1.78GHz and increases to 14MHz at 490MHz. It should be noted that higher order sub-harmonic injection can be supported with a reduced frequency range (i.e. increased minimum operating frequency). At the maximum operating frequency, the 3<sup>rd</sup> harmonic injection locking bandwidth is 255MHz and decreases to 43MHz at the 490MHz operating frequency. Based on the general equation of locking bandwidth [57]:

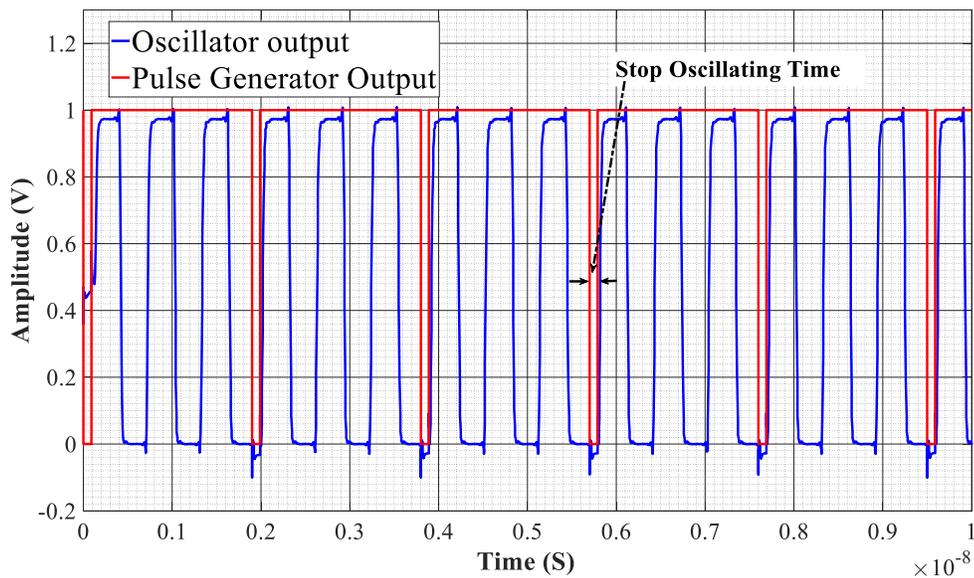
$$\omega_L = \frac{\omega_0}{2Q} \cdot \frac{I_{inj}}{I_{osc}} \quad (4.1)$$

When  $\omega_0$  (free running frequency), controlled by FCW, decrease the locking bandwidth decreases as well, which is in agreement with the simulation results.

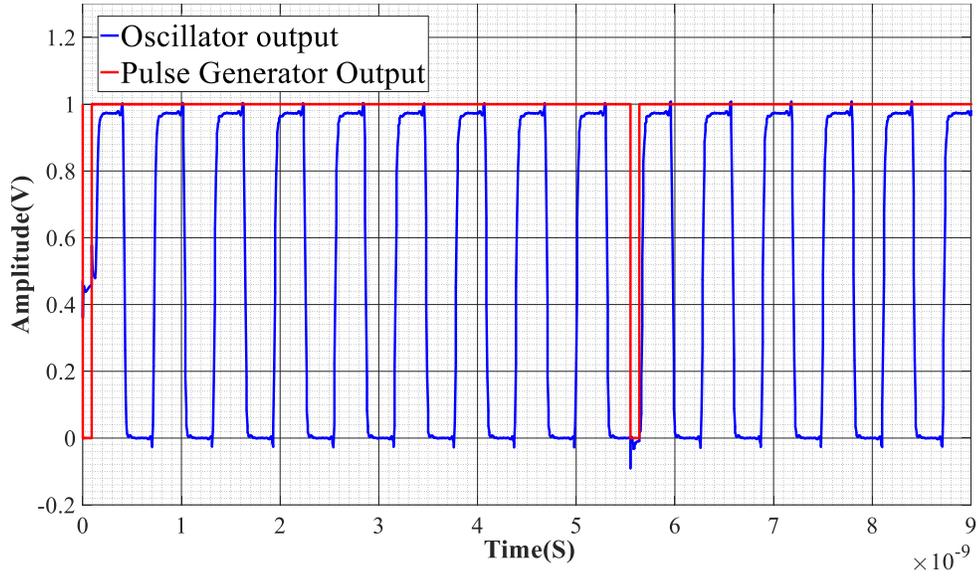


**Fig. 4.9 Locking Bandwidth and Step Size of the 3<sup>rd</sup> harmonic Injection Locking**

A simulation of 3<sup>rd</sup> harmonic injection locking in the time domain is shown in Fig. 4.10. In Fig. 4.10, the oscillator output frequency, blue curve, is three times higher than the injection clock, red curve. In the unlocked stage, the jitter of the free running oscillator will be continuously accumulated. However, in the locked stage, the injection clock, which is a low noise periodic signal, will be injected into the free running oscillator during every third clock cycle. As shown in Fig. 4.10, the free running oscillator is locked to the injection clock at the rising edge of every third clock cycle. The stop oscillating time is the period of time when the oscillator is disabled when injection clock is low. A simulation of 9<sup>th</sup> harmonic injection locking in the time domain is similar to 3<sup>rd</sup> harmonic injection locking with an output frequency nine times higher than the injection clock, as shown in Fig. 4.11.



**Fig. 4.10 3<sup>rd</sup> Sub-harmonic Injection locking Simulation in Time Domain**

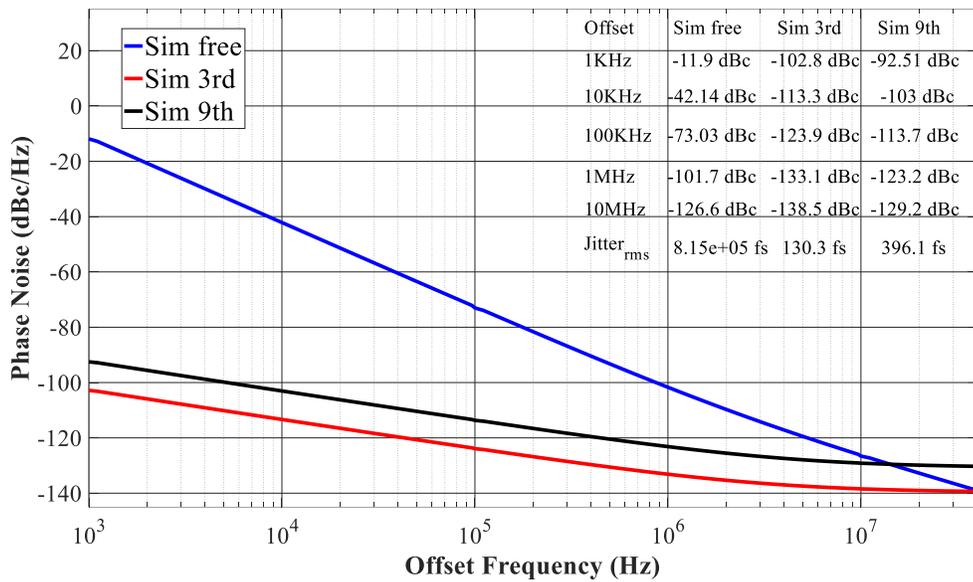


**Fig. 4.11 9<sup>th</sup> Sub-harmonic Injection locking Simulation in Time Domain**

The phase noise simulation results for the oscillator operating at 1.62GHz are shown in Fig. 4.12. The blue curve shows the simulated free running oscillator phase noise. The red curve and black curve represent the phase noise of the 3<sup>rd</sup>, and the 9<sup>th</sup> harmonics injection locking. At 1MHz frequency offset, the phase noise of the 3<sup>rd</sup> harmonic injection locking oscillator is -133.1dBc/Hz, and has an approximately 30dB improvement compared to the phase noise of free running oscillator at the same frequency offset. If comparing the phase noise values of the 3<sup>rd</sup> and the 9<sup>th</sup> harmonic at the same frequency offset, we can find the phase noise performance degradation follows the equation:

$$PN_{diff} = 20 * \log \frac{N_1}{N_2} \quad (4.2)$$

Based on the simulation result,  $N_1$  equals to 3 and  $N_2$  equals to 9. Therefore  $PN_{diff}$  equals approximately -9.5dB which is in good agreement with the simulation results. The small variation is believed to be due to the accuracy of the device models used in the simulations.



**Fig. 4.12 Simulated Phase Noise at 1.62 GHz**

#### 4.5 Conclusion

The simulation results of individual blocks are illustrated in this chapter. The simulation results include clock tree, pulse generator, S/P shift register simulation results. Additionally, some critical parameters of ILRO are elaborated in the chapter, such as frequency tuning range, locking bandwidth, and phase noise performance. Moreover, some important data is shown on plots. The simulation results are generally in good agreement with theoretical calculations with some small differences believed to be due to the accuracy of simulation device models and the simplifications used in the theoretical models.

## 5 Chapter: Implementation and Measurement Results

Measurement environment settings and measurement results are illustrated in this chapter. Section 5.1 focuses on the Printed Circuit Board (PCB) design and fabricated chip description. Measurement results of the ILRO are illustrated in Section 5.2.

### 5.1 Fabricated Microchip and Measurement Environment Setting

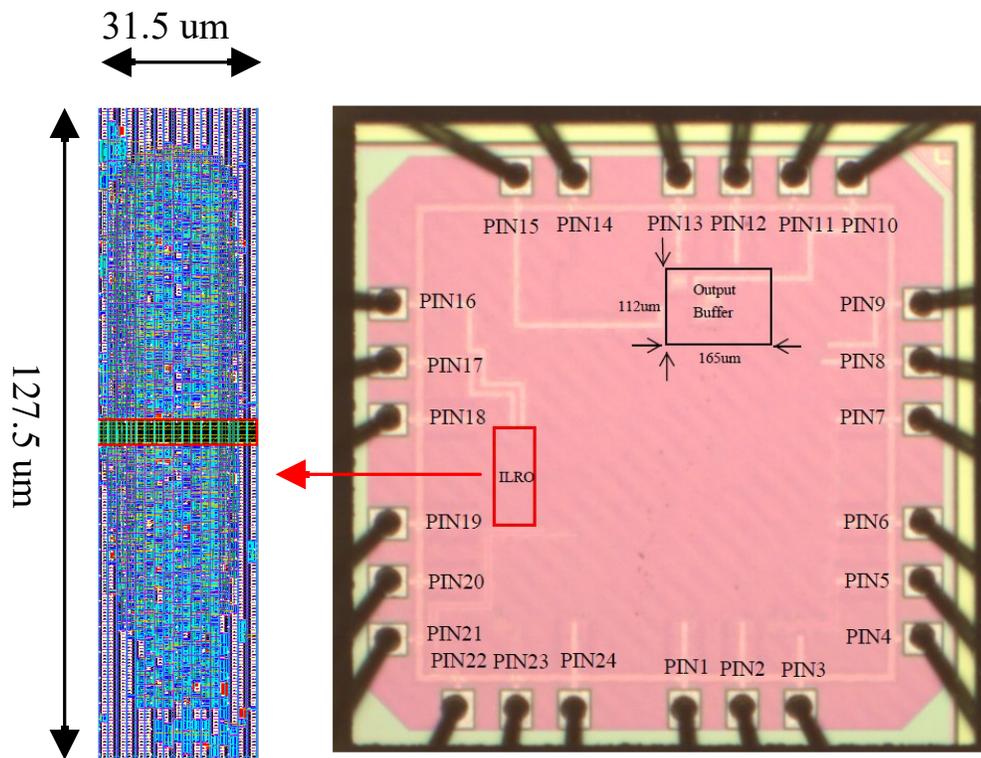
The proposed synthesized ILRO was fabricated using 65nm TSMC technology. The chip was packaged in a 24-pin Ceramic Flat Package (CFP) package and mounted on a custom Printed Circuit Board (PCB).

The microphotograph of the fabricated ILRO is shown in Fig. 5.1. The red area is the ILRO core of the proposed design, and the total area of the proposed design is 31.5um by 127.5um. The pin names and Input/Output (I/O) types are listed in Table 4. PIN6 is utilized for operation states selection from fractional state to integer state. PIN11 and PIN14 are the differential outputs of the proposed work. PIN17 and PIN21 are the power supplies for the ILRO core and pulse generator. PIN18 is the input port for the injection signal coming from Signal Generator (SG). PIN7, PIN19, PIN20, PIN23, and PIN24 belong to S/P shift register. A number of pins (PIN1, PIN2, PIN5, and PIN8) were used for a second design placed on the chip that was not part of this work.

