Investigation of Analog Calibration Systems for Spurious Tone Suppression in Frequency Triplers

CRISTIAN ALEXANDRU GHIHANIS
BARATH SANTHANAKRISHNAN SUDARSAN COPPARAM
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

LUND UNIVERSITY

Investigation of Analog Calibration Systems for Spurious Tone Suppression in Frequency Triplers

Cristian Alexandru Ghihanis
cr5474gh-s@student.lu.se

Barath Santhanakrishnan Sudarsan Copparam
ba4268co-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Academic Supervisor: Baktash Behmanesh
Company Supervisor: Magnus Wasting, Nordic Semiconductor

Examiner: Pietro Andreani

June 14, 2023

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

The market for Wi-Fi receiver designs for latest Wi-Fi standards, that cover RF bands in the 2.4 GHz, 5 GHz and 6 GHz spectrum, require increasingly stringent power consumption limitations as more of the market is driven towards battery-powered devices. In order to meet those needs, it is crucial to explore different ways of reducing power consumption in various receiver blocks. One such power consumer can be the LO generating stage. As present standards require higher operating frequencies, circuit blocks such as frequency synthesizers, filters and amplifiers lead to increased power requirements in the receiver chain. In addition, issues such as spurious tones from inductive coupling, as well as IM2 and IM3 mixing products, may necessitate additional filtering stages, leading to stricter requirements for the remainder of the circuit. This thesis explores the possibility of reducing power consumption in the LO generation stage by implementing analog calibration schemes for a frequency synthesizer block, which would enable running the LO source at a lower frequency. The LO is then brought to the required frequency through the use of the Edge-Combining Delay-Locked Loop (EC DLL) based synthesizer. This would relax energy requirements in the LO stage, which can then be better used in the rest of the receiver chain. Two self-calibration systems are proposed for the synthesizer. A mathematical support is derived and explained, formulating expressions which help quantifying the expected synthesizer performance, denoted as Spur-to-Carrier Ratio (SCR). A design flow for optimal spurious performance in EC DLL is suggested. Then, a design for the calibration mechanisms is constructed and simulated in $22nm$ CMOS technology. Specifications for the required performance are derived based on the IEEE802.11ax standard interferer scenarios and final results are discussed. The circuit is only realised as a proof-of-concept schematic design with a power supply of $800mV$. The final circuit improves the SCR performance by $7.7dB$ with no significant power consumption requirements added to the EC DLL circuit. The worst-case, total calibration time required for spurious tones compensation is 75.21μs. At the end, a conclusion is formulated regarding the feasibility of this approach.

Popular Science Summary

The technology in today's world is in a continuous evolution. The push is always towards solutions that are more inter-connected, consume less battery power and provide better services to meet users' needs. Moreover, future devices will need to gather and communicate even more data between themselves to better identify what is demanded of them and what is the best solution for this. For example, if at the end of every working day you prefer to take a long bath, then your phone should be able to gather all of this information, identify your needs and communicate them to the smart devices in your bathroom to have it prepared by the time you arrive home.

An easy and already available solution to provide all of this communication quick and efficient could be to use Wi-Fi to integrate future smart devices. Wi-Fi networks are already available all around us in our everyday life, especially in everyone's homes. However, with increasing worry for sustainable technology and ever increasing energy demands, the devices of next generation should use as little energy as possible for their communications.

This thesis aims to provide one possible solution for more battery-efficient wireless communication, in the form of a hardware component. When two devices communicate, each of them uses a physical transmitter to send data, a physical receiver to intercept it and an antenna, which actually converts the information of interest to signal waves in the radio frequency (RF) spectrum. These RF signals travel through the air, over long distances, to transfer information between two devices. The good thing is that all of the infrastructure for these Wi-Fi telecommunications is already there.

So how can they be made more energy efficient? Well, a big part of Wi-Fi communications is represented by time spent waiting and receiving information. So, if the receiver can be constructed in a way that consumes less power, while maintaining the same performance, then a lot of battery life could be saved. This work tries to provide a solution: by investigating a different approach for a critical part of the receiver hardware, and checking if the effect can lead to smart devices that can last even longer with no changes in their batteries.

List of Acronyms

ADC	Analog-to-Digital Converter.
ARP	Antenna Reference Point.
ASIC	Application-Specific Integrated Circuit.
AWS	Advanced Wireless Services.
BB	Base Band.
BPSK	Binary Phase-Shifting Keying modulation.
BW	Bandwidth.
CP	Charge Pump.
CPU	Central Processing Unit.
CSDC	Current Starved Delay Cell.
DCD	Duty Cycle Distortion.
DCR	Direct-Conversion Receivers.
DLL	Delay-Locked Loop.
DSC	Delay-Sensing Circuit.
DTC	Digital-to-Time Converter.
EC	Edge-Combining.

FB	VCDL output for the Frequency Synthesizer.
FFT	Fast Fourier Transform.
GCD	Greatest Common Denominator.
IC	Integrated Circuit.
IF	Intermidiate Frequency.
IIP	Input-referred Intercept Point.
IM	Intermodulation product.
ITU	International Telecommunication Union.
LNA	Low-Noise Amplifier.
LO	Local Oscillator.
LP	Loop Filter.
LSB	Least Significant Bit.
LTE	Long-Term Evolution.
MCS	Modulation Coding Scheme.
PD	Phase Detector.
PDF	Probability Density Function.
PLL	Phase-Locked Loop.
PVT	Process, Voltage and Temperature variations.
Q	Quality factor.
QAM	Quadrature Amplitude Modulation.
REF	Reference signal for the Frequency Synthesizer.
RF	Radio Frequency.

RMS	Root Mean Square.
SCR	Spur-to-Carrier Ratio.
SD	Standard Deviation.
SNR	Signal-to-Noise Ratio.
SPE	Static Phase Error.
TSPC	True Single-Phased Clock.
VCDL	Voltage-Controlled Delay Line.
VCO	Voltage-Controlled Oscillator.
WLAN	Wireless Local Area Network.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

The interest for wireless connectivity in today's world is increasing at a rapid pace. According to [1], the telecommunication sector is critical for enabling the development of future technologies in the battle for a sustainable future. As the market for wireless devices grows every year [2], as well as the amount of Wi-Fi traffic [3], the need for new standards which can sustain the evolving interest for interconnection drives the industry towards developing new standards which can meet the demands, such as $5/6G$ and Wi-Fi $6/6E$.

As stated in [1] and [4], the industry is slowly focusing towards more energy-efficient solutions that can still support increasing number of parallel communications between devices such as complex sensor arrays and wearable Internet-of-Things (IoT) devices. The two technologies work in parallel, however they are meant to cover different scenarios. Whilst $5G$ improves power consumption by enabling low-power modes during low and medium traffic, the Wi-Fi Alliance is pushing towards innovative power consumption features with Wi-Fi 6 for the existing $2.4GHz$ and $5GHz$ bands. This facilitates the possibility of lowering power consumption needs for new technological applications whilst maintaining compatibility with previous ones.

Based on information from [5], Wi-Fi 6 is addressing a set of needs from the industry, one of the main ones being high-density connectivity, defined by multiple parallel transmissions with different devices at the same time. Some of these devices require more power consumption optimisation features, while others can require very high throughput rates for applications such as $4K$ video streaming or Extended Reality (XR). These devices can range from everyday items such as smartphones and laptops to more dedicated ones such as smart watches or smart household appliances. With so many consumers communicating simultaneously, power consumption in wireless transceivers needs to be kept to a minimal.

According to [6], upcoming CMOS transceiver components for Wi-Fi 6 applications will necessitate Power Amplifiers with require higher linearity and flatter response curves for the transmitters, as well as Low-Noise Amplifiers (LNA) with an even lower noise threshold for receivers. Moreover, newer standards such as Wi-Fi $6E$ would require the use of higher operating frequencies, which could expand on those issues. Such requirements translate into either more complex blocks, result-

ing in larger silicon area and higher production costs, or more power consumption for the previously mentioned circuits, forcing more stringent requirements for the rest of the transceiver to preserve competitive energy consumption figures.

For Wi-Fi receiver chains, one of the main "power consumers" is the LO generating circuit in the mixing stage which need to be run at even higher frequencies then the desired carrier. The spectrum of the generated LO signal is usually accompanied by undesired spurious tones which can down-convert big interferers in-band with the carrier, making the information it contains irretrievable. The typical solution is to add a filter in the LO generating component which heavily attenuates the spurious tones, however a filter operating at such high frequencies heavily increases the total power consumption of the receiver chain.

A better solution would be running the LO core at a lower frequency and employing a Frequency Synthesizer circuit, such as an Edge-Combining Delay-Locked Loop (EC DLL) architecture, to bring the LO to the desired frequency. This solution would, however, suffer from the same spurious tones issues mentioned before.

Unless, an analog self-calibration system could be implemented which can minimise or totally suppress the generation of spurious tones in the EC DLL circuit, which is the aim of this master thesis work. This calibration system could lead to lower performance requirements and less power consumption in the LO generation circuit, which can be redistributed to other blocks of the transceiver.