**Table 4 Microchip Pins Arrangement**

Pin Number	Name	I/O Type
PIN1	CNT_START	Input
PIN2	CNT_STOP	Input
PIN3	VSS	Ground

PIN4	TACVDD	Power Supply
PIN5	SR_MODE	Input
PIN6	MR_MODE	Input
PIN7	RESET	Input
PIN8	VDD1	Power Supply
PIN9	VSS1	Ground
PIN10	VSS2	Ground
PIN11	OUTP	Output
PIN12	DC_BIAS	Power Supply
PIN13	EXTRA_BIAS	Power Supply
PIN14	OUT_N	Output
PIN15	VDD2	Power Supply
PIN16	VSS3	Ground
PIN17	VDD3	Power Supply
PIN18	LO_INJ_CLK	Input
PIN19	SR_CLK	Input
PIN20	SR_IN	Input
PIN21	VDD4	Power Supply
PIN22	VDD5	Power Supply
PIN23	MR_SR_OUT	Output
PIN24	SR_LOAD	Input

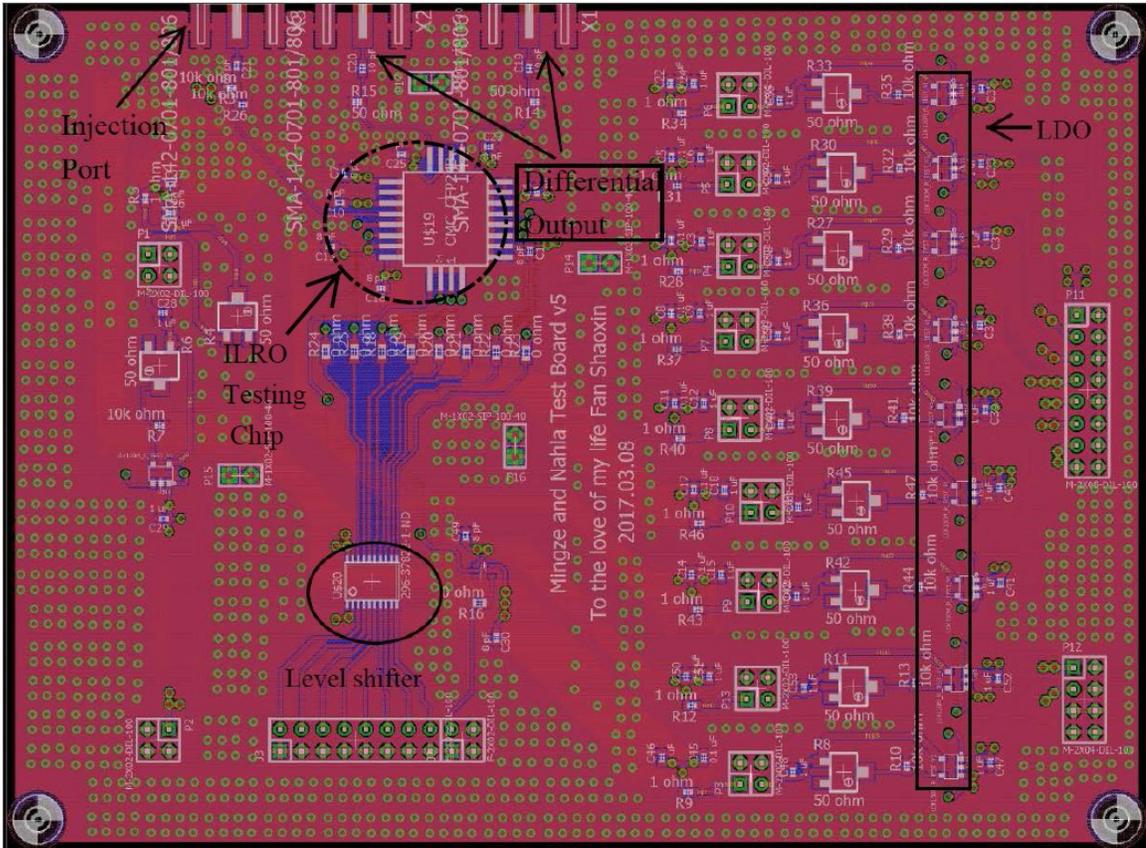


**Fig. 5.1 Microphotograph of the Fabricated ILRO**

A 4-layer custom PCB board was designed for testing the proposed design. As shown in Fig. 5.2 the test chip is placed in the top center of the PCB board. Two differential outputs with symmetrical routing were designed to provide low phase noise and drive a 50 ohm load. Additionally, a 50 ohm injection port was designed in the left top corner of the board. All three ports communicate with testing instruments through Sub-Miniature version A (SMA) connectors.

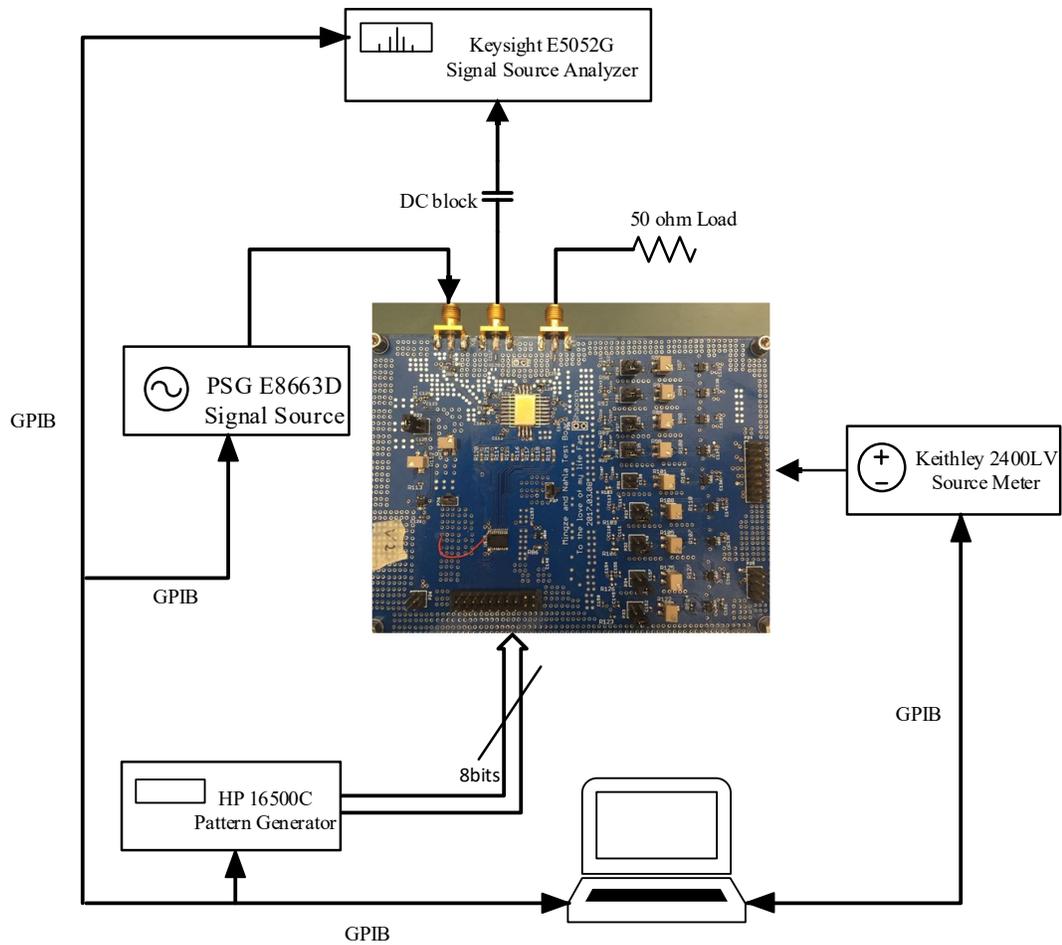
In order to reduce the total number of external power supplies, the test board uses a bank of low noise Low Drop Output (LDO) regulators as can be seen on the right side of the test board. Based on this approach, using a single external power supply can support all the necessary on-board supply voltages. The individual LDOs can also be bypassed, if necessary, allowing an external power supply to drive an on-board supply voltage.

Additionally, the FCW and states selection commands are imported into the test chip through an 8-bit level shifter, which is utilized for voltage level translation from a 3.3V test pattern generator supply voltage to the 1V digital I/O voltage of the test chip.



**Fig. 5.2 4-Layer Customized PCB Layout**

A schematic of the test bench of the proposed design is shown in Fig. 5.3. The figure also includes a detailed view of the custom PCB. The phase noise of the ILRO was evaluated using a Keysight E5052G signal source analyzer, and the reference signal for injection locking was generated using a PSG E8663D signal source. Communication between the computer and testing instruments was through a General Purpose Interface Bus (GPIB).



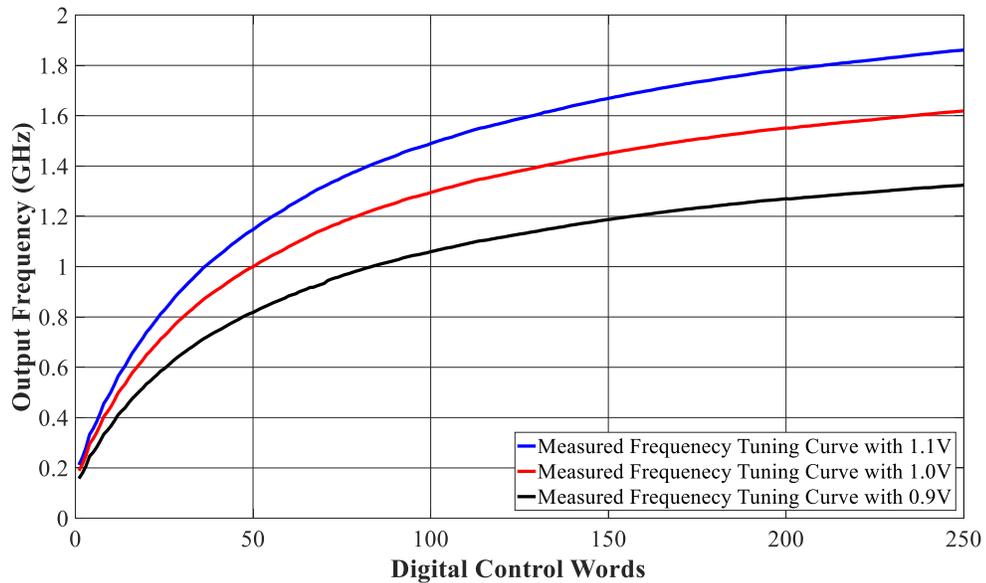
**Fig. 5.3 ILRO Test Bench**

## 5.2 ILRO Measurement Results

Critical measurement results of ILRO are discussed in this section including frequency tuning, locking bandwidth, injection locked output signal spectrum, and phase noise the free running and injection locking oscillator.

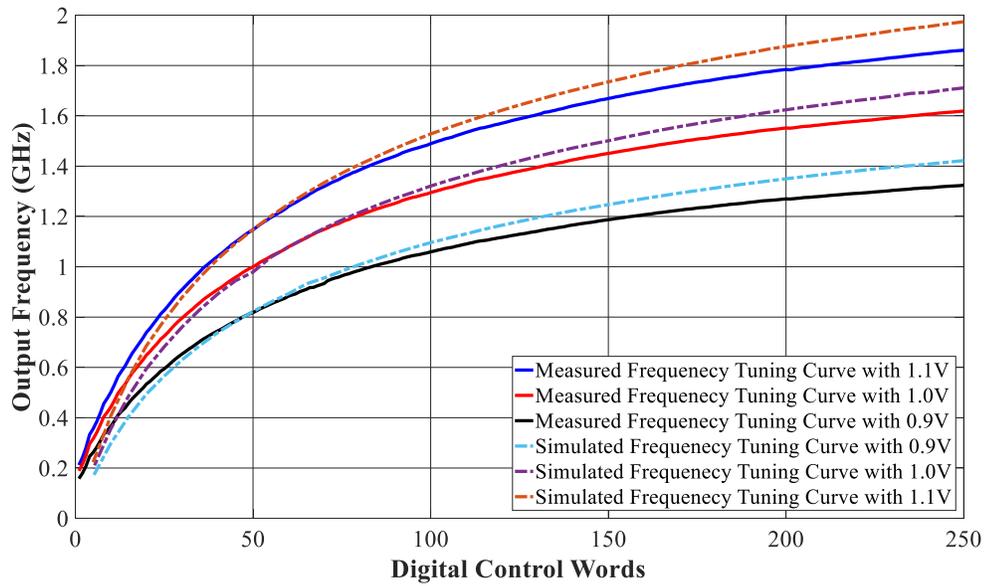
The frequency tuning curves for the ILRO are shown in Fig. 5.4. Three curves are shown with supply voltages of 0.9V, 1V, and 1.1V respectively. The tuning range of the ILRO is from 210MHz to a maximum of 1.81GHz with a 1.1V supply. It should be noted that the maximum output frequency is limited by the vendor supplied (TSMC) digital

library. The proposed work utilizes the academic version of the library which does not include a tristate inverter standard cell. The tristate inverter function was accomplished using a tristate buffer in series with an inverter. Therefore, the number of delay cells and load capacitance in the oscillator matrix is significantly increased thereby reducing the output frequency.



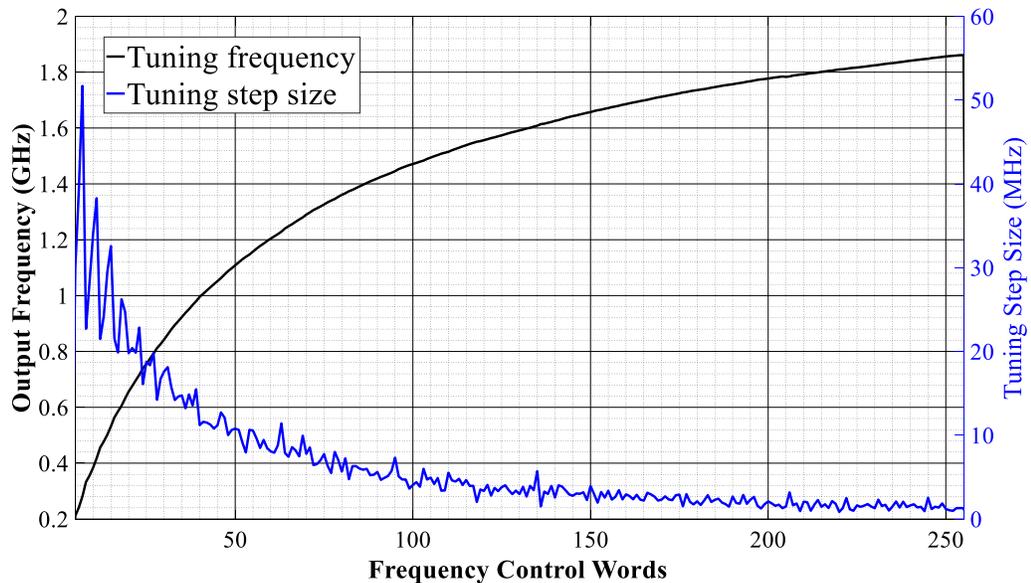
**Fig. 5.4 Frequency Tuning with the Different Supply Voltages**

Additionally, the output frequency comparison between extracted simulation results and measurement results are also plotted in Fig. 5.5. It is believed that extra parasitic routing components of the final P&R design, which could not be fully simulated, are a major cause in differences between the simulation and measurement results.



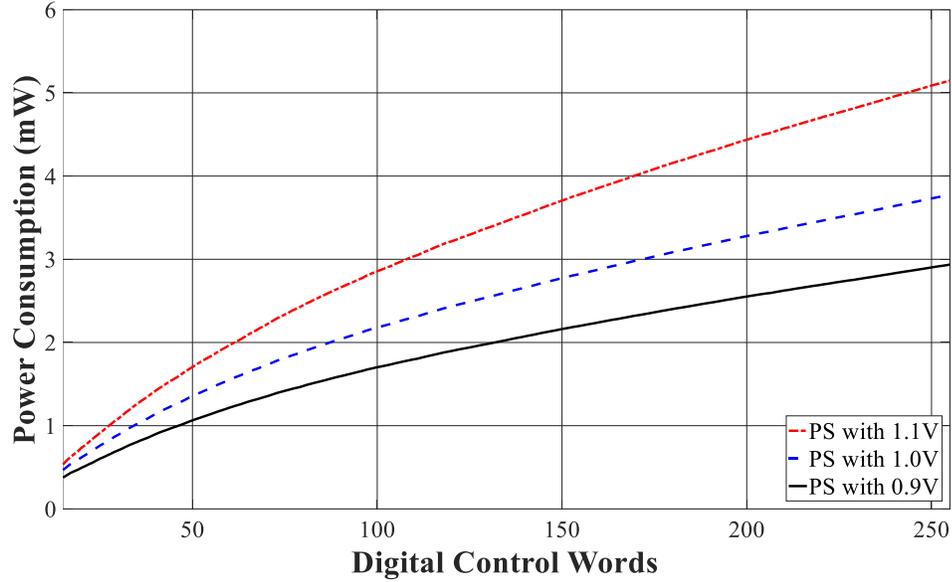
**Fig. 5.5 Frequency Tuning Simulation and Measurement Results**

The relationship between output frequency and tuning step size with a 1.1V supply is illustrated in Fig. 5.6. The step size is relatively small (less than 10MHz) when the FCW is above 55. The relatively large step size can be tolerated because the oscillator design of the proposed work has large locking bandwidth.



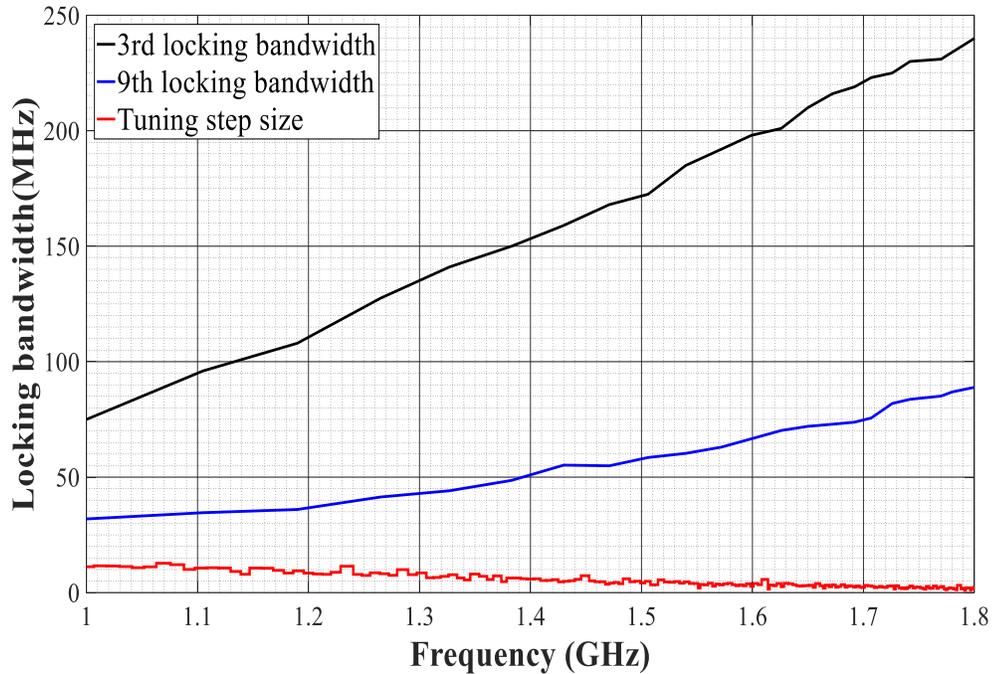
**Fig. 5.6 Frequency Tuning Curve and Tuning Step Size**

The free running power consumption curves of the ILRO with different supply voltages are shown in Fig. 5.7. With decreasing supply voltage or FCW, the power consumption decreases as well, which is in agreement with previous studies [49] [63].



**Fig. 5.7 The Power Consumption of Free Running Oscillator with Different Supply Voltages**

The locking bandwidth and tuning step sizes of the proposed work are shown in Fig. 5.8. As the frequency decreases or injection sub-harmonic order increases the locking bandwidth decreases. In order to provide accurate tuning of the ILRO a locking bandwidth of at least three times the tuning step size is required [63]. The tuning step size is 1.3MHz at 1.8GHz and increases to 11.2MHz at 1GHz. As can be seen in Fig. 5.8, the minimum operating frequency of the 9<sup>th</sup> harmonic is 1GHz. The minimum operating frequency for the 3<sup>rd</sup> harmonic locking is 770MHz (not shown in figure). It should be noted that higher order harmonic injection can be supported with a reduced frequency range (i.e. increased minimum operating frequency).

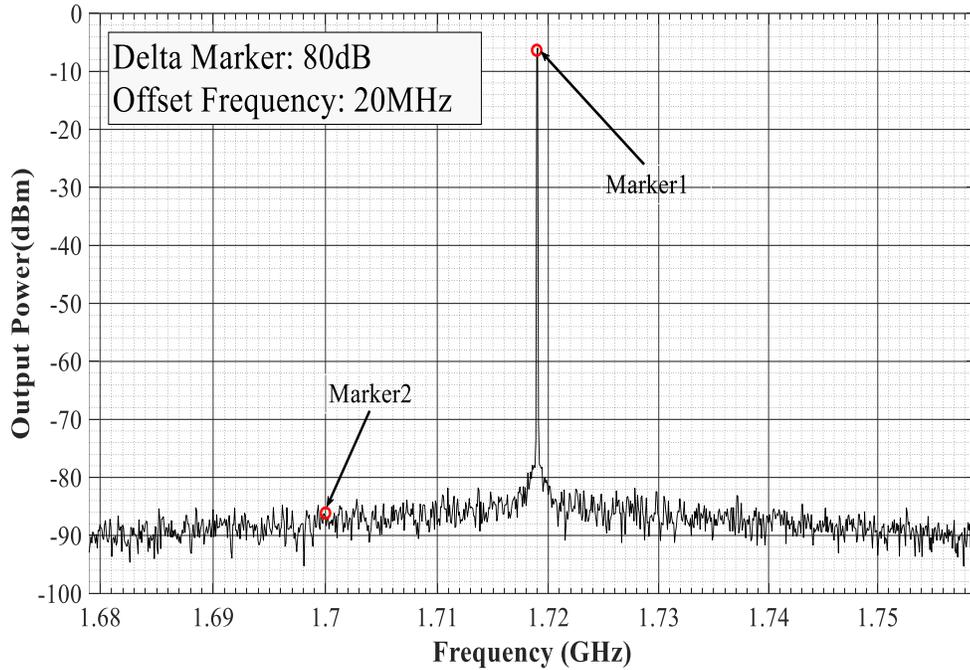


**Fig. 5.8 Locking Bandwidth of Different Injection Harmonics and Tuning Step Size as a Function of Frequency**

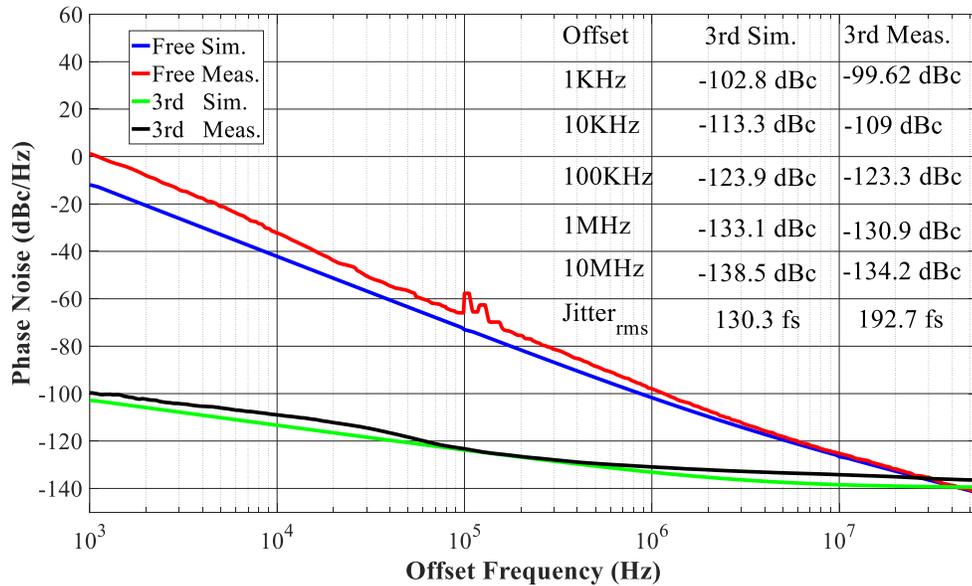
The output spectrum with 80MHz bandwidth of the 3<sup>rd</sup> sub-harmonic locked ILRO is shown in Fig. 5.9. The output spectrum shows that any close in spurs are -80dB or lower. The far out reference spur (not shown) is -35dB.