1.2 Typical Wi-Fi receiver

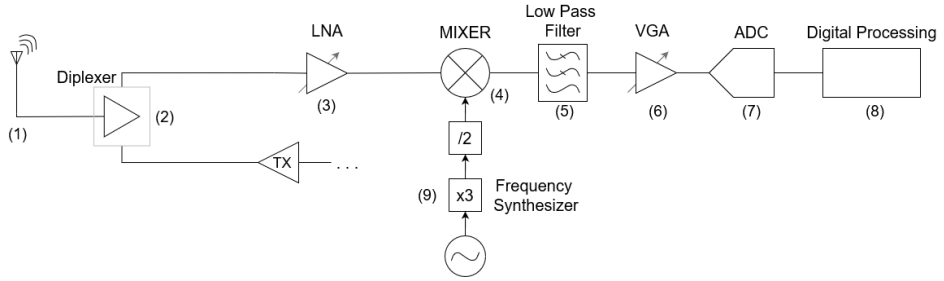

Figure 1.1: Receiver chain block diagram

Figure 1.1 illustrates a typical Wi-Fi receiver chain similar to the one considered for the context of this thesis. The first part of the chain is the antenna represented by block (1) in the figure above, which absorbs the energy of RF signals transmitted through the atmosphere and converts it to electrical signals. The antenna is usually designed with an impedance of 50 Ω for RF applications.

The next block is a Diplexer component, represented by (2), which enables the use of the same antenna component for both Receiver and Transmitter chain. Moreover, a Diplexer component also provides some attenuation to the amplitude of unwanted interferers, similar to the effect of a band-pass filter. Because the Diplexer is chosen in accordance to the input frequency range expected of the

transceiver device, there should be no attenuation for the expected frequencies of carriers.

Afterwards, a special Low-Noise Amplifier (LNA) block is represented by (3). Its purpose is to provides a high amplification factor in the beginning stages of the receiver to reduce the impact of noise generation in future receiver blocks, according to the Friis formula for noise figure. Typical LNA designs ensure a gain of around $25dB$ combined with either a large operation bandwidth with a flat frequency response over several GHz or variable resonant circuits for each RF band. Here, an LNA equipped with an LC tank is considered, which also provides some attenuation effect over any interferers, lowering requirements for LO generation.

The mixer (4) performs frequency mixing between the desired RF signal and the generated LO. The purpose of this stage is to ensure the down-conversion of the carrier to Intermediate Frequency (IF) or Base Band (BB) frequency range through frequency multiplication operations which creates new frequency components for all harmonic content of the input signals. Its output will contain the desired RF signal at a much lower frequency. This is of great help for the forthcoming stages, because they can be designed for lower frequencies, making them less sensitive to parasitic capacitance. Here, the BB scenario is considered, ensuring that the output signal will appear in the 0 to $10MHz$ frequency range.

The LO generating stage contains the component designed in this thesis: a Frequency Tripler block (9) which multiplies the frequency of the input clock signal by a factor of 3. The output of the block will afterwards experience a frequency division operation, which brings the LO frequency in the frequency range desired for the application.

Thus, the BB filtering stage (5) follows which can be designed as a Low-pass with a cutoff frequency of $10MHz$. This is a typical approach for Direct-Conversion Receivers (DCR, also known as homodyne or zero-IF) as the carrier produced at the mixing stage is usually centred at DC. If the mixing output would be of IF range, then the filtering would need to cover a certain frequency range, qualifying as a Band-pass. In this project, a BB filter is assumed which needs to attenuate any unwanted frequency components close to the carrier.

Lastly, the weak signal at the output of the BB filter is then amplified to a desired level by a Variable-Gain Amplifier (6), operating also in the BB range. The output of this stage needs to be large enough to be detected by the input of an Analog-to-Digital Converter (ADC) (7), which is the last analog block before transforming the signal into information in the digital domain. The ADC should have its own anti-aliasing filter to prevent frequency folding. The information is then decoded, processed and digitally filtered before being sent to the digital processing (8) blocks of the Wi-Fi device.

1.3 Thesis scope

This thesis started as a research request from Nordic Semiconductor to investigate whether the power consumption for the Nordic's Wi-Fi receiver could be lowered by changing the current Local Oscillator (LO) generating circuit, which uses a PLL

and filter that operate at around 10 GHz. The new version would operate the PLL at a lower frequency (3.333 GHz) and then feed the generated signal as an input to the synthesizer to generate a similar 10 GHz clock signal. The need for such a circuit comes in response to more stringent power consumption requirements in newer generation Wi-Fi receiver designs, which necessitate innovative ways of reducing the energy needed by the more power-hungry blocks, such as the filter block in the LO circuit.

In theory, the suggested solution would solve the issue as long as the circuit would, ideally, not generate any new spurious tones which could down-convert interferer signals during the mixing stage, putting even more stringent requirements on later stages of the receiver circuit. However, in real life, the manufacturing process for circuits guarantees that all Integrated Circuits (ICs) will have some amount of mismatches during production which will increase the probability for higher amplitude spurs being generated in the Frequency Synthesizer block.

Of course, these could always be alleviated manually by connecting every chip that has been produced to a spectrum analyser device and identifying the correct settings to offset mismatches in the circuit. However, this solution would heavily increase the testing time and money requirement / IC. Thus, the purpose of this thesis project: to derive a on-chip, self-calibration system for the Frequency Synthesizer block, which can minimise or completely suppress the spurious tones generated in the LO block of the receiver device.

This thesis is a proof-of-concept for spurious tones reduction using analog self-calibration systems to correct the effect of circuit imperfections. The focus will be on: determining the improvement in synthesizer spurious performance, as well as the power consumption and total calibration time needed for the final circuit.

The digital logic employed for the calibration systems will not be implemented, instead a set of Python and Matlab scripts will be used to simulate the workflow of the calibration procedure. Design optimisations for proper performance across Process, Voltage and Temperature variations (PVT) is outside the scope of the thesis. Similarly, no layout implementation will be covered due to time constraints. No correction circuit for filtering imperfections in the generated LO signal will be covered. No limit on area was set, however sizes for capacitors need to be maintained within pF ranges. The power supply voltage is set to $800mV$ in the context of a $22nm$ CMOS technology being used.

1.4 Thesis structure

A short list of the following chapters in this thesis report, together with a brief explanation of their purpose:

- Chapter 2, Theory support: A detailed explanation of concepts and mathematical foundations necessary for the proper derivation of specifications and theoretical fundamentals for self-calibration mechanisms.
- Chapter 3, Wi-Fi standard and derivation of Specifications: What are the expected performance metrics for the Frequency Synthesizer component in relation with interferer scenarios applied to the Receiver chain presented in Section 1.2.

- Chapter 4, Self-calibration mechanisms: An in-depth description and explanation of the components, behaviour and the mathematical basis of the calibration mechanisms used for spur suppression in the Frequency Synthesizer circuit.
- Chapter 5, Circuit implementation: A presentation and explanation of the design of all sub-blocks included in the self-calibration routines. How do they work? Are there any special considerations for their behaviour? What is their performance inside the overall circuit?
- Chapter 6, Results and discussion: A presentation of final results illustrating the impact of the self-calibration circuits. A summary of the whole thesis, together with comments regarding operation and limitations of the calibration procedure and possible future improvements.
- Chapter 7, Conclusions: A short summary with conclusions of the overall thesis. Can this approach boost the performance of the Frequency Synthesizer enough to motivate a replacement of the current LO generation circuit?

Chapter 2

Theory support

2.1 Mathematical representation of signals and harmonics

All of the theory presented in this section are based on information from [7].

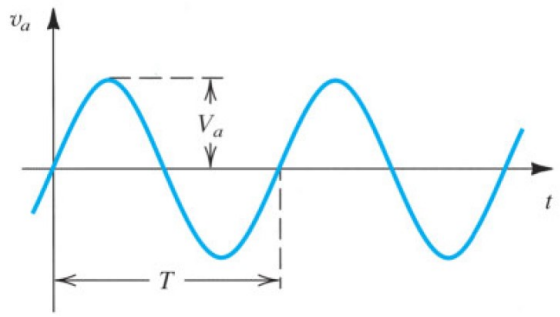

Figure 2.1: A sinusoidal signal, represented in time domain [7]

A signal containing information about the physical world can be represented mathematically as a function. In the case of electrical signals, they can be represented as a time-varying quantity, where the information from the signal could be represented as a change in amplitude, phase or frequency. This is called time domain representation of a signal, such as in Figure 2.1. An example of such a signal could be the output generated by a voltage source, represented below:

$$v_a(t) = V_a sin(\omega t + \phi) \tag{2.1}$$

$$\omega = 2\pi f, \quad f = \frac{1}{T}$$

Where, V_a is the maximum amplitude of the signal [Volts], ω is the angular velocity of the signal [radians/sec], ϕ is the initial phase offset of the signal [radians], f is the frequency of the signal [Hertz] and T is the period of the signal [sec].

If a signal is a periodic function, then by using the Fourier transformation, it can be represented as an infinite sum of sinusoidal waveforms whose frequencies are harmonically related, unless the signal being represented is a pure tone. In

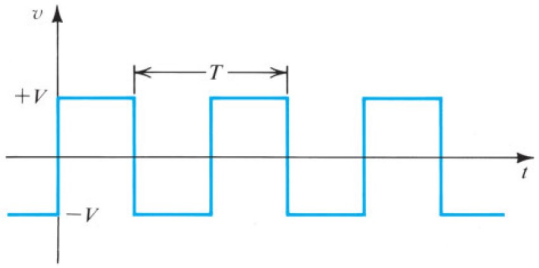

Figure 2.2: A square wave signal, represented in time domain [7]

reality few signals are perfect sine waves, then part of the energy of the electrical signal will always be split between its fundamental component (the fundamental frequency of the signal) and some smaller sinusoidal components, usually of higher frequencies, denoted as harmonics. An example of such a signal would be a periodic square wave signal, presented in Figure 2.2

$$v(t) = \frac{4V}{\pi}(sin(\omega_0 t) + \frac{1}{3}sin(3\omega_0 t) + \frac{1}{5}sin(5\omega_0 t) + ...) \tag{2.2}$$

Where $\omega_0 = \frac{2\pi}{T}$ is the fundamental frequecy of the square wave signal.