Based on the measurement results, the phase noise performance of the locked signal with different operating frequencies gives the similar performance, therefore one of the random operating frequency is chosen by the author. The phase noise simulation and measurement results for the oscillator operating at 1.62GHz are shown in Fig. 5.10. The blue curve and the green curve show the simulated phase noise of the free running oscillator and the 3<sup>rd</sup> harmonic injection locked oscillator respectively. The red and black curves show the measured phase noise of the free running oscillator and the 3<sup>rd</sup> harmonic injection locked oscillator respectively. As can be seen, the simulation and measurement results are

closely matched. The reduced noise in the simulation results is believed to be due to the accuracy of the device models used in the simulations. Additionally, the proposed work consumes 7.15mW with 1.1V supply at 1.62GHz carrier frequency.

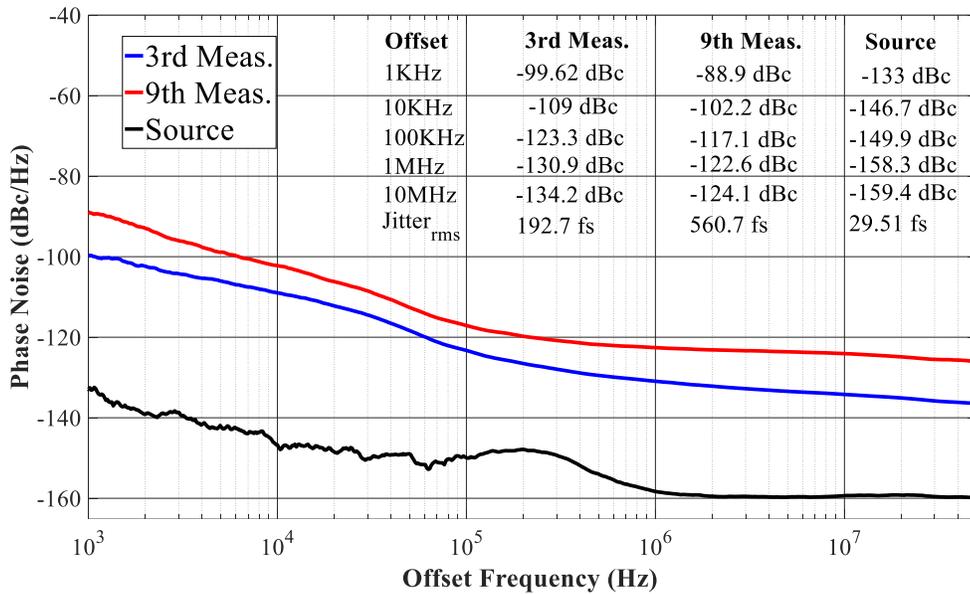


**Fig. 5.9 Injection Locked ILRO Output Spectrum with 80MHz Bandwidth**



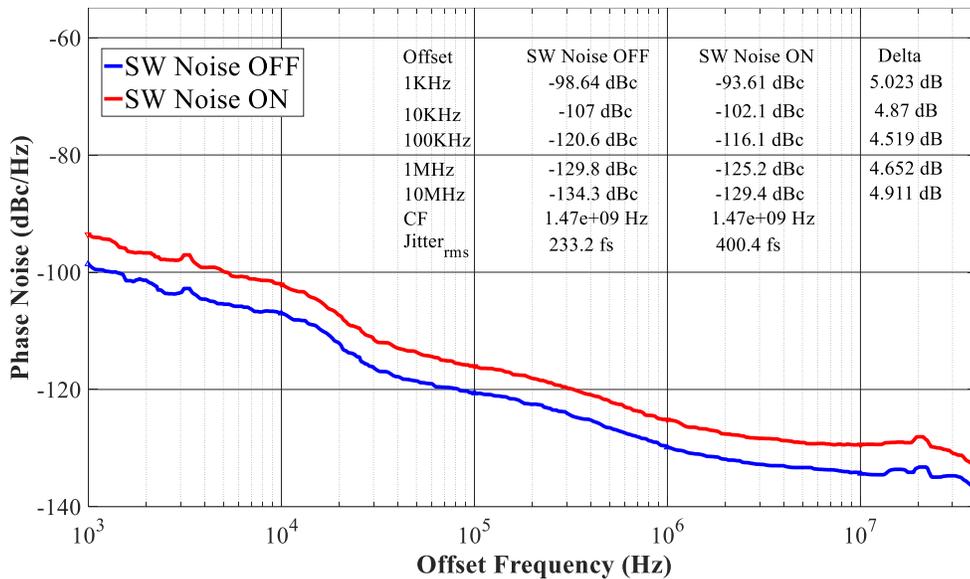
**Fig. 5.10 Phase Noise Simulation and Measurement Result at 1.62GHz**

The measured phase noise of the ILRO for the 3<sup>rd</sup> and 9<sup>th</sup> sub-harmonic is shown in Fig.5.11. The phase noise was evaluated using a Keysight E5052G signal source analyzer, and the reference signal for injection locking was generated using a PSG E8663D signal source. The ILRO phase noise for the 3<sup>rd</sup> and the 9<sup>th</sup> sub-harmonics is -130.9 dBc/Hz and -122.6dBc/Hz respectively, at 1 MHz offset frequency from a carrier frequency of 1.62GHz. The 3<sup>rd</sup> sub- harmonic phase noise maps to 192.7fs RMS jitter when integrated from 1 KHz to 40MHz.



**Fig. 5.11 Measured Phase Noise at 1.62GHz**

Sensitivity to on chip digital SWitching (SW) noise was measured by clocking six thousand closely located flip-flops with a random data pattern. The results are shown in Fig. 5.12. When the flip-flops are toggled, the SW noise is coupled into IRLO substrate and power supplies. As can be seen in the figure , when the SW turns on, the phase noise performance is degraded by approximately 5dB. It is believed that the SW noise performance could be improved with a differential oscillator design at the cost of extra area and power.



**Fig. 5.12 Comparison of Phase Noise with Digital Switching Noise On and OFF**

The performance of prior art is listed in Table 5 for comparison. The phase noise, chip area, and FOM are improved in the proposed work.

**Table 5 Performance and Comparison**

	[49]	[67]	[68]	[69]	This work
Freq.[GHz]	0.39-1.41	2.4	2.2-2.5	2.39-2.55	1-1.8
Power [mW]	0.78	12.6	128	9	7.15
RMS jitter [Ps] 1kHz – 40MHz	2.8	0.145	NA	4.6	0.1927
PN,1MHz offset frequency	-118	-129	-103	-119	-131@ 1.62GHz
Locking bandwidth(MHZ)	NA	NA	NA	NA	240
Area(mm <sup>2</sup> )	0.0066	0.64	NA	NA	0.004
Topology	Injection locking	Injection locking	PLL	ADPLL	Injection locking
Technology	CMOS 65nm	CMOS 180nm	CMOS 180nm	CMOS 130nm	CMOS 65nm
FOM <sup>a</sup> (dB)	-232	-246	NA	-217	-246
FOM <sup>b</sup> (dB)@1MHz	-182	-186	-150	-178	-187

$$FOM^a = 20 \log \left( \frac{\tau_t}{1s} \right) + 10 \log \left( \frac{P}{1mW} \right) [70]$$

$$FOM^b = L\{f_{offset}\} - 20 \log \left( \frac{f_o}{f_{offset}} \right) + 10 \log \left( \frac{P}{1mW} \right) [63]$$

Where  $\tau_t$  is the RMS jitter,  $P$  is the power consumption,  $L\{f_{offset}\}$  is the phase noise at 1MHz offset, and  $f_o$  is the carrier frequency

### 5.3 Conclusion

The measurement environment settings are illustrated in this chapter. Comparison between measurement results and simulation results are discussed in this chapter as well.

The ILRO measurements focused on the frequency tuning range, output spectrum, locking bandwidth, and phase noise performance. Moreover, the proposed ILRO achieves excellent FOM compared with previous work.

## **6 Chapter: Thesis Conclusion**

A fully synthesized ILRO is presented in this work. The proposed design has advantages of large frequency tuning range, small die area, large locking bandwidth, and excellent phase noise performance. All of the building blocks are synthesized from a vendor supplied standard cell library (TSMC). The design is compatible with deep submicron technologies and can be ported to more advanced technologies where the frequency tuning range, resolution, area, and power consumption can be improved with CMOS process scaling.

### **6.1 Accomplishments**

The proposed design attempts to combine a novel injection locking technique with a fully synthesized ring oscillator. The chip area is dramatically reduced compared to the previous work [31] – [35] [63] and based on the author’s knowledge this design achieves the largest locking bandwidth for both 3<sup>rd</sup> and 9<sup>th</sup> harmonics injection locking in the 2GHz frequency range.

### **6.2 Issues in the Design**

The extra parasitic routing components are hard to predict during the synthesizing procedures, and are a major cause in differences between simulation and measurement results. The main parasitic components generated by Cadence are capacitors, which are carried out by Capacitor Extractions. Additionally, the increasing frequency step size at lower output frequencies is a limiting factor in terms of usable output frequency range particularly for higher order harmonics.

### **6.3 Future Work**

The low frequency step size can be improved with a larger ring oscillator array (at the cost of more power) and more advanced tuning algorithm [21]. The differential ring structure can also be introduced in future designs to enhance the noise immunity. A fractional injection locked method could also be explored for higher injection locked frequency resolution [71] [72]. Additionally, a high-speed counter and replica DCO [49] can be added for automated frequency tuning. Finally, a fully synthesized ADPLL using FPGAs could be explored [73].

## References

- [1] W. Krenik, D. D. Buss and P. Rickert, "Cellular handset integration -SIP versus SOC," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1839-1946, Aug. 2005.
- [2] B. Razavi, "Recent developments in RF receivers," in *2014 IEEE Proceedings of the Custom Integrated Circuits Conference (CICC)*, San Jose, CA, USA, 2014.
- [3] R.B.Staszewski, J.L.Wallberg and S.Rezeq, "All-digital PLL and transmitter for mobile phones," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2469-2482, Dec. 2005.
- [4] A. A. Abidi, "RF CMOS comes of age," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 4, pp. 549-561, Nov. 2004.
- [5] R. B. Staszewski, C.-M. Hung and K.MAggio, "All-digital phase-domain TX frequency synthesizer for Bluetooth radios in 0.13/ $\mu\text{m}$  CMOS," in *2004 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2004.
- [6] R. Staszewski, D. Leipold and C.-M. Hung, "A first digitally-controlled oscillator in a deep-submicron CMOS process for multi-GHz wireless applications," in *2003 IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, Philadelphia, PA, USA, 2003.

- [7] S. Lee, "A low cost 24 GHz RF transceiver with four-RX and one-TX for radar sensor," in *2015 International SoC Design Conference (ISOCC)*, Gyungju, South Korea, 2015.
- [8] G. Jung and S. Park, "Performance improvements of Universal Mobile Telecommunications System enhanced uplink using mitigation scheme in single Tx and dual Rx dual-SIM dual-active devices," *Electronics Letters*, vol. 51, no. 25, pp. 2160-2162, Oct. 2015.
- [9] W. Choi, T. Kim and J. Shim, "23.8 A 1V 7.8mW 15.6Gb/s C-PHY transceiver using tri-level signaling for post-LPDDR4," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2017.
- [10] B. Can, B. S. Bisla and S. Patnaik, "Novel Fractional Spur Relocation in All Digital Phase Locked Loops," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, CA, USA, 2017.
- [11] Y. P. Chen, L. W. Massengill and B. L. Bhuvva, "Single-Event Characterization of Bang-bang All-digital Phase-locked Loops (ADPLLs)," *IEEE Transactions on Nuclear Science*, vol. 62, no. 6, pp. 2650 - 2656, Dec. 2015.
- [12] J.-M. N. Akre, J. Juillard, D. Galayko and E. Colinet, "Synchronization Analysis of Networks of Self-Sampled All-Digital Phase-Locked Loops," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 4, pp. 708 - 720, Apr. 2012.

- [13] T.-Y. Hsu, B.-J. Shieh and C.-Y. Lee, "An all-digital phase-locked loop (ADPLL)-based clock recovery circuit," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 8, pp. 1063 - 1073, Aug. 1999.
- [14] Y. Park and D. D. Wentzloff, "An all-digital PLL synthesized from a digital standard cell library in 65nm CMOS," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, CA, USA, 2011.
- [15] M. Elkholy and K. Entesari, "A Wideband Low-Power LC-DCO-Based Complex Dielectric Spectroscopy System in 0.18- $\mu\text{m}$  CMOS," *IEEE Transactions on Microwave Theory and Techniques*, vol. PP, no. 99, pp. 1-14, Apr. 2017.
- [16] D. Yang, W. Deng and B. Liu, "An HDL-synthesized injection-locked PLL using LC-based DCO for on-chip clock generation," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Chiba, Japan, 2017.
- [17] J. Rogers and C. Plett, in *Radio Frequency Integrated Circuit Design*, Ottawa, ON, Canada, Artech House, Apr. 2003, p. 245.
- [18] D. Yang, W. Deng and B. Liu, "An LC-DCO based synthesizable injection-locked PLL with an FoM of  $-250.3\text{dB}$ ," in *European Solid-State Circuits Conference, ESSCIRC Conference 2016: 42nd*, Lausanne, Switzerland, Sept. 2016.
- [19] S. Kamran and N. Ghaderi, "A novel high speed CMOS pseudo-differential ring VCO with wide tuning control voltage range," in *2017 Iranian Conference Electrical Engineering (ICEE)*, Tehran, Iran, Iran, 2017.

- [20] H. E. Taheri and M. Ehsanian, "A high-performance LC-VCO based adaptive bandwidth, adaptive jitter phase locked loop," in *Electrical Engineering (ICEE), 2017 Iranian Conference*, Tehran, Iran, Iran, July 2017.
- [21] W. Grollitsch and R. Nonis, "A fractional-N, all-digital injection-locked PLL with wide tuning range digitally controlled ring oscillator and Bang-Bang phase detection for temperature tracking in 40nm CMOS," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, Lausanne, Switzerland, 2016.
- [22] S. Liu, Y. Zheng, W. M. Lim and W. Yang, "Ring Oscillator Based Injection Locked Frequency Divider Using Dual Injection Paths," *IEEE Microwave and Wireless Components Letters*, vol. 25, no. 5, pp. 322 - 324, Mar. 2015.
- [23] K. Yousef, H. Jia and A. Allam, "An eight-phase CMOS injection locked ring oscillator with low phase noise," in *2014 IEEE International Conference Ultra-WideBand (ICUWB)*, Paris, France, 2014.
- [24 ] S.-y. Lee, S. Amakawa, N. Ishihara and K. Masu, "2.4–10 GHz Low-Noise Injection-Locked Ring Voltage Controlled Oscillator in 90 nm Complementary Metal Oxide Semiconductor," *Japanese Journal of Applied Physics*, vol. 50, no. 4S, Apr. 2011.
- [25] R. B. Staszewski and P. T. Balsara, in *ALL-DIGITAL FREQUENCY SYNTHESIZER IN DEEP-SUBMICRON CMOS*, John Wiley & Sons, Inc, 2006, pp. 2-5.

- [26] A. Hajimiri, S. Limotyrakis and T. H. Lee, "Jitter and Phase Noise in Ring Oscillators," *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, vol. 34, no. 6, pp. 790 - 804, June 1999.
- [27] SiTime, "Clock Jitter Definitions and Measurement Methods," Jan. 2014. [Online]. Available: <https://www.sitime.com/support2/documents/AN10007-Jitter-and-measurement.pdf>.
- [28] N. Roberts, "Phase noise and jitter -- a primer for digital designers," 7 July 2003. [Online]. Available: [http://www.eetimes.com/document.asp?doc\\_id=1277196](http://www.eetimes.com/document.asp?doc_id=1277196).
- [29] J. S. Hamel and R. Norris, "LC Tank Voltage Controlled Oscillator," 2005. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.448.6283&rep=rep1&type=pdf>.
- [30] M. J. Hemmati, "Ultra-low-phase-noise CMOS LC quadrature voltage controlled oscillator with Colpitts topology," *Electronics Letters*, vol. 50, no. 3, pp. 166-168, Feb. 2014.
- [31] G. Li and E. Afshari, "A Low-Phase-Noise Multi-Phase Oscillator Based on Left-Handed LC-Ring," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 9, pp. 1822 - 1833, Aug. 2010.
- [32] E.-S. A. Kytonaki and Y. Papananos, "A Low-Voltage Differentially Tuned Current-Adjusted 5.5-GHz Quadrature VCO in 65-nm CMOS Technology," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 5, pp. 254 - 258, June 2011.

- [33] Z. Zahir and G. Banerjee, "A multi-tap inductor based 2.0–4.1 GHz wideband LC-oscillator," in *2016 IEEE Asia Pacific Conference Circuits and Systems (APCCAS)*, Jeju, South Korea, 2016.
- [34] F. Pepe, A. Bonfanti and S. Levantino, "A wideband voltage-biased LC oscillator with reduced flicker noise up-conversion," in *2013 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, Seattle, WA, USA, 2013.
- [35] A. Kavala, D.-S. Kim, S. Jang and D.-K. Jeong, "A 5.6 GHz LC digitally controlled oscillator with high frequency resolution using novel quadruple resolution varactor," in *2010 International SoC Design Conference*, Seoul, South Korea, 2010.
- [36] J. Rogers and C. Plett, in *Radio Frequency Integrated Circuit Design*, Ottawa, Ontario, Canada, Artech House, Apr. 2003, pp. 245-302.
- [37] F. Chicco, R. Capoccia, A. Pezzotta and C. Enz, "Linear analysis of phase noise in LC oscillators in deep submicron CMOS technologies," in *2017 International Conference on Noise and Fluctuations (ICNF)*, Vilnius, Lithuania, Lithuania, 2017.
- [38] D. Murphy, J. J. Rael and A. A. Abidi, "Phase Noise in LC Oscillators: A Phasor-Based Analysis of a General Result and of Loaded Q," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1187 - 1203, Dec. 2009.
- [39] R. K. Pokharel, P. Nugroho, A. Anand, K. Kanaya and K. Yoshida, "Digitally controlled CMOS quadrature ring oscillator with improved FoM for GHz range all-digital phase-locked loop applications," in *2012 IEEE/MTT-S International Microwave Symposium Digest*, Montreal, QC, Canada, 2012.

- [40] O. Nizhnik, R. K. Pokharel, H. Kanaya and K. Yoshida, "Low Noise Wide Tuning Range Quadrature Ring Oscillator for Multi-Standard Transceiver," *IEEE Microwave and Wireless Components Letters*, vol. 19, no. 7, pp. 470 - 472, June 2009.
- [41] C. Li and J. Lin, "A 1–9 GHz Linear-Wide-Tuning-Range Quadrature Ring Oscillator in 130 nm CMOS for Non-Contact Vital Sign Radar Application," *IEEE Microwave and Wireless Components Letters*, vol. 20, no. 1, pp. 34 - 36, Nov. 2009.
- [42] P. Nugroho, R. K. Pokharel and A. Anand, "Development of low phase noise quadrature output digitally controlled CMOS ring oscillator," in *Asia-Pacific Microwave Conference (APMC) 2011*, Melbourne, VIC, USA, 2011.
- [43] S. S. Nagam and P. R. Kinget, "A -236.3dB FoM sub-sampling low-jitter supply-robust ring-oscillator PLL for clocking applications with feed-forward noise-cancellation," in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, Austin, TX, USA, USA, 2017.
- [44] D. Liao, R. Wang and F. F. Dai, "A low-noise inductor-less fractional-N sub-sampling PLL with multi-ring oscillator," in *Radio Frequency Integrated Circuits Symposium (RFIC), 2017 IEEE*, Honolulu, HI, USA, USA, July 2017.
- [45] J. Lee, "Oscillators," Electrical Engineering Department National Taiwan University, Taiwan, China.
- [46] S. Docking and M. Sachdev, "A method to derive an equation for the oscillation frequency of a ring oscillator," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 2, pp. 259 - 264, Feb. 2003.

- [47] J. A. McNeill and D. S. Ricketts, in *The Designer's Guide to Jitter in Ring Oscillators*, Springer, 2009, pp. 13 - 34.
- [48] A. Hajimiri and T. H. Lee, "A General Theory of Phase Noise in Electrical Oscillators," *IEEE Journal OF Solid-State Circuits*, vol. 33, no. 2, pp. 189 - 204, Feb. 1998.
- [49] W. Deng, D. Yang, T. Ueno, K. Okada and A. Matsuzawa, "A Fully Synthesizable All-Digital PLL With Interpolative Phase Coupled Oscillator, Current-Output DAC, and Fine-Resolution Digital Varactor Using Gated Edge Injection Technique," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 68 - 80, Sept. 2014.
- [50] J. Rogers, C. Plett and F. Dai, in *Integrated Circuit Design for High-Speed Frequency Synthesis*, Ottawa, Ontario, Canada, ARTECH HOUSE, INC, 2006, pp. 43-299.
- [51] A.-J. Annema, B. Nauta, R. v. Langevelde and H. Tuinhout, "Analog circuits in ultra-deep-submicron CMOS," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 132 - 143, Jan. 2005.
- [52] J. A. Tierno, A. V. Rylyakov and D. J. Friedman, "A Wide Power Supply Range, Wide Tuning Range, All Static CMOS All Digital PLL in 65 nm SOI," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 42 - 51, Jan. 2008.
- [53] V. Prasad and C. Sharma, "A Review of Phase Locked Loop," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 6, pp. 98 - 104, June 2012.

- [54] C. Priyanka and P. Latha, "Design and implementation of time to digital converters," in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2015.
- [55] R. B. Staszewski, D. Leipold and K. Muhammad, "Digitally controlled oscillator (DCO)-based architecture for RF frequency synthesis in a deep-submicrometer CMOS Proces," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 11, pp. 815 - 828, Nov. 2003.
- [56] R. B. Staszewski and P. T. Balsara, in *ALL-DIGITALFREQUENCY SYNTHESIZER IN DEEP-SUBMICRON CMOS*, JOHN WILEY & SONS, INC., 2006, pp. 130 -131.
- [57] B. Razavi, "A Study of Injection Locking and Pulling in Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1415 - 1424, Aug. 2004.
- [58] R. Adler, "A Study of Locking Phenomena in Oscillators," *Proceedings of the IRE*, vol. 34, no. 6, pp. 351 - 357, June 1946.
- [59] L. Paciorek, "Injection locking of oscillators," *Proceedings of the IEEE*, vol. 53, no. 11, pp. 1723 - 1727, Nov. 1965.
- [60] B. Mesgarzadeh and A. Alvandpour, "First-Harmonic Injection-Locked Ring Oscillators," in *IEEE Custom Integrated Circuits Conference 2006*, San Jose, CA, USA, 2006.
- [61] J.-C. Chien and L.-H. Lu, "Analysis and Design of Wideband Injection-Locked Ring Oscillators With Multiple-Input Injection," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 9, pp. 1906 - 1915, Aug. 2007.

- [62] K. Hu, T. Jiang and J. Wang, "A 0.6 mW/Gb/s, 6.4–7.2 Gb/s Serial Link Receiver Using Local Injection-Locked Ring Oscillators in 90 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 899 - 908, Mar. 2010.
- [63] Z. Bai, X. Zhou, R. D. Mason and G. Allan, "A 2-GHz Pulse Injection-Locked Rotary Traveling-Wave Oscillator," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 6, pp. 1854 - 1866, Apr. 2016.
- [64] Y. Soliman and R. D. Mason, "Application of Subharmonic Injection Locking of LC Oscillators to LO-Based Phase-Shifting Phased-Array Architectures," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 12, pp. 3475 - 3484, Oct. 2010.
- [65] G. Werner, N. Roberto and D. D. Nicola, "A 1.4psrms-period-jitter TDC-less fractional-N digital PLL with digitally controlled ring oscillator in 65nm CMOS," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, CA, USA, 2010.
- [66] E. G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 665 - 692, May 2001.
- [67] Y.-C. Huang and S.-I. Liu, "A 2.4-GHz Subharmonically Injection-Locked PLL With Self-Calibrated Injection Timing," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 2, pp. 417 - 428, Dec. 2012.
- [68] M. Schulz, N. Joram and M. El-Shennawy, "Integrated tri-state PLL for the control of a switched injection-locked oscillator at 2.45 GHz," in *2016 German Microwave Conference (GeMiC)*, Bochum, Germany, 2016.

- [69] J. Wu, Z. Wang and C. Chen, "A 2.4-GHz All-Digital PLL With a 1-ps Resolution 0.9-mW Edge-Interchanging-Based Stochastic TDC," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 10, pp. 917 - 921, July 2015.
- [70] I.-T. Lee, K.-H. Zeng and S.-I. Liu, "A 4.8-GHz Dividerless Subharmonically Injection-Locked All-Digital PLL With a FOM of  $-252.5$  dB," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 9, pp. 547 - 551, July 2013.
- [71] W. Deng, D. Yang and A. T. Narayanan, "14.1 A 0.048mm<sup>2</sup> 3mW synthesizable fractional-N PLL with a soft injection-locking technique," in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, San Francisco, CA, USA, 2015.
- [72] P. Park, J. Park and H. Park, "An all-digital clock generator using a fractionally injection-locked oscillator in 65nm CMOS," in *2012 IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, San Francisco, CA, USA, 2012.
- [73] N. B. Ameer, N. Masmoudi and M. Loulou, "Design and FPGA-based multi-channel, low phase-jitter ADPLL for audio data converter," in *2013 IEEE 11th International New Circuits and Systems Conference (NEWCAS)*, Paris, France, 2013.