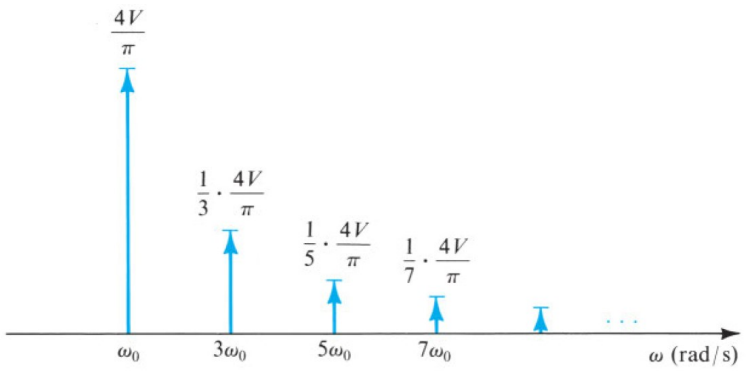

Figure 2.3: A square wave signal, represented in frequency domain, referenced from[7]

In this case, the higher frequency components ($3\omega_0$, $5\omega_0$ etc.) represent its harmonics, which appear at multiplies of the fundamental frequency of the signal being represented. (visible in Figure 2.3)

However, few signals are truly periodic in practical integrated circuits (ICs), and these imperfections translate into the signal being represented by a whole spectrum of frequency components, not only the fundamental and its harmonics. This effect translates into one possible source of noise in an IC.

2.2 Frequency-Synthesizer architecture

The common types of Frequency-Synthesizer architectures used for RF transceivers are Phase-Locked Loop (PLL) and Delay-Locked Loop (DLL) circuits.

Multiple variations of Frequency Synthesizer architectures exist for high-frequency clock signals, however, for as the sole purpose of the block is to triple the frequency of the received signal, an Edge-Combining Delay-Locked Loop (EC DLL) based Frequency Synthesizer circuit has been chosen. The EC DLL architecture is of high interest to this day, with papers such as [8] and [9] exploring methods of reducing spurious performance, while others such as [10], [11] and [12] reveal industry's interest in this architecture for future wireless communication applications.

In this thesis project, an EC DLL-based circuit is employed as a frequency multiplier. As such, this section has the purpose of describing the characteristics of such a Frequency-Synthesizer architecture, along with its uses, drawbacks, and sources of spurious tones. A general EC DLL circuit is illustrated in Figure 2.4.

The main selling points for EC DLL-based Frequency Synthesizers are its low power consumption, low chip area usage with good phase noise performance, as well as the lack of jitter accumulation compared with its PLL counterpart, due to the way the DLL design functions. The architecture suffers from limited Bandwidth, a large setting time and a high sensitivity to the accuracy of the input clock signal, besides being able to produce a very limited set of output frequencies because of its multiplication behaviour of the input frequency. However, for the purposes of this thesis, the disadvantages do not pose a problem, as re-calibration procedures are common to maintain IC performance during operation and the self-calibration procedure should improve stability across voltage and temperature shifts.

2.2.1 Edge Combining Delay-Locked Loop architecture overview

In a DLL circuit, the reference input signal is pushed through a series of N cascaded Delay Gate stages, commonly denoted as a Voltage-Controlled Delay Line (VCDL), where each gate applies a certain amount of phase offset to the reference signal. The purpose of the DLL circuit is to ensure a perfect 360^o phase offset between the input reference and the VCDL output signals. This ensures that each Delay Stage adds an phase offset of $\frac{360^o}{N}$, obtaining a total of $N-1$ offsetted signals, which are then combined by the EC circuit to obtain a clock signal with higher frequency. The number of Delay Stages employed determines the output frequency of the Frequency-Synthesizer block, and different types of EC blocks can result in more flexibility regarding the choice of output frequency for the synthesised signal.

To obtain this effect, the rising edge of the VCDL output is compared with that of the reference signal in order to ascertain the difference in phase between the two signals. This comparison takes place in the Phase Detector/ Comparator (PD) component, which then outputs two signals, generally denoted as "UP" and "DN" towards the Charge Pump (CP) circuit. The pulse width of the two control signals depend on the difference between rising edges of the reference and VCDL output signals. The "UP" signal pushes the delay offset generated by the VCDL cells to a lower delay offset, thus a higher operating frequency, whilst the "DOWN" signal

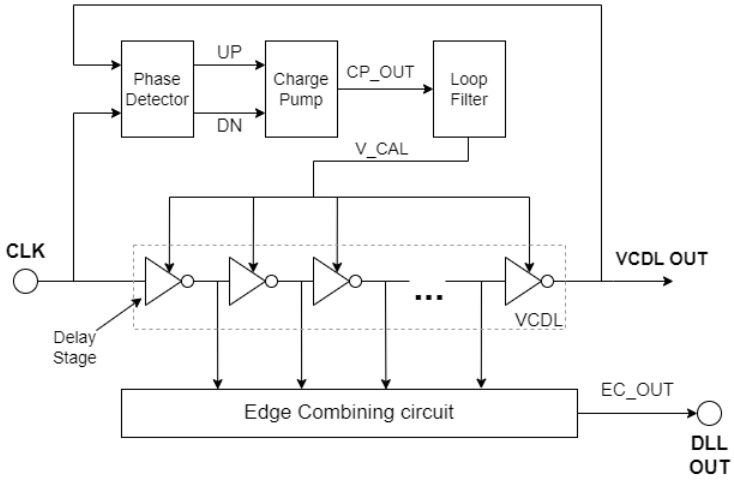

Figure 2.4: Typical EC DLL-based Frequency-Synthesizer architecture

does the opposite. The combined effect of the two control signals is determined by which signal has the longest pulse width.

The two PD outputs are used to control a set of current sources in the CP circuit, where the pulse width of each signal determine the operation time of each current source. This difference in operation time translates into a higher, lower or constant calibration voltage "CP_OUT" . This calibration voltage is then filtered in the Loop Filter (LP) block and distributed as the "V_{cal}" signal to all of the Delay Gate stages in the VCDL line, regulating the amount of phase offset added by all Delay Stages.

When the output of the VCDL and the reference signal become in-phase, the DLL is considered to enter "lock state", where all of the control signals remain constant and the delay added by the VCDL stages is stable. At that point, the phase offseted outputs of each Delay Stage are combined in the EC Circuit by overlapping consecutive signals into one output, usually by using a combinatorial logic circuit. The EC can be done in multiple ways, depending also on the design of the VCDL block. However, one of the most common EC circuit is composed of a system of series and parallel XOR gates.

In an optimal case, the delay added by each Delay Gate stage should be locked to a $delay = \frac{T_{ref}}{N}$, where T_{ref} is the reference signal period and N is the number of Delay Gate stages used in the VCDL. However, this can change depending on possible post-manufacturing mismatches appearing in multiple blocks of the synthesizer circuit, the effect of which will be discussed in a later subsection.

The entire feedback flow in the previous paragraphs can be visualised in the flowchart presented in Figure 2.5.

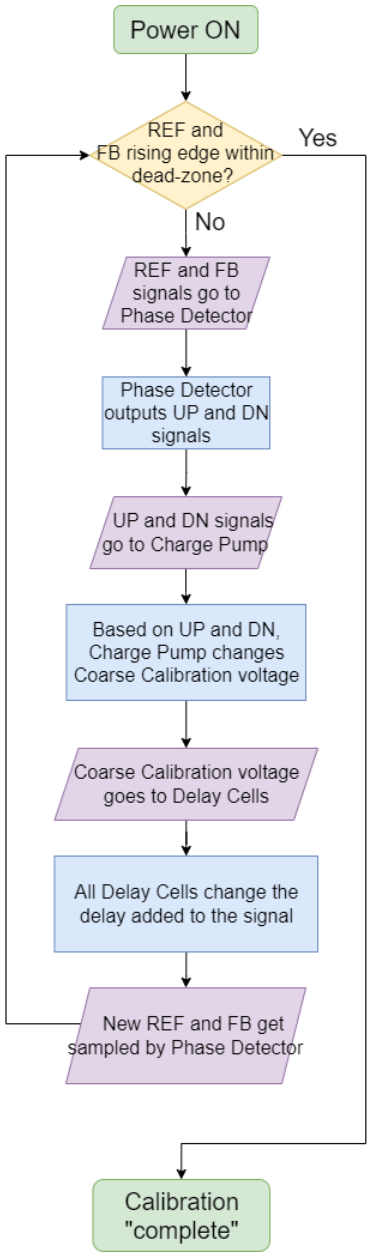

Figure 2.5: Flowchart of DLL feedback

Depending on the purpose and scope of the application, the DLL block can be used as a:

- Frequency Multiplier, by combining the intermediary signals generated by the Delay Gates in the VCDL;
- Clock Recovery circuit, by oversampling a reference signal using multiple out-of-phase clock signals;
- Phase Shifting circuit, by employing multiple phases which are generated in the VCDL block as clock signals for other components of an ASIC;

In this thesis, an EC DLL circuit is employed as a Frequency Multiplier, which generates an output at triple the frequency of the reference clock signal for a $5GHz$ Wi-Fi device. The reference clock signal's frequency is $f_{ref} = 3.33GHz$ and the output's frequency is $f_{out} = 9.99GHz$. The resulting clock signal will be used as the Local Oscillator (LO) element of the mixer block of a Wi-Fi receiver device. When used for this purpose, spurious tones in the spectrum of the LO signal should be minimised, otherwise they can lead to down-conversion of interferers over the carrier in the Intermediate Frequency (IF) or Base Band (BB) spectrum.