## Appendices

The Verilog HDL codes are shown in following. The top view of the proposed work, and the important function blocks, such as DCO, clock tree, thermometer-decoder and shift register is shown in the following pages.

### Top View Codes:

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 17:15:01 10/12/2016
// Design Name: Fully Synthesized Injection Locked Ring Oscillator
// Module Name: ILRO_TOP_VIEW
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module ILRO_TOP_VIEW(
/////////////////////////////////////////////////////////////////
// first buffer tree parameter
/////////////////////////////////////////////////////////////////
    input in_b1,
    //second buffer tree
    input sec_osc_in,
/////////////////////////////////////////////////////////////////
// buffer bridge
/////////////////////////////////////////////////////////////////
    output bridge_out,
    output tail_tail,

/////////////////////////////////////////////////////////////////
// ther1_ther2_one_team
/////////////////////////////////////////////////////////////////
```

```

input clk,
input din,
input reset,
input enable_register,
output testport

    );
wire fianl_output;
wire first_oscillator_test_port;
wire connection_for_sec_buffer_out;
wire enable;
wire final_select;
wire out_buffer_connectionx;
wire sec_buffer_out;
wire [4:0]first_select;
wire [2:0]sec_select;
wire [4:0]buffer_mux_osc2;

/* oscillator_1 output connect with control system */
wire
o1_osc1,o2_osc1,o3_osc1,o4_osc1,o5_osc1,o6_osc1,o7_osc1,o8_osc1,o9_osc1,o10_osc1;
wire
o11_osc1,o12_osc1,o13_osc1,o14_osc1,o15_osc1,o16_osc1,o17_osc1,o18_osc1,o19_osc1,o20_osc1;
wire
o21_osc1,o22_osc1,o23_osc1,o24_osc1,o25_osc1,o26_osc1,o27_osc1,o28_osc1,o29_osc1;
    wire x_osc1;
    wire [9:0]in_ther1;
    wire [7:0]in_ther2;

/* oscillator 1 frequency control part with 10-bits */

wire [959:0] frequency_control_osc1;

/* oscillator 2 frequency control part with 9-bits */

wire [255:0] frequency_control_osc2;

wire [31:0]inj_osc1;

wire [31:0]buffermux;
wire inj1,inj2,inj3,inj4,inj5,inj6,inj7,inj8,inj9,inj10;
wire inj11,inj12,inj13,inj14,inj16,inj17,inj18,inj19,inj20;
wire inj21,inj22,inj23,inj24,inj25,inj26,inj27,inj28,inj29,inj30;
wire inj31,inj32,inj33,inj34,inj35,inj36,inj37,inj38,inj39,inj40;

```

```

wire inj41,inj42,inj43,inj44,inj45,inj46,inj47,inj48,inj49,inj50;
wire inj51,inj52,inj53,inj54,inj55,inj56,inj57,inj58,inj59,inj60;
wire inj61,inj62,inj64;
wire inj15,inj63;
wire o1_osc2,o2_osc2,o3_osc2,o4_osc2,x_osc2;

```

```

////////////////////////////////////
// buffer tree 1
////////////////////////////////////
first_oscillator_tree_mogai2 tree1(

    .first_in(in_b1),
    .en(enable),
    .ot(inj_osc1)
);
////////////////////////////////////
// osc1 functuion
////////////////////////////////////
first_oscillator_new combine_osc1_edition2(

    .en(frequency_control_osc1),
    .inj(inj_osc1),
    .x(x_osc1),
    .o1(o1_osc1),
    .o2(o2_osc1),
    .o3(o3_osc1),
    .o4(o4_osc1),
    .o5(o5_osc1),
    .o6(o6_osc1),
    .o7(o7_osc1),
    .o8(o8_osc1),
    .o9(o9_osc1),
    .o10(o10_osc1),
    .o11(o11_osc1),
    .o12(o12_osc1),
    .o13(o13_osc1),
    .o14(o14_osc1),
    .o15(o15_osc1),
    .o16(o16_osc1),
    .o17(o17_osc1),
    .o18(o18_osc1),
    .o19(o19_osc1),
    .o20(o20_osc1),
    .o21(o21_osc1),
    .o22(o22_osc1),

```

```

        .o23(o23_osc1),
        .o24(o24_osc1),
        .o25(o25_osc1),
        .o26(o26_osc1),
        .o27(o27_osc1),
        .o28(o28_osc1),
        .o29(o29_osc1),
        .o30(o30_osc1)
    );

/////////////////////////////////////////////////////////////////
// buffer connection between oscillator 1 and Mux 1
/////////////////////////////////////////////////////////////////

all_digital_osc1_buffer_chain_edition1 buffer_chain(

    .enable(enable),
    .o0(x_osc1),
    .o1(o1_osc1),
    .o2(o2_osc1),
    .o3(o3_osc1),
    .o4(o4_osc1),
    .o5(o5_osc1),
    .o6(o6_osc1),
    .o7(o7_osc1),
    .o8(o8_osc1),
    .o9(o9_osc1),
    .o10(o10_osc1),
    .o11(o11_osc1),
    .o12(o12_osc1),
    .o13(o13_osc1),
    .o14(o14_osc1),
    .o15(o15_osc1),
    .o16(o16_osc1),
    .o17(o17_osc1),
    .o18(o18_osc1),
    .o19(o19_osc1),
    .o20(o20_osc1),
    .o21(o21_osc1),
    .o22(o22_osc1),
    .o23(o23_osc1),
    .o24(o24_osc1),
    .o25(o25_osc1),
    .o26(o26_osc1),
    .o27(o27_osc1),

```

```

.o28(o28_osc1),
.o29(o29_osc1),
.o30(o30_osc1),
.ot(buffermux),
.o31(in_b1)
);

////////////////////////////////////////////////////////////////
// 1st oscillator output mux ;mux 1 fuction
////////////////////////////////////////////////////////////////

mux_osc1_osc1_mux_new(

.select(first_select),
.d(buffermux),
.out_buffer_connection(out_buffer_connectionx)

);

////////////////////////////////////////////////////////////////
// first mux output test port
////////////////////////////////////////////////////////////////

new_buffer_chain_for_demux_copy_first_osc_test_portx(

.in(out_buffer_connectionx),
.enable(enable),
.out(first_oscillator_test_port)

);

////////////////////////////////////////////////////////////////
// buffer bridge function
////////////////////////////////////////////////////////////////

buffer_chain_connection_bridge(

.en_buffer(enable),
.in_buffer(out_buffer_connectionx),
.out_buffer(bridge_out)
);
////////////////////////////////////////////////////////////////
// thermodecoder 1 function
////////////////////////////////////////////////////////////////

```

```

first_thermodecoder_new ther11(

    .IN(in_ther1),
    .ot(frequency_control_osc1),
    .enable(enable)

);
////////////////////////////////////////////////////////////////
// osc2 function
////////////////////////////////////////////////////////////////
sec_oscillator_education1 combine_osc2_education2(
    .en(frequency_control_osc2),
    .inj1(inj1),.inj2(inj2),.inj3(inj3),.inj4(inj4),.inj5(inj5),.inj6(inj6),.inj7(inj7),.inj8(inj8),.inj9(inj9),.inj10(inj10),
    .inj11(inj11),.inj12(inj12),.inj13(inj13),.inj14(inj14),.inj15(inj15),.inj16(inj16),.inj17(inj17),.inj18(inj18),.inj19(inj19),.inj20(inj20),
    .inj21(inj21),.inj22(inj22),.inj23(inj23),.inj24(inj24),.inj25(inj25),.inj26(inj26),.inj27(inj27),.inj28(inj28),.inj29(inj29),.inj30(inj30),
    .inj31(inj31),.inj32(inj32),.inj33(inj33),.inj34(inj34),.inj35(inj35),.inj36(inj36),.inj37(inj37),.inj38(inj38),.inj39(inj39),.inj40(inj40),
    .inj41(inj41),.inj42(inj42),.inj43(inj43),.inj44(inj44),.inj45(inj45),.inj46(inj46),.inj47(inj47),.inj48(inj48),.inj49(inj49),.inj50(inj50),
    .inj51(inj51),.inj52(inj52),.inj53(inj53),.inj54(inj54),.inj55(inj55),.inj56(inj56),.inj57(inj57),.inj58(inj58),.inj59(inj59),.inj60(inj60),
    .inj61(inj61),.inj62(inj62),.inj63(inj63),.inj64(inj64),
    .x(x_osc2),
    .o1(o1_osc2),
    .o2(o2_osc2),
    .o3(o3_osc2),
    .o4(o4_osc2)
);

////////////////////////////////////////////////////////////////
// buffer_tree_63_k_clk
////////////////////////////////////////////////////////////////
black_tech_test2_mogai2 tree_63_64(

    .enall(enable),
    .din(sec_osc_in),

    .j1(inj1),.j2(inj2),.j3(inj3),.j4(inj4),.j5(inj5),.j6(inj6),.j7(inj7),.j8(inj8),.j9(inj9),.j10(inj10),
    .j11(inj11),.j12(inj12),.j13(inj13),.j14(inj14),.j15(inj15),.j16(inj16),.j17(inj17),.j18(inj18),.j19(inj19),.j20(inj20),
    .j21(inj21),.j22(inj22),.j23(inj23),.j24(inj24),.j25(inj25),.j26(inj26),.j27(inj27),.j28(inj28),.j29(inj29),.j30(inj30),

```

```

.j31(inj31),.j32(inj32),.j33(inj33),.j34(inj34),.j35(inj35),.j36(inj36),.j37(inj37),.j3
8(inj38),.j39(inj39),.j40(inj40),
.j41(inj41),.j42(inj42),.j43(inj43),.j44(inj44),.j45(inj45),.j46(inj46),.j47(inj47),.j4
8(inj48),.j49(inj49),.j50(inj50),
.j51(inj51),.j52(inj52),.j53(inj53),.j54(inj54),.j55(inj55),.j56(inj56),.j57(inj57),.j5
8(inj58),.j59(inj59),.j60(inj60),
.j61(inj61),.j62(inj62),.j63(inj63),.j64(inj64)

);

////////////////////////////////////
// output buffer for 2nd oscillator
////////////////////////////////////

sec_osc_out_buffer sec_out_buffer(

    .en_sec(enable),
    .in1_sec(o1_osc2),
    in2_sec(o2_osc2),
    .in3_sec(o3_osc2),
    .in4_sec(o4_osc2),
    .in5_sec(x_osc2),
    .out_sec(buffer_mux_osc2)
);

////////////////////////////////////
// 2nd oscillator output Mux
////////////////////////////////////

sec_osc_out_mux sec_mux(

    .select(sec_select),
    .d(buffer_mux_osc2),
    .sec_buffer_out(sec_buffer_out)
);

new_buffer_chain_osc2_mux_connection_mogai buffer_chain_for_connection_fianl(

    .in(sec_buffer_out),
    .out(connection_for_sec_buffer_out),
    .enable(enable)
);

```

```

mux_for_final_output final_output_decision(
    .first_oscillator_test_port(first_oscillator_test_port),
    .sec_buffer_out(connection_for_sec_buffer_out),
    .select_final(final_select),
    .fianl_output(fianl_output)
);
final_tail last_stage(
    .en(enable),
    .final(fianl_output),
    .final_tail(tail_tail)
);

```

```

////////////////////////////////////
// thermodecoder 2 function
////////////////////////////////////

```

```

sec_oscillator_thermodecoder ther22(
    .enable(enable),
    .ot(frequency_control_osc2),
    .in(in_ther2)
);

```

```

shift_register_mogai2 shfit_register_test(
    .system_enable(enable),
    .testport(testport),
    .din(din), //signal inpur
    .clk(clk), // clock input
    .reset(reset), // reset signal
    .enable(enable_register),
    .ooutt01(in_ther1[0]),
    .ooutt11(in_ther1[1]),
    .ooutt21(in_ther1[2]),
    .ooutt31(in_ther1[3]),
    .ooutt41(in_ther1[4]),
    .ooutt51(in_ther1[5]),
    .ooutt61(in_ther1[6]),
    .ooutt71(in_ther1[7]),
    .ooutt81(in_ther1[8]),
    .ooutt91(in_ther1[9]),
    .ooutt02(in_ther2[0]),

```