Sources of spurious tones in EC DLLs

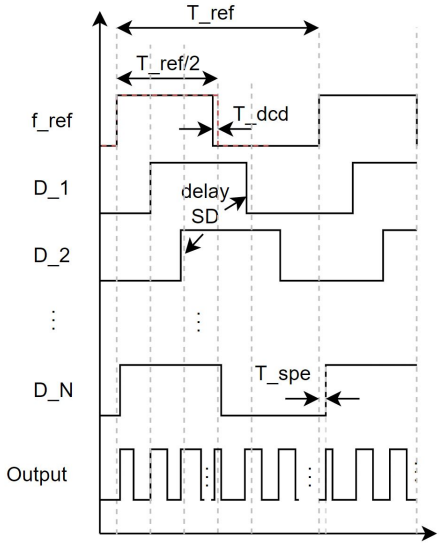

Figure 2.6: Spur sources in a DLL circuit

According to the definition given in [13], a spurious tone is an "Emission on a frequency, or frequencies, which are outside the necessary bandwidth and the level of which may be reduced without affecting the corresponding transmission of information. Spurious emissions include harmonic emissions, parasitic emissions, intermodulation products, and frequency conversion products but exclude out-of-band emissions"

As per the analysis done in [14], there are three possible sources of signal distortions that can lead to the generation of spurious tones in the frequency spectrum of the output signal:

1. Duty Cycle Distortion (DCD), which are mismatches in the duty cycle of the reference clock signal;
2. Static Phase Error (SPE), generated by mismatches in the PD and CP blocks of the Frequency-Synthesizer;
3. Delay Standard Deviation (delay SD), which represents mismatches between individual phase offsets added by each Delay Stage;

Each source of imperfections mentioned above can be viewed in Figure 2.6.

The first type of spur source is the DCD which can appear in the waveform of the reference clock signal as a deviation of the clock's falling edge from the intended timing. This results in a slight deviation of the reference from the intended duty cycle (for example, having a duty cycle of 49,8% instead of 50%), which leads to modifications in the pulse widths of intermediary outputs produced in the VCDL block (ϕ_{1-N}). These modifications or skews in the final edge combined output of the synthesizer is responsible for the generation of unwanted harmonic components in the spectrum of the DLL output, observable in the Fourier series approximation of the DLL output signal (which will be described in more detail in Section 2.3).

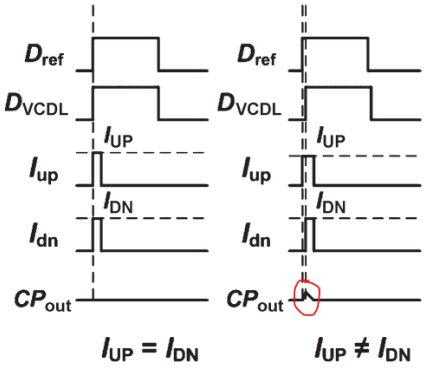

Figure 2.7: SPE effect over VCDL output

The Static Phase Error or SPE represents an improper phase offset in all Delay Stages, generated by improper calibration voltages from V_{cal}. This error is caused by mismatches in the PD block, resulting in out-of-sync UP and DN signals (observed in Figure 2.7) or from imbalances between the pull-up and the pull-down networks in the CP component. As the CP directly controls the amplitude of V_{cal}, this imperfection cascades further to the entire VCDL block.

The accumulated delay errors lead to a VCDL output that is out-of-phase with the reference signal during the locked state, where the error is represented by the ϕ_N phase error parameter. This phase error leads to unbalanced pulse widths in the combined output of the Frequency-Synthesizer, which are represented as undesired harmonics in the frequency domain.

The last source of spurs generation is the delay Standard Deviation (delay SD) of each Delay Stage, represented by small skews in the phase offset of each Delay Stage, during lock state. These misalignments are generated by mismatches produced in the Delay Stages during manufacturing, resulting in different phase offsets D_{1-N} for the same V_{cal} voltage, in comparison with the ideal average delay μ. This results in skewed clock edges in the final combined output of the component, and thus, in spurious tones in the output. The magnitude and the frequencies of the spurs are also affected by the number of Delay Stages used in the DLL circuit, where more stages result in more undesired harmonics in the frequency domain, as well as spurious tones closer to the carrier's frequency.

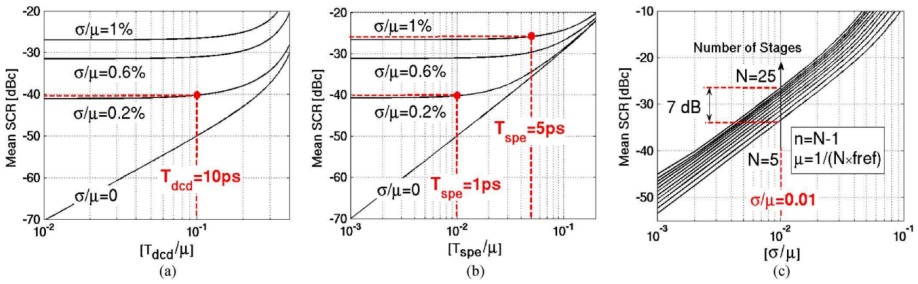

Figure 2.8: Dependency between total Spur-to-Carrier Ratio and individual spur sources, referenced from [14]

According to [14], all three factors affect the total magnitude, frequency and number of spurious tones generated by the EC DLL, reducing any one factor provides diminishing returns without improving the magnitude of the other two factors. In order to properly ascertain the effect of the spurious tones in the spectrum of the synthesizer's output, a new parameter, denoted as the Spur-to-Carrier Ratio, will be employed for the rest of this project. All of the mathematical support for deriving the Spur-to-Carrier Ratio (SCR) formula for an EC DLL-based circuit will be explored in Section 2.3 of this thesis.

Thus, the best Spur-to-Carrier Ratio can only be obtained through improvements in multiple spur-generating imperfections of the circuit. This conclusion is reflected in the formula derived by [14] for approximating the mean SCR value which includes effects from all design imperfections mentioned above.

2.3 A statistical approach for estimating SCR in EC DLL circuits

All of the calculations in this section of the thesis are based on the analytical model derived in [14]. However, a summarised derivation of the formulas will be included to facilitate the understanding of the mathematical basis constructed upon in Chapter 4.

This thesis describes the design of an on-chip calibration system in a typical EC DLL-based Frequency-Synthesizer. As mentioned in Subsection 2.2.1, spurious

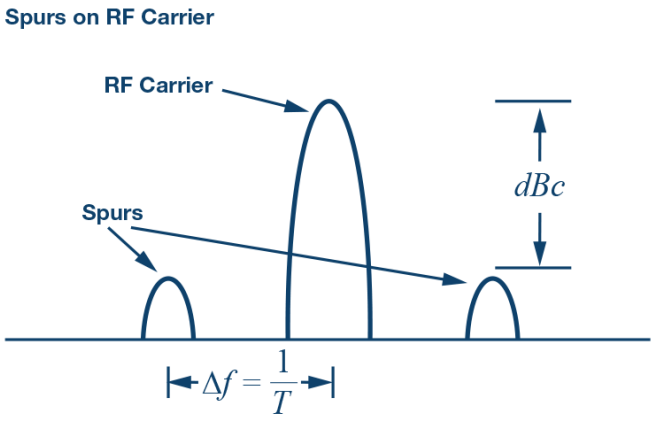

Figure 2.9: Spur-to-Carrier Ratio representation

tones in this type of Frequency-Synthesizer architecture (which are harmonics of the reference signal) are affected by mismatches from multiple sub-blocks of the circuit, and the magnitude of the resulting frequency components has a dependency to all three sources of imperfections. This relationship makes it difficult to identify the effect of correcting one source of spurs over the LO signal's amplitude.

As such, a unified parameter that can correctly characterise the effect of all of these sources on the Synthesizer's spurious performance needs to be formulated. According to [14], this parameter is the Spur-to-Carrier Ratio. SCR is a way of measuring the magnitude of the spurious tones in regard to the carrier's magnitude, measured in *dBc*. This parameter promotes a facile method of measuring improvements in the Synthesizer's spurious performance after employing different calibration techniques and helps to correctly identify hard limits for spur-generating imperfections in the Edge-Combining Delay-Locked Loop (EC DLL) based circuit.

The Spur-to-Carrier Ratio is a parameter that defines the difference in magnitude between a spurious tone and the carrier signal, defined in *dBc* and observable in figure 2.9. In order to derive a formula for the Spur-to-Carrier Ratio of the EC DLL circuit employed in this thesis, a model needs to be defined first for the edge-combined output of the circuit, as well as the influence of each spur-generating parameter, mentioned in the previous subsection.

As a start, SCR can be defined as an individual parameter for each spurious tone in the output spectrum of the synthesizer, in the following way:

$$SCR_n = \frac{S_n}{C} \tag{2.3}$$

Where S_n represents n^{th} harmonic in the EC DLL output spectrum, which represents a spurious tone for the mixing stage, and C represents the 10 GHz carrier signal, identified for $n = N$.

In the case of the circuit used in this thesis project, the input clock's frequency is $f_{ref} = 3.33GHz$ and the number of stages used in the VCDL sub-block is $N = 3$. Thus, the output signal f_{out} appears at a frequency of $f_{out} = f_{carrier} =$

$f_{ref} * N = 10GHz$ and the spurious tones occur in the frequency spectrum at $S_n = n * f_{ref}$, where n is any integer multiplier of the input frequency, except for $n = N$. For this project, the spurious tones of most interest are going to be the second-order harmonic $f_{spur1} = f_{carrier} - f_{ref}$ and the fourth order harmonic $f_{spur2} = f_{carrier} + f_{ref}$, which appear for $n = 2$ and $n = 4$. (which will be elaborated more in Chapter 3)

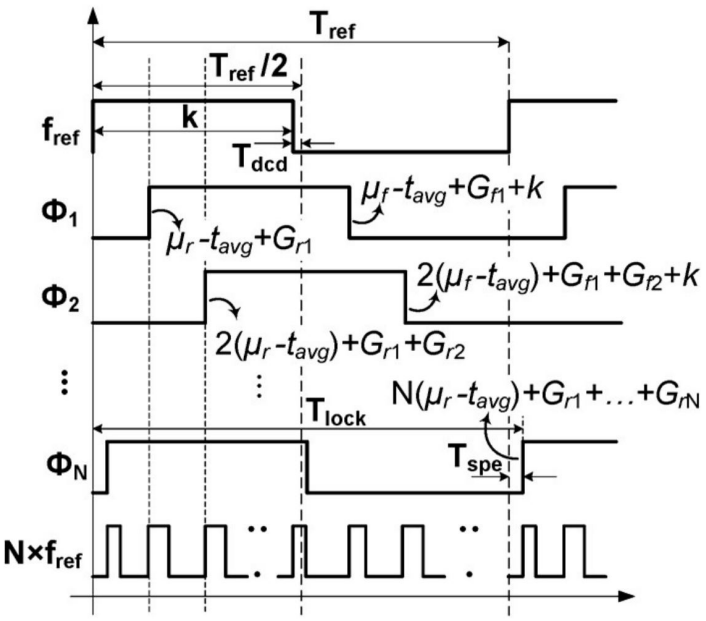

Figure 2.10: Spur sources in a DLL circuit, referenced from [14]

In order to define the magnitude of the different spurious tones or that of the carrier, an expression needs to be formulated for the combined output of the Frequency-Synthesizer. Each of the $m = 1...N$ individual outputs of the VCDL block of the circuit, which are later combined to obtain the synthesised output, can be expressed through a Fourier series, composed of the individual sinusoidal components of the signal:

$$\phi_m = \sum_{n=1}^{\infty} a_{n,m} * cos(n\omega_{ref}t) + \sum_{n=1}^{\infty} b_{n,m} * sin(n\omega_{ref}t) \tag{2.4}$$

Where $a_{n,m}$ represents the coefficients of the cosinusoidal components of the signal, while $b_{n,m}$ represents the coefficients of the sinusoidal components of the signal.

$$a_{n,m} = \frac{2A_o}{T_{ref}} \int_{m\mu+\sum_{i=1}^{m} G_{ri}-mt_{avg}}^{k+m\mu+\sum_{i=1}^{m} G_{fi}-mt_{avg}} cos(n\omega_{ref}t)dt \tag{2.5}$$

$$b_{n,m} = \frac{2A_o}{T_{ref}} \int_{m\mu+\sum_{i=1}^{m} G_{ri}-mt_{avg}}^{k+m\mu+\sum_{i=1}^{m} G_{fi}-mt_{avg}} sin(n\omega_{ref}t)dt \tag{2.6}$$

In the expressions above, it can be noticed that each of the coefficients derived from the Fourier series seems to be affected by a lot of complicated parameters. A few of them represent the effect of the three different sources of spurs mentioned previously and are represented in Figure 2.10.

The effect of delay Standard Deviation can be defined through a statistical model of the possible mismatch effects in a Delay Stage of the VCDL. If we consider that the delay of each stage is a random variable of normal distribution, then the effect of mismatches over the rising and falling edges of each stage's output can be modeled as two independent Gaussian random variables: $\Delta t_{d,r} \approx N(\mu_r, \sigma_r^2)$ and $\Delta t_{d,f} \approx N(\mu_f, \sigma_f^2)$. However, an assumption can be made that the mean of the rising and falling edge delays are equal, meaning $\mu_r = \mu_f = \mu = \frac{T_{ref}+T_{SPE}}{N}$, which should be equal to the "ideal delay" μ added by an ideal Delay Gate during lock state.

Furthermore, that means that instead of representing the Gaussian distribution of all rising and falling edges, a new set of normally-distributed variables can be used to represent the rising and falling edges' skews per Delay Cell. Respectively, $G_{r1}, G_{r2}...G_{rN} \approx N(0, \sigma_r^2)$ represent the zero-mean Gaussian variables for skews in each of the $m = 1...N$ rising edge, respectively $G_{f1}, G_{f2}...G_{fN} \approx N(0, \sigma_f^2)$ for the falling edge mismatches.

However, the fact that the DLL regulates a V_{cal} voltage based on the rising edges of the reference and the VCDL output means that there is an averaging effect applied to the offsets of all Delay Gates, also averaging the effect of their individual mismatches. This can be modeled by a Gaussian distributed variable t_{avg}, which shows the average delay added by each Delay Gate in the circuit:

$$t_{avg} = \frac{1}{N} \sum_{m=1}^{N} G_{rm} \approx N(0, \frac{\sigma_r^2}{N}) \tag{2.7}$$

All of these mismatches also affect the locking time period for the DLL, which is the time period for a reference pulse to be output from the VCDL block. The DLL used in this project is a rising-edge-locked system, and the total lock time will also be affected by the effect of the SPE. As SPE is a source of error that is unrelated to the VCDL block, it cannot be compensated or averaged by the DLL feedback loop. Thus, T_{lock} can be expressed as:

$$T_{lock} = \sum_{m=1}^{N} \Delta t_{d,rm} = T_{ref} + T_{SPE} \tag{2.8}$$

Last, the effect of the DCD can be observed in every output signal from the VCDL Delay Gates, as the mismatch in the input clock signal's duty cycle becomes replicated in every Delay Cell. This modifies the pulse width of each ϕ_m signal from $\frac{T_{ref}}{2}$ to a new value, denoted as "k":

$$k = \frac{T_{ref}}{2} + T_{dcd} \tag{2.9}$$

Now that all of the effects of the sources of mismatches in an EC DLL circuit have been modeled, a representation of the combined synthesizer output can be expressed as $\phi_{out}(t)$:

$$\phi_{out}(t) = \sum_{m=1}^{N} \phi_m(t) \quad = \sum_{n=1}^{\infty}(cos(n\omega_{ref}t) \sum_{m=1}^{N} a_{n,m}) + \sum_{n=1}^{\infty}(sin(n\omega_{ref}t) \sum_{m=1}^{N} b_{n,m}) \tag{2.10}$$

Where, through linearity, the cosinusoidal coefficients of the equation can be summed into $a_n = \sum_{m=1}^{N} a_{n,m}$ and the sinusoidal coefficients can be summed into $b_n = \sum_{m=1}^{N} b_{n,m}$.

As these summed coefficient terms represent the magnitude of the Fourier series components of the synthesizer's output, they can be used as vector components (a_n, b_n) to calculate the magnitude of any frequency of interest in its spectrum. As such, an expression can be derived for any harmonics of the EC output, including that of the carrier:

$$S_n = |a_n + jb_n| = \sqrt{a_n^2 + b_n^2}, \quad n \neq N \tag{2.11}$$

$$C = |a_N + jb_N| = \sqrt{a_N^2 + b_N^2} \tag{2.12}$$

Now, substituting 2.11 and 2.12 in equation 2.3, the following expression for the SCR parameter is obtained:

$$SCR_n = \frac{\sqrt{a_n^2 + b_n^2}}{\sqrt{a_N^2 + b_N^2}}, \quad n \neq N \tag{2.13}$$

However, to better define the actual value of the SCR parameter for any specific magnitude for the spur-generating imperfections, a more expansive formula for the a_n and b_n coefficients needs to be derived. Such a formula can be obtained by expanding the sum inside of each Fourier coefficient as follows:

$$a_n = \frac{2A_0}{n\pi} \sum_{m=1}^{N} [\sin \frac{n\pi}{T_{ref}}(k+X_{fm}-X_{rm}) * \cos \frac{n\pi}{T_{ref}}(\frac{2m(T_{ref}+T_{spe})}{N}+k+X_{fm}+X_{rm})] \tag{2.14}$$

$$b_n = \frac{2A_0}{n\pi} \sum_{m=1}^{N} [\sin \frac{n\pi}{T_{ref}}(k+X_{fm}-X_{rm}) * \sin \frac{n\pi}{T_{ref}}(\frac{2m(T_{ref}+T_{spe})}{N}+k+X_{fm}+X_{rm})] \tag{2.15}$$

Where X_{fm} and X_{rm} represent symbols used for ease of understanding of the previous two equations and which are equal to:

$$X_{fm} = -\frac{m}{N}\sum_{i=1}^{N} G_{ri} + \sum_{i=1}^{m} G_{fi}$$

$$X_{rm} = -\frac{m}{N}\sum_{i=1}^{N} G_{ri} + \sum_{i=1}^{m} G_{ri}$$

As a direct effect of the expanded versions of the Fourier coefficients in equations 2.14 and 2.15, it can be observed that a_n and b_n are a convolution of simple Gaussian functions, implying that they can be approximated to two independently and identically distributed Gaussian random variables of zero mean and equal variances σ^2_{cof}:

$$a_n \approx N(0, \sigma^2_{cof}),\ b_n \approx N(0, \sigma^2_{cof}) \tag{2.16}$$

Where, in order to evaluate the Fourier coefficients of the output spur as a function of delay SD, the variance formula of a Gaussian variable can be utilised. The formula is represented, as a reminder, in equation 2.17:

$$var[X] = E[X^2] - (E[X])^2 \tag{2.17}$$

Where $E[X]$ is the expectation operator and $E[X]^2$ is the second moment of a Gaussian variable.