```
.ooutt12(in_ther2[1]),  
.ooutt22(in_ther2[2]),  
.ooutt32(in_ther2[3]),  
.ooutt42(in_ther2[4]),  
.ooutt52(in_ther2[5]),  
.ooutt62(in_ther2[6]),  
.ooutt72(in_ther2[7]),  
//.ooutt82(in_ther2[8]),  
.select1(first_select),  
.select2(sec_select),  
.final_select(final_select)  
//.select_final(select_final)
```

```
);  
endmodule
```

## Thermometer-Decoder Codes:

```
// Module Name:  sec_oscillator_thermodecoder
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module sec_oscillator_thermodecoder(
    enable,in,ot
    );

//wire [7:0] OUT;
reg [255:0] ot;
input enable ;
input [7:0]in;
output [255:0]ot ;

always @(enable or in )begin
    if (!enable) begin
        ot<=0;
    end
    else begin
        ot <=~(~(0)<<(in));
    end
end

endmodule
```

**Cock Tree Codes:**

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 22:12:09 10/19/2016
// Design Name:
// Module Name: second_clock_tree
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module second_clock_tree_(
enall,din,
j1,j2,j3,j4,j5,j6,j7,j8,j9,j10,
j11,j12,j13,j14,j15,j16,j17,j18,j19,j20,
j21,j22,j23,j24,j25,j26,j27,j28,j29,j30,
j31,j32,j33,j34,j35,j36,j37,j38,j39,j40,
j41,j42,j43,j44,j45,j46,j47,j48,j49,j50,
j51,j52,j53,j54,j55,j56,j57,j58,j59,j60,
j61,j62,j63,j64
);
input enall;
input din ;
output j1,j2,j3,j4,j5,j6,j7,j8,j9,j10;
output j11,j12,j13,j14,j15,j16,j17,j18,j19,j20;
output j21,j22,j23,j24,j25,j26,j27,j28,j29,j30;
output j31,j32,j33,j34,j35,j36,j37,j38,j39,j40;
output j41,j42,j43,j44,j45,j46,j47,j48,j49,j50;
output j51,j52,j53,j54,j55,j56,j57,j58,j59,j60;
output j61,j62,j63,j64;

wire w1,w2,w3,w4,w5,wb0,wb1;
```

```
bufifl n0(wb0,din,enall);
bufifl n1(wb1,wb0,enall);
```

```
bufifl n3(w1,wb1,enall);
bufifl n4(w2,w1,enall);
bufifl n5(w3,w2,enall);
bufifl n6(w4,w3,enall);
bufifl n7(w5,w3,enall);
```

```
bufifl b64(j1,w4,enall); //L7
bufifl b65(j2,w4,enall);
bufifl b66(j3,w4,enall);
bufifl b67(j4,w4,enall);
bufifl b68(j5,w4,enall);
bufifl b69(j6,w4,enall);
bufifl b70(j7,w4,enall);
bufifl b71(j8,w4,enall);
bufifl b72(j9,w4,enall);
bufifl b73(j10,w4,enall);
bufifl b74(j11,w4,enall);
bufifl b75(j12,w4,enall);
bufifl b76(j13,w4,enall);
bufifl b77(j14,w4,enall);
bufifl b78(j15,w4,enall);
bufifl b79(j16,w4,enall);
bufifl b80(j17,w4,enall);
bufifl b81(j18,w4,enall);
bufifl b82(j19,w4,enall);
bufifl b83(j20,w4,enall);
bufifl b84(j21,w4,enall);
bufifl b85(j22,w4,enall);
bufifl b86(j23,w4,enall);
bufifl b87(j24,w4,enall);
bufifl b88(j25,w4,enall);
bufifl b89(j26,w4,enall);
bufifl b90(j27,w4,enall);
bufifl b91(j28,w4,enall);
bufifl b92(j29,w4,enall);
bufifl b93(j30,w4,enall);
bufifl b94(j31,w4,enall);
bufifl b95(j32,w4,enall);
bufifl b96(j33,w5,enall);
bufifl b97(j34,w5,enall);
```

```
bufifl b98(j35,w5,enall);
bufifl b99(j36,w5,enall);
bufifl b100(j37,w5,enall);
bufifl b101(j38,w5,enall);
bufifl b102(j39,w5,enall);
bufifl b103(j40,w5,enall);
bufifl b104(j41,w5,enall);
bufifl b105(j42,w5,enall);
bufifl b106(j43,w5,enall);
bufifl b107(j44,w5,enall);
bufifl b108(j45,w5,enall);
bufifl b109(j46,w5,enall);
bufifl b110(j47,w5,enall);
bufifl b111(j48,w5,enall);
bufifl b112(j49,w5,enall);
bufifl b113(j50,w5,enall);
bufifl b114(j51,w5,enall);
bufifl b115(j52,w5,enall);
bufifl b116(j53,w5,enall);
bufifl b117(j54,w5,enall);
bufifl b118(j55,w5,enall);
bufifl b119(j56,w5,enall);
bufifl b120(j57,w5,enall);
bufifl b121(j58,w5,enall);
bufifl b122(j59,w5,enall);
bufifl b123(j60,w5,enall);
bufifl b124(j61,w5,enall);
bufifl b125(j62,w5,enall);
bufifl b126(j63,w5,enall);
bufifl b127(j64,w5,enall);
```

```
endmodule
```

## Digitally Controlled Oscillator Codes:

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 21:00:07 10/05/2016
// Design Name:
// Module Name: sec_oscillator_edition1
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module sec_oscillator_edition1(
en,
    o2,
    o1,
    o3,
    o4,
    x,
    inj1,inj2,inj3,inj4,inj5,inj6,inj7,inj8,inj9,inj10,
    inj11,inj12,inj13,inj14,inj15,inj16,inj17,inj18,inj19,inj20,
    inj21,inj22,inj23,inj24,inj25,inj26,inj27,inj28,inj29,inj30,
    inj31,inj32,inj33,inj34,inj35,inj36,inj37,inj38,inj39,inj40,
    inj41,inj42,inj43,inj44,inj45,inj46,inj47,inj48,inj49,inj50,
    inj51,inj52,inj53,inj54,inj55,inj56,inj57,inj58,inj59,inj60,
    inj61,inj62,inj63,inj64
);

input [255:0]en;
input inj1,inj2,inj3,inj4,inj5,inj6,inj7,inj8,inj9,inj10;
input inj11,inj12,inj13,inj14,inj15,inj16,inj17,inj18,inj19,inj20;
input inj21,inj22,inj23,inj24,inj25,inj26,inj27,inj28,inj29,inj30;
input inj31,inj32,inj33,inj34,inj35,inj36,inj37,inj38,inj39,inj40;
input inj41,inj42,inj43,inj44,inj45,inj46,inj47,inj48,inj49,inj50;
input inj51,inj52,inj53,inj54,inj55,inj56,inj57,inj58,inj59,inj60;
```

```

input inj61,inj62,inj63,inj64;
output o2;
output o1,o3,o4/*,o5,o6*/;
output x ;
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

    notifl t1(o1,x ,en[0]);
    notifl t2(o2,o1,en[1]);
    notifl t3(o3,o2,en[2]);
    notifl t4(o4,o3,en[3]);
    notifl t5(x,o4,inj1);
                                                    //level 1

    notifl t8(o1,x ,en[4]);
    notifl t9(o2,o1,en[5]);
    notifl t10(o3,o2,en[6]);
    notifl t11(o4,o3,en[7]);
    notifl t12(x,o4,inj2);
                                                    // level 2

    notifl t15(o1,x ,en[8]);
    notifl t16(o2,o1,en[9]);
    notifl t17(o3,o2,en[10]);
    notifl t18(o4,o3,en[11]);
    notifl t19(x,o4,inj3);
                                                    // leve 3

    notifl t22(o1,x ,en[12]);
    notifl t23(o2,o1,en[13]);
    notifl t24(o3,o2,en[14]);
    notifl t25(o4,o3,en[15]);
    notifl t26(x,o4,inj4);
                                                    // level 4

    notifl t29(o1,x ,en[16]);
    notifl t30(o2,o1,en[17]);
    notifl t31(o3,o2,en[18]);
    notifl t32(o4,o3,en[19]);
    notifl t33(x,o4,inj5);
                                                    // level 5

```

```
notifl t36(o1,x ,en[20]);
notifl t37(o2,o1,en[21]);
notifl t38(o3,o2,en[22]); // level 6
notifl t39(o4,o3,en[23]);
notifl t40(x,o4,inj6);
```

```
notifl t43(o1,x ,en[24]);
notifl t44(o2,o1,en[25]);
notifl t45(o3,o2,en[26]);
notifl t46(o4,o3,en[27]); // level 7
notifl t47(x,o4,inj7);
```

```
notifl t50(o1,x ,en[28]);
notifl t51(o2,o1,en[29]);
notifl t52(o3,o2,en[30]);
notifl t53(o4,o3,en[31]); // level 8
notifl t54(x,o4,inj8);
```

```
notifl t57(o1,x ,en[32]);
notifl t58(o2,o1,en[33]);
notifl t59(o3,o2,en[34]);
notifl t60(o4,o3,en[35]); // level 9
notifl t61(x,o4,inj9);
```

```
notifl t64(o1,x ,en[36]);
notifl t65(o2,o1,en[37]);
notifl t66(o3,o2,en[38]);
notifl t67(o4,o3,en[39]); // level 10
notifl t68(x,o4,inj10);
```

//  
//  
//

notif1 t71(o1,x ,en[40]);  
notif1 t72(o2,o1,en[41]);  
notif1 t73(o3,o2,en[42]); //level 11  
notif1 t74(o4,o3,en[43]);  
notif1 t75(x,o4,inj11);

notif1 t78(o1,x ,en[44]);  
notif1 t79(o2,o1,en[45]);  
notif1 t80(o3,o2,en[46]);  
notif1 t81(o4,o3,en[47]); // level 12  
notif1 t82(x,o4,inj12);

notif1 t85(o1,x ,en[48]);  
notif1 t86(o2,o1,en[49]);  
notif1 t87(o3,o2,en[50]); // leve 13  
notif1 t88(o4,o3,en[51]);  
notif1 t89(x,o4,inj13);

notif1 t92(o1,x ,en[52]);  
notif1 t93(o2,o1,en[53]);  
notif1 t94(o3,o2,en[54]);  
notif1 t95(o4,o3,en[55]); // level 14  
notif1 t96(x,o4,inj14);

notif1 t99(o1, x ,en[56]);  
notif1 t100(o2,o1,en[57]);  
notif1 t101(o3,o2,en[58]); // level 15  
notif1 t102(o4,o3,en[59]);  
notif1 t103(x,o4,inj15);

notif1 t106(o1,x ,en[60]);  
notif1 t107(o2,o1,en[61]);  
notif1 t108(o3,o2,en[62]); // lvel 16  
notif1 t109(o4,o3,en[63]);  
notif1 t110(x,o4,inj16);

```
notifl t113(o1,x ,en[64]);
notifl t114(o2,o1,en[65]);
notifl t115(o3,o2,en[66]);
notifl t116(o4,o3,en[67]); // level 17
notifl t117(x,o4,inj17);
```

```
notifl t120(o1,x ,en[68]);
notifl t121(o2,o1,en[69]);
notifl t122(o3,o2,en[70]);
notifl t123(o4,o3,en[71]); // level 18
notifl t124(x,o4,inj18);
```

```
notifl t127(o1,x ,en[72]);
notifl t128(o2,o1,en[73]);
notifl t129(o3,o2,en[74]);
notifl t130(o4,o3,en[75]); // level 19
notifl t131(x,o4,inj19);
```

```
notifl t134(o1,x ,en[76]);
notifl t135(o2,o1,en[77]);
notifl t136(o3,o2,en[78]);
notifl t137(o4,o3,en[79]); // level 20
notifl t138(x,o4,inj20);
```