This is a useful conclusion, as it enables the approximation of the formula of spurious tones S_n with a Rayleigh random variable with a Probability Density Function (PDF) $p_x(x)$ and mean value $E[S_n]$, that are derived respectively as follows:

$$p_X(x) = \begin{cases} (\frac{x}{\sigma^2_{cof}})exp(\frac{-x^2}{2\sigma^2_{cof}}), & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{2.18}$$

$$E[S_n] = \sigma_{cof}\sqrt{\frac{\pi}{2}} \tag{2.19}$$

Next, by utilizing the first and second-order Taylor series approximation of sinusoidal and cosinusoidal Fourier coefficients, expressions can be derived for the mean statistical value of a_n and b_n:

$$E[a_n] = A_n \sum_{m=1}^{N} \alpha_m K_s - \alpha_m K_s \lambda^2 m(\sigma_f^2 + \sigma_r^2) - \beta_m K_s \lambda^2 m(\sigma_f^2 + \sigma_r^2(\frac{2m}{N} - 1)) \tag{2.20}$$

$$E[b_n] = A_n \sum_{m=1}^{N} \beta_m K_s - \beta_m K_s \lambda^2 m(\sigma_f^2 + \sigma_r^2) + \alpha_m K_s \lambda^2 m(\sigma_f^2 + \sigma_r^2(\frac{2m}{N} - 1)) \tag{2.21}$$

where

$$\alpha_m = \cos(\frac{n\pi}{T_{ref}}[\frac{2m}{N}(T_{ref} + T_{spe}) + k]) \tag{2.22}$$

and

$$\beta_m = \sin(\frac{n\pi}{T_{ref}}[\frac{2m}{N}(T_{ref} + T_{spe}) + k]) \tag{2.23}$$

Then, after deriving the variance expressions var$[a_n]$ and var$[b_n]$using 2.17, the value of σ^2_{cof} can be approximated by the averaged sum of var$[a_n]$ and var$[b_n]$ as:

$$\sigma^2_{cof} = \frac{var[a_n] + var[b_n]}{2}$$

$$\sigma^2_{cof} = (\frac{2A_0}{T_{ref}})^2(\sigma^2_f + \sigma^2_r) * (\frac{N(N+1)}{4} + \sum_{m=1}^{N-1} m(\alpha_m \sum_{m+1}^{N} \alpha_i + \beta_m \sum_{i=m+1}^{N} \beta_i)) \tag{2.24}$$

Using these new expressions, a formula for the mean value of SCR could be derived as long as SCR could be approximated with a Rayleigh variable or similar. Thankfully, this can be done through a series of approximations, starting with replacing carrier with its mean and re-write the equation 2.3 as:

$$SCR_n = \frac{S_n}{E[C]} \tag{2.25}$$

We then approximate and then simplify E[C] as follows:

$$E[C] = E[\sqrt{a^2_N + b^2_N}] \approx \sqrt{(E[a_N])^2 + (E[b_N])^2}$$

$$E[C] \approx \frac{2}{N\pi} |\frac{\sin(\frac{kN\pi}{T_{ref}}) \sin(\frac{T_{spe}N\pi}{T_{ref}})}{\sin(\frac{T_{spe}\pi}{T_{ref}})}| \tag{2.26}$$

As such, since SCR is now the ratio between a Rayleigh variable S_n and a constant, the linearity property can be used to also approximate SCR as a Rayleigh variable with the following PDF $p_N(x)$ and mean value $E[SCR_n]$:

$$p_N(x) =, \begin{cases} (\frac{x}{\sigma^2_R}) exp(\frac{-x^2}{2\sigma^2_{cof}}), & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{2.27}$$

$$E[SCR_n] = \sigma_R \sqrt{\frac{\pi}{2}} \tag{2.28}$$

where

$$\sigma_R = \frac{\sigma_{cof}}{E[C]} \tag{2.29}$$

Lastly, according to [15], the pdf representation of the SCR variable in equation 2.27 can be transformed into a Logarithmic representation through a change of variables, resulting in the following Log-Rayleigh PDF:

$$p_Y(y) = (\frac{10^{\frac{y}{10}}}{\sigma_R^2})exp(\frac{-10^{\frac{y}{10}}}{2\sigma_R^2}), \tag{2.30}$$

2.4 Spur-Aware Synthesizer Design Flow

Based on the details presented in Section 2.3, a new methodology of designing EC DLL circuits is suggested in [14], which has been utilized during the design of this thesis' Frequency Synthesizer block. The design flow is based on the following calculations: For $T_{spe} \approx T_{dcd} \approx 0$, it can be derived that:

$$var[a_n]_{min} = \frac{N(\sigma_f^2 + \sigma_r^2)[2 - \cos(\frac{2n\pi}{N})]}{2T_{ref}^2 \sin^2(\frac{n\pi}{N})} \tag{2.31}$$

$$var[b_n]_{min} = \frac{N(\sigma_f^2 + \sigma_r^2)[2 + \cos(\frac{2n\pi}{N})]}{2T_{ref}^2 \sin^2(\frac{n\pi}{N})} \tag{2.32}$$

$$\sigma_{cof.min}^2 \approx \frac{var[a_n]_{min} + var[b_n]_{min}}{2} = \frac{N(\sigma_f^2 + \sigma_r^2)}{T_{ref}^2 \sin^2(\frac{n\pi}{N})} \tag{2.33}$$

The mean magnitude of the carrier(n = N) in equation (2.13) can be simplified as follows:

$$E[C] = \frac{2}{\pi} \tag{2.34}$$

By substituting equations (2.20) and (2.21) into (2.15) and (2.16) respectively, the lower bound on achievable mean SCR is determined as:

$$E[SCR_n] \geq \frac{\pi\sqrt{\pi}}{2\sqrt{2N}sin(\frac{n\pi}{N})} * \frac{\sqrt{\sigma_f^2 + \sigma_r^2}}{\mu} \tag{2.35}$$

The above expression indicates the best achievable spur suppression for a given delay mismatch and very small SPE and DCD values. Therefore, equation (2.22) can be formulated in a way that defines an upper bound on the normalized delay SD, as follows:

$$\frac{\sqrt{\sigma_f^2 + \sigma_r^2}}{\mu} \leq SCR_{spec} * \frac{2\sqrt{2N}sin|\frac{n\pi}{N}|}{\pi\sqrt{\pi}} \tag{2.36}$$

After determining the design parameters including the reference clock frequency (f_{ref}) and the number of delay stages (N) from the specifications of the wireless standard, the upper bound on the delay SD, SD_{max}, is computed using a given SCR_{spec} limitation, satisfied when SPE and DCD are sufficiently small. To

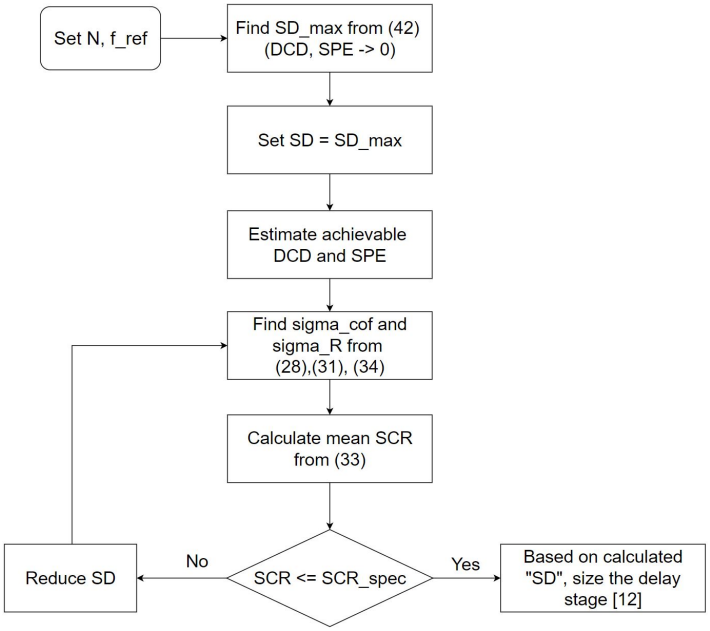

Figure 2.11: Proposed Spur Aware Design Flow, referenced from [14]

consider the effect of non-zero DCD and SPE, the σ_{cof} is calculated using SD_{max} as well as the estimated achievable SPE and DCD values, E[C] from equation 2.26, and σ_R from equation 2.29. The σ_R formula can be substituted in equation 2.25 to find $E[SCR_n]$ in 2.28.

Now, comparing the value of $E[SCR_n]$ with the SCR_{spec} that has been derived, if $E[SCR_n]$ is larger than SCR_{spec}, then the assumed delay SD margin needs to be lowered and the procedure needs to be re-run until $E[SCR_n] \leq SCR_{spec}$. Thus, based on the values of SPE, DCD, and the delay SD was chosen for the specified "N" and "f_{ref}" used in the Frequency Synthesizer, the SD_{max} limit might need to be changed. The value of n also changes with the change in N and, thus, the SD_{max} as well.

Therefore, it is crucial to have all the values of the mismatch effects such as SPE, DCD, and delay SD close to "0%" to maintain the level of the spurious tones within the spur suppression requirement.