```
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
```

```
notifl t141(o1,x ,en[80]);
notifl t142(o2,o1,en[81]);
notifl t143(o3,o2,en[82]); //level 21
```

notif1 t144(o4,o3,en[83]);  
notif1 t145(x,o4,inj21);

notif1 t148(o1,x ,en[84]);  
notif1 t149(o2,o1,en[85]);  
notif1 t150(o3,o2,en[86]);  
notif1 t151(o4,o3,en[87]); // level 22  
notif1 t152(x,o4,inj22);

notif1 t155(o1,x ,en[88]);  
notif1 t156(o2,o1,en[89]);  
notif1 t157(o3,o2,en[90]); // leve 23  
notif1 t158(o4,o3,en[91]);  
notif1 t159(x,o4,inj23);

notif1 t162(o1,x ,en[92]);  
notif1 t163(o2,o1,en[93]);  
notif1 t164(o3,o2,en[94]);  
notif1 t165(o4,o3,en[95]); // level 24  
notif1 t166(x,o4,inj24);

notif1 t169(o1,x ,en[96]);  
notif1 t170(o2,o1,en[97]);  
notif1 t171(o3,o2,en[98]); // level 25  
notif1 t172(o4,o3,en[99]);  
notif1 t173(x,o4,inj25);

notif1 t176(o1,x ,en[100]);  
notif1 t177(o2,o1,en[101]);  
notif1 t178(o3,o2,en[102]); // lvel 26  
notif1 t179(o4,o3,en[103]);  
notif1 t180(x,o4,inj26);

notif1 t183(o1,x ,en[104]);  
notif1 t184(o2,o1,en[105]);  
notif1 t185(o3,o2,en[106]);  
notif1 t186(o4,o3,en[107]); // level 27  
notif1 t187(x,o4,inj27);

notif1 t190(o1,x ,en[108]);

```
notifl t191(o2,o1,en[109]);
notifl t192(o3,o2,en[110]);
notifl t193(o4,o3,en[111]); // level 28
notifl t194(x,o4,inj28);
```

```
notifl t197(o1,x ,en[112]);
notifl t198(o2,o1,en[113]);
notifl t199(o3,o2,en[114]);
notifl t200(o4,o3,en[115]); // level 29
notifl t201(x,o4,inj29);
```

```
notifl t204(o1,x ,en[116]);
notifl t205(o2,o1,en[117]);
notifl t206(o3,o2,en[118]);
notifl t207(o4,o3,en[119]); // level 30
notifl t208(x,o4,inj30);
```

```
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
```

```
notifl t211(o1,x ,en[120]);
notifl t212(o2,o1,en[121]);
notifl t213(o3,o2,en[122]); //level 31
notifl t214(o4,o3,en[123]);
notifl t215(x,o4,inj31);
```

```
notifl t218(o1,x ,en[124]);
notifl t219(o2,o1,en[125]);
notifl t220(o3,o2,en[126]);
notifl t221(o4,o3,en[127]); // level 32
notifl t222(x,o4,inj32);
```

```
notifl t225(o1,x ,en[128]);
notifl t226(o2,o1,en[129]);
notifl t227(o3,o2,en[130]); // leve 33
notifl t228(o4,o3,en[131]);
notifl t229(x,o4,inj33);
```

```
notifl t232(o1,x ,en[132]);
notifl t233(o2,o1,en[133]);
notifl t234(o3,o2,en[134]);
notifl t235(o4,o3,en[135]); // level 34
notifl t236(x,o4,inj34);
```

notifl t239(o1,x ,en[136]);  
notifl t240(o2,o1,en[137]);  
notifl t241(o3,o2,en[138]); // level 35  
notifl t242(o4,o3,en[139]);  
notifl t243(x,o4,inj35);

notifl t246(o1,x ,en[140]);  
notifl t247(o2,o1,en[141]);  
notifl t248(o3,o2,en[142]); // lvel 36  
notifl t249(o4,o3,en[143]);  
notifl t250(x,o4,inj36);

notifl t253(o1,x ,en[144]);  
notifl t254(o2,o1,en[145]);  
notifl t255(o3,o2,en[146]);  
notifl t256(o4,o3,en[147]); // level 37  
notifl t257(x,o4,inj37);

notifl t260(o1,x ,en[148]);  
notifl t261(o2,o1,en[149]);  
notifl t262(o3,o2,en[150]);  
notifl t263(o4,o3,en[151]); // level 38  
notifl t264(x,o4,inj38);

notifl t267(o1,x ,en[152]);  
notifl t268(o2,o1,en[153]);  
notifl t269(o3,o2,en[154]);  
notifl t270(o4,o3,en[155]); // level 39  
notifl t271(x,o4,inj39);

notifl t274(o1,x ,en[156]);  
notifl t275(o2,o1,en[157]);  
notifl t276(o3,o2,en[158]);  
notifl t277(o4,o3,en[159]); // level 40  
notifl t278(x,o4,inj40);

////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////

notifl t281(o1,x ,en[160]);  
notifl t282(o2,o1,en[161]);  
notifl t283(o3,o2,en[162]); //level 41  
notifl t284(o4,o3,en[163]);  
notifl t285(x,o4,inj41);

```
notifl t288(o1,x ,en[164]);
notifl t289(o2,o1,en[165]);
notifl t290(o3,o2,en[166]);
notifl t291(o4,o3,en[167]); // level 42
notifl t292(x,o4,inj42);
```

```
notifl t295(o1,x ,en[168]);
notifl t296(o2,o1,en[169]);
notifl t297(o3,o2,en[170]); // leve 43
notifl t298(o4,o3,en[171]);
notifl t299(x,o4,inj43);
```

```
notifl t302(o1,x ,en[172]);
notifl t303(o2,o1,en[173]);
notifl t304(o3,o2,en[174]);
notifl t305(o4,o3,en[175]); // level 44
notifl t306(x,o4,inj44);
```

```
notifl t309(o1,x ,en[176]);
notifl t310(o2,o1,en[177]);
notifl t311(o3,o2,en[178]); // level 45
notifl t312(o4,o3,en[179]);
notifl t313(x,o4,inj45);
```

```
notifl t316(o1,x ,en[180]);
notifl t317(o2,o1,en[181]);
notifl t318(o3,o2,en[182]); // lvel 46
notifl t319(o4,o3,en[183]);
notifl t320(x,o4,inj46);
```

```
notifl t323(o1,x ,en[184]);
notifl t324(o2,o1,en[185]);
notifl t325(o3,o2,en[186]);
notifl t326(o4,o3,en[187]); // level 47
notifl t327(x,o4,inj47);
```

```
notifl t330(o1,x ,en[188]);
notifl t331(o2,o1,en[189]);
notifl t332(o3,o2,en[190]);
notifl t333(o4,o3,en[191]); // level 48
notifl t334(x,o4,inj48);
```

```
notifl t337(o1,x ,en[192]);
notifl t338(o2,o1,en[193]);
```

```
notifl t339(o3,o2,en[194]);
notifl t340(o4,o3,en[195]);           // level 49
notifl t341(x,o4,inj49);
```

```
notifl t344(o1,x ,en[196]);
notifl t345(o2,o1,en[197]);
notifl t346(o3,o2,en[198]);
notifl t347(o4,o3,en[199]);         // level 50
notifl t348(x,o4,inj50);
```

```
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
```

```
notifl t351(o1,x ,en[200]);
notifl t352(o2,o1,en[201]);
notifl t353(o3,o2,en[202]);         //level 51
notifl t354(o4,o3,en[203]);
notifl t355(x,o4,inj51);
```

```
notifl t358(o1,x ,en[204]);
notifl t359(o2,o1,en[205]);
notifl t360(o3,o2,en[206]);
notifl t361(o4,o3,en[207]);         // level 52
notifl t362(x,o4,inj52);
```

```
notifl t365(o1,x ,en[208]);
notifl t366(o2,o1,en[209]);
notifl t367(o3,o2,en[210]);         // leve 53
notifl t368(o4,o3,en[211]);
notifl t369(x,o4,inj53);
```

```
notifl t372(o1,x ,en[212]);
notifl t373(o2,o1,en[213]);
notifl t374(o3,o2,en[214]);
notifl t375(o4,o3,en[215]);         // level 54
notifl t376(x,o4,inj54);
```

```
notifl t379(o1,x ,en[216]);
notifl t380(o2,o1,en[217]);
notifl t381(o3,o2,en[218]); // level 55
notifl t382(o4,o3,en[219]);
notifl t383(x,o4,inj55);
```

```
notifl t386(o1,x ,en[220]);
notifl t387(o2,o1,en[221]);
notifl t388(o3,o2,en[222]); // lvel 56
notifl t389(o4,o3,en[223]);
notifl t390(x,o4,inj56);
```

```
notifl t393(o1,x ,en[224]);
notifl t394(o2,o1,en[225]);
notifl t395(o3,o2,en[226]);
notifl t396(o4,o3,en[227]); // level 57
notifl t397(x,o4,inj57);
```

```
notifl t400(o1,x ,en[228]);
notifl t401(o2,o1,en[229]);
notifl t402(o3,o2,en[230]);
notifl t403(o4,o3,en[231]); // level 58
notifl t404(x,o4,inj58);
```

```
notifl t407(o1,x ,en[232]);
notifl t408(o2,o1,en[233]);
notifl t409(o3,o2,en[234]);
notifl t410(o4,o3,en[235]); // level 59
notifl t411(x,o4,inj59);
```

```
notifl t414(o1,x ,en[236]);
notifl t415(o2,o1,en[237]);
notifl t416(o3,o2,en[238]);
notifl t417(o4,o3,en[239]); // level 60
notifl t418(x,o4,inj60);
```



## S/P Shift Register Codes:

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 21:06:34 10/18/2016
// Design Name:
// Module Name: final_register_v2
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module final_register_v2(
din ,clk*/,dout*/ ,reset,enable,
ooutt01,ooutt11,ooutt21,ooutt31,ooutt41,ooutt51,ooutt61,ooutt71,ooutt81,ooutt91,
ooutt02,ooutt12,ooutt22,ooutt32,ooutt42,ooutt52,ooutt62,ooutt72,select1,testport,final_
lect,system_enable);

//output [26:0] dout ;
output testport;

wire [20:0] dout ;
input reset;
input din ;
wire din ;
input clk ;
wire clk ;
reg [20:0]s;
//// output buffer
input enable ;
reg [20:0] final_out;
output ooutt01,ooutt11,ooutt21,ooutt31,ooutt41,ooutt51,ooutt61,ooutt71,ooutt81,ooutt91;
output ooutt02,ooutt12,ooutt22,ooutt32,ooutt42,ooutt52,ooutt62,ooutt72;
output select1;
//output [2:0]select2;
```

```

output final_select;
output system_enable;
/*output [18:0]oout;
wire [18:0] oout;
*/
always @( posedge (clk) or posedge (reset) ) begin
if (reset )
s <= 21'b00000000000000000000;
else begin
s[20]<=din;
s[19]<=s[20];
s[18]<=s[19];
s[17]<=s[18];
s[16]<=s[17];
s[15]<=s[16];
s[14]<=s[15];
s[13]<=s[14];
s[12]<=s[13];
s[11]<=s[12];
s[10]<=s[11];
s[9]<=s[10];
s[8]<=s[9];
s[7]<=s[8];
s[6]<=s[7];
s[5]<=s[6];
s[4]<=s[5];
s[3]<=s[4];
s[2]<=s[3];
s[1]<=s[2];
s[0]<=s[1];
end
end
assign dout = s;
assign testport = s[0];
always @ (posedge enable ) begin
final_out[0] <= dout[0];
final_out[1] <= dout[1];
final_out[2] <= dout[2];
final_out[3] <= dout[3];
final_out[4] <= dout[4];
final_out[5] <= dout[5];
final_out[6] <= dout[6];
final_out[7] <= dout[7];
final_out[8] <= dout[8];
final_out[9] <= dout[9];
final_out[10] <= dout[10];

```

```

        final_out[11] <= dout[11];
        final_out[12] <= dout[12];
        final_out[13] <= dout[13];
        final_out[14] <= dout[14];
        final_out[15] <= dout[15];
        final_out[16] <= dout[16];
        final_out[17] <= dout[17];
        final_out[18] <= dout[18];
        final_out[19] <= dout[19];
        final_out[20] <= dout[20];
        /*final_out[26] <= dout[26];*/
end
assign ooutt02 = final_out[0];
assign ooutt12 = final_out[1];
assign ooutt22 = final_out[2];
assign ooutt32 = final_out[3];
assign ooutt42 = final_out[4];
assign ooutt52 = final_out[5];
assign ooutt62 = final_out[6];
assign ooutt72 = final_out[7];
/*assign ooutt82 = final_out[8];*/
assign ooutt01 = final_out[8];
assign ooutt11 = final_out[9];
assign ooutt21 = final_out[10];
assign ooutt31 = final_out[11];
assign ooutt41 = final_out[12];
assign ooutt51 = final_out[13];
assign ooutt61 = final_out[14];
assign ooutt71 = final_out[15];
assign ooutt81 = final_out[16];
assign ooutt91 = final_out[17];
assign select1 = final_out[18];
assign final_select = final_out[19];
assign system_enable = final_out[20];

endmodule

```