In Figure 2.12, the transistor-level Monte Carlo simulation results for the Frequency Multiplier's SCR are compared with the values derived from the analytical model explained in Section 2.3. The dashed curve shows the normalized closed-form SCR PDF from equation 2.30, whilst the bar chart represents the SCR histogram obtained from Monte-Carlo simulations of a physical EC DLL circuit.

Large variations in SPE, DCD, and, delay SD are taken in computing the $E[SCR_n]$ to evaluate the accuracy of the proposed Rayleigh approximated closed-form formula.

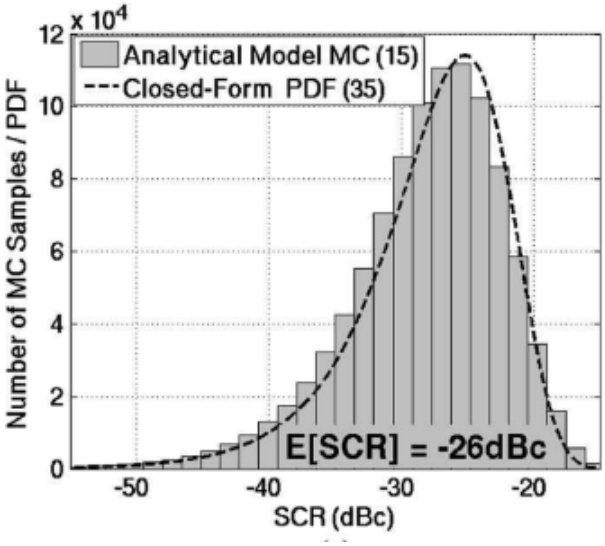

Figure 2.12: Mean Spur-to-Carrier Ratio, referenced from [14]

In this case, since we have specifications of $f_{ref} = 3.33GHz$, number of Stages as $N = 3$ (3 pairs of delay stages due to differential delay line design), $f_{out} = f_{carrier} = 10GHz$ means the interferer scenario of interest is represented by out-of-band interferers falling in the $0.8GHz$ to $2.4GHz$ range. These could coincide with some of the spurious tones generated by the Frequency Synthesizer block, which need to be within the derived specifications limit to not affect the information in the carrier signal. More details about how this affects the specifications needed for the calibration system performance improvement will be explored in more detail in Chapter 3.

2.5 Properties of an invertible matrix

Some of the properties of the inverse matrix according to [16] are as follows:

1. An n-by-n matrix square matrix A is called invertible if there exists an n-by-n square matrix B such that

$$AB = BA = I_n \tag{2.37}$$

   where In denotes the n-by-n identity matrix and the multiplication used is ordinary matrix multiplication.

2. A square matrix that is not invertible is called singular or degenerate. A square matrix with entries in a field is singular if and only if its determinant is zero and a square matrix with entries is invertible if and only if its determinant is non-zero.

3. The transpose A^T is an invertible matrix.
4. The columns of A are linearly independent.
5. The rows of A are linearly independent.

Chapter 3

Wi-Fi standard and derivation of Specifications

3.1 Deriving the specifications

According to [17] as far as deriving the specifications for the required SCR is concerned, various types of interference scenarios need be considered during the analysis of the Wireless system to get the best-case SCR and the worst-case SCR. When considering interferer signals where the wanted Wi-Fi signal is the victim, it is important to consider practical application scenarios. It depends on where the interferer may be located with respect to the Wi-Fi Receiver, for example, it may be co-located in the same device, such as a laptop, or located in a different device which is in proximity to a different system transmitting a signal at the same time.

The interferer signals can be broken down further into in-band interferers, e.g. Wi-Fi interferers, out-of-band interferers, e.g. WLAN 802.11 interferers(5.8 GHz), cellular interferers, e.g. any LTE frequency bands and so on. These types of interferers need to be considered separately, since out-of-band interferers can be rejected more easily by filtering than in-band interferers. However, out-of-band interferers that fall close to the band edge can also be problematic, because of their proximity to the band, thus posing limitations on filtering.

The entire specification derivation is done by taking into account practical, real-time scenarios in regards to the amplitude of received interferer and carrier signals and in reference to the Receiver chain in Figure 1.1 from Chapter 1.

3.1.1 Desired signal "far" from receiver

Consider the scenario where the receiver needs a signal transmitted by a base station which is 500m away, while an interferer source is just 1m away. The interferer signal can be of any form, such as a mobile phone, laptop, etc. For this scenario, the interferer is considered to be a mobile phone transmitting within then Advanced Wireless Services(AWS) frequency band, which is a wireless telecommunications protocol used for mobile voice and data services, video, and messaging.

The AWS band uses microwave frequencies in two segments, from 1710 to 1755 MHz for uplink, and from 2110 to 2155 MHz for downlink. The service is intended

to be used by mobile devices such as wireless phones for mobile voice, data, and messaging services, according to [16].

Carrier and interferer signals

By considering the distance between the transmitter and the receiver(r) to be quite far, the path loss is more at the Antenna Reference Point(ARP) according to the given formula as follows:

$$S(in\ dBm) = \frac{1}{4 * pi * r^2} * \frac{\lambda^2}{4 * pi} * P_{TX} \tag{3.1}$$

In the above equation, P_{TX} represents the transmit power in dBm, λ represents the wavelength of a signal in m for a particular frequency and S represents the transmit power at the ARP of a receiver in dBm. As we see in the equation above, the path loss is represented by the equation below:

$$Path\ Loss(in\ dBm) = 10 * log_{10}(\frac{1}{4 * pi * r^2} * \frac{\lambda^2}{4 * pi}) \tag{3.2}$$

The path loss in linear scale has an inverse relationship with respect to the square of the distance between the transmitter and the receiver(r_w). In this case, we choose values for (P_{TX}), the transmit power of the desired signal, (P_{TX_w}) as +20 dBm, the transmit power of the interferer signal($P_{TX_{int}}$) as +23 dBm frequency of the wanted signal(f_w) = 5.16 GHz, and frequency of the interferer signal(f_{int}) is chosen as one of the uplink frequencies of the AWS band which is = 1.715 GHz from the above discussion. The values for (r_w) and (r_{int}) are chosen as 50m and 1m respectively.

Firstly, using the values of f_w, f_{int} and $c = 3 * 10^8 m/s$, we calculate λ for the wanted and the interferer signals respectively and substitute them in the equation 3.1 to get the values of S(in dBm) for the carrier signal(S_w) and the interferer signal(S_{int}). But, there are some construction materials which yield signal attenuation. In this case, it is taken into consideration that the concrete material yields the attenuation of the wanted signal by 22 dB as specified in [18].

$$S_w = -82.67\ dBm \tag{3.3}$$

$$S_{int} = -14.12\ dBm \tag{3.4}$$

P_{TX_w}	+20 dBm
$P_{TX_{int}}$	+23 dBm
f_w	5.16 GHz
f_{int}	1.715 GHz
r_w	500 m
r_{int}	1 m
S_w	-82.67 dBm
S_{int}	-14.12 dBm

Table 3.1: Values for Far Case Scenario

The frequency spectrum at the input of the Diplexer is shown in Figure 3.1 with the amplitude levels of the carrier signal and the interferer signal in dBm at the respective frequencies.

Since we deal with a scenario where the desired signal is 'far' from the receiver, the BPSK modulation scheme with an MCS Index of '0' can be chosen. The SNR Wi-Fi table below shows the SNR required for each MCS index as well as the modulation, specified in [19]. The sensitivity value of -92 dBm is chosen to be 10 dB lower than the minimum sensitivity(20 MHz channel spacing) for the BPSK modulation scheme as represented by the red dotted line in Figure 3.1.

MCS Index	Modulation	Required SNR(dB)
0	BPSK	5

Table 3.2: SNR Wifi Table for MCS Index '0'

A Diplexer component similar to the specified in [20], the attenuation of the diplexer at the f_{int} is approximately 25 dB. The amplitude of the wanted signal stays the same since there is no attenuation at the f_w whereas the amplitude of the interferer signal reduces by 25 dB to -39.1 dBm as shown in the Figure 3.2 below. However, the amplitude of the interferer signal would get attenuated even

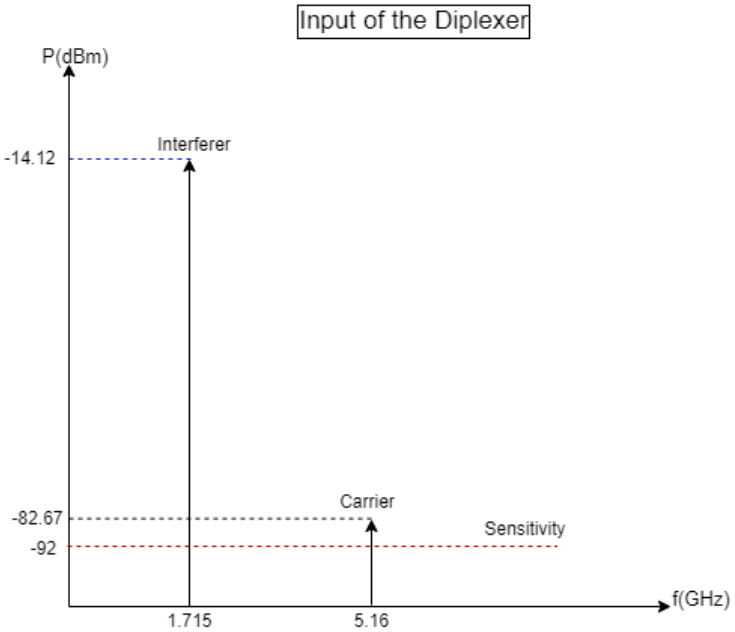

Figure 3.1: Input of the Diplexer in Far Case Scenario

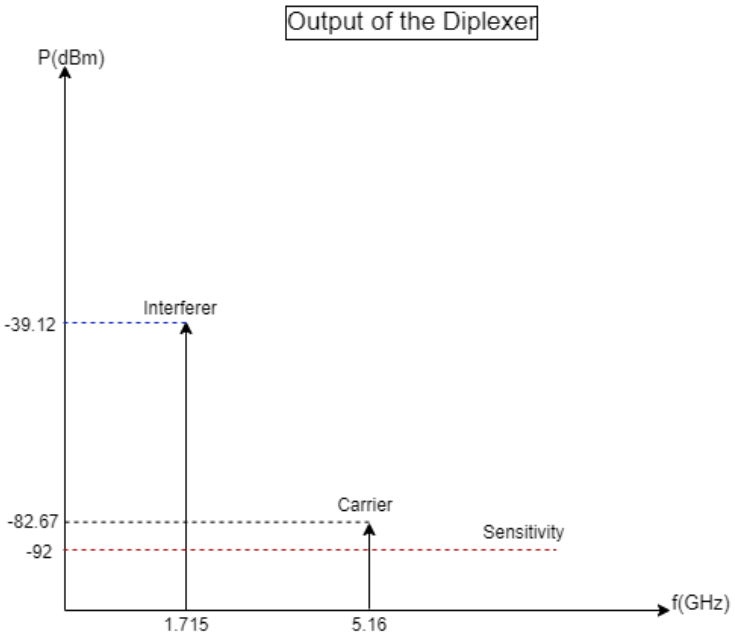

Figure 3.2: Output of the Diplexer in Far Case Scenario

more for higher values of attenuation than the value of 25 dB.

A typical LNA forms the next block of the receiver chain. Here, a second-order Band Pass Filter is used as the resonant tank circuit of the LNA with certain Gain and attenuation values based on the characteristics of the filter such as the Bandwidth(BW), the order of the filter(n), Quality Factor(Q) and the peak frequency(f_p).

A second-order Bandpass Filter is used in the design whose polynomial transfer function can be expressed as follows:

$$H(s) = \frac{\frac{w_m}{Q}s}{s^2 + \frac{w_m}{Q}s + w_m^2} \tag{3.5}$$

where w_m is the peak angular frequency.

Q	15
n	2
f_p	5.16 GHz
f_{int}	1.715 GHz

Table 3.3: Specifications for the Band-Pass Filter

Using the values from the Table 3.3, the bode-plot(Magnitude(in dB) vs Frequency(in GHz)) of the band-pass filter is plotted as shown in the Figure 3.3 below.

From the Figure 3.3, it is interesting to note that the interferer signal at f_{int}(1.715 GHz) undergoes attenuation by 32 dB.

The amplitude levels at the input of the LNA is translated from power(dBm) unit to voltage(dB) unit, which in turn is represented for both the signals, the interferer signal amplitude and the wanted signal's amplitude in dB at the input of the LNA as -52.13 dB and -93.68 dB respectively. From the Figure 3.4 below, we see that the interferer signal at the output of the LNA has a voltage level of -84.13 dB since it undergoes an attenuation of 32 dB whereas the wanted signal stays at the same voltage level in dB since the signal does not attenuate.

At the output of the LNA,

$$V_w = -95.68dB + A \tag{3.6}$$

$$V_{int} = -52.13dB - 32dB + A => -84.13dB + A \tag{3.7}$$

where A represents the LNA gain. But for the derivation of the specifications, the LNA gain is neglected.

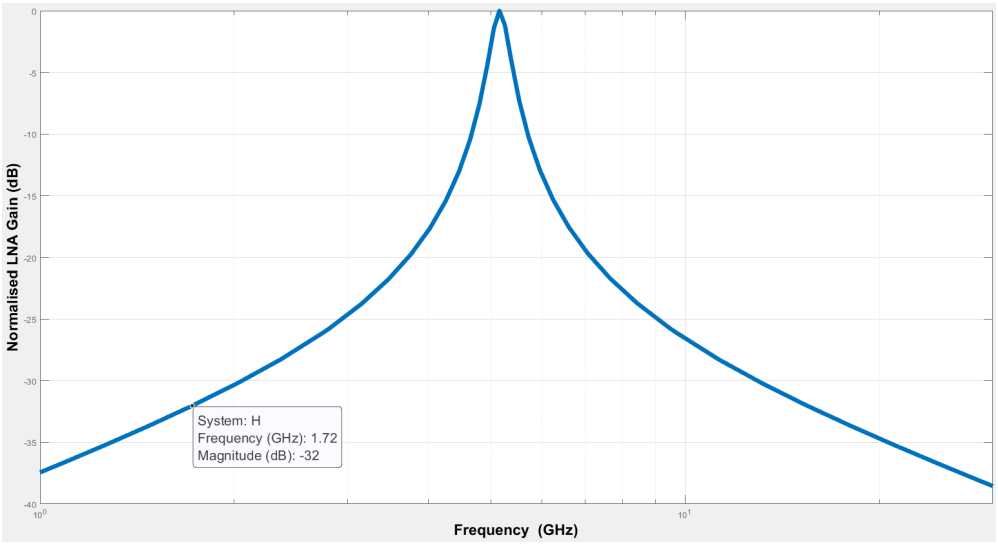

Figure 3.3: Frequency Response Curve of the Band-Pass Filter

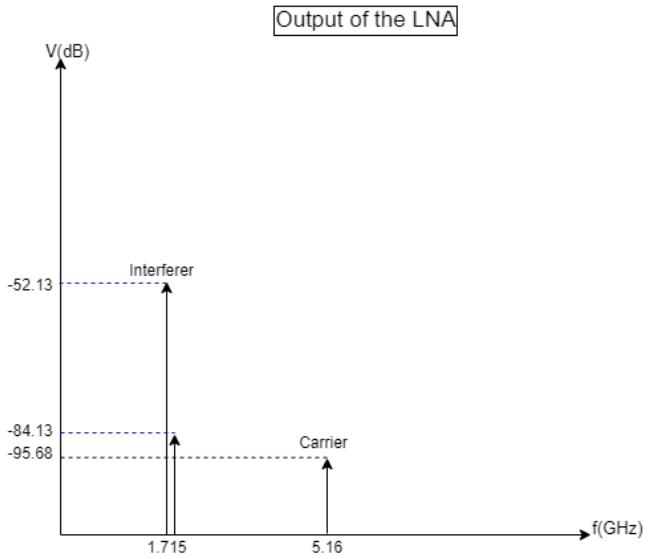

Figure 3.4: Output of the LNA in Far Case Scenario

The Frequency Tripler block forms the next block in the Receiver chain after the LO block. The Frequency Tripler produces the carrier signal at $f_c = 3 * f_{fundamental}$, and harmonics of the fundamental signal as spurious tones at a frequency offset of the fundamental frequency from the carrier signal, i.e., $f_{spur1} = f_c - f_{fundamental}$ and $f_{spur2} = f_c - f_{fundamental}$ as shown in the Figure 3.5 below. The voltage level of the spurious tones is represented as X(in dB) since its value has to be computed in order to estimate the required SCR(in dBc) for the entire system. The frequency offset between the carrier signal(f_c) and the spurious signals(f_{spur1} and f_{spur2}) is equal to the $f_{fundamental}$ = 3.44 GHz.

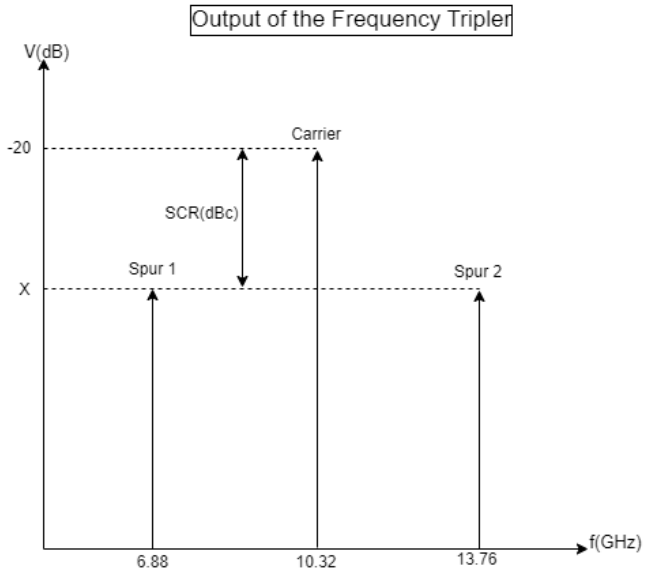

Figure 3.5: Output of the Frequency Multiplier

The Divider-by-2 block divides the carrier as well as the harmonic frequencies by a factor of 2, by maintaining the same frequency offset of 3.44 GHz between the carrier signal and the spurious signals. However, the level of the spurious tones reduces by 6 dB at the output of the Divider-by-2 block whereas the carrier voltage($V_{c,LO}$) stays at the same level of -20 dB.

The mixer basically does frequency multiplication of the RF and the LO signals to produce frequencies at the IF.However, the voltage levels of the signals at the output of the Mixer are written in the form of equations below as follows:

At the output of the Mixer,

$$V_{c,Mixer} = V_{c,LO} + V_w + G_{conv} => -20 - 95.68 + G_{conv} \tag{3.8}$$

$$V_{int,Mixer} = (X - 6)dB + V_{int} + G_{conv} => X - 6 - 84.13 + G_{conv} \tag{3.9}$$

where G_{conv} is the conversion gain of the Mixer.

$$V_{c,Mixer} = -115.68 + G_{conv} \tag{3.10}$$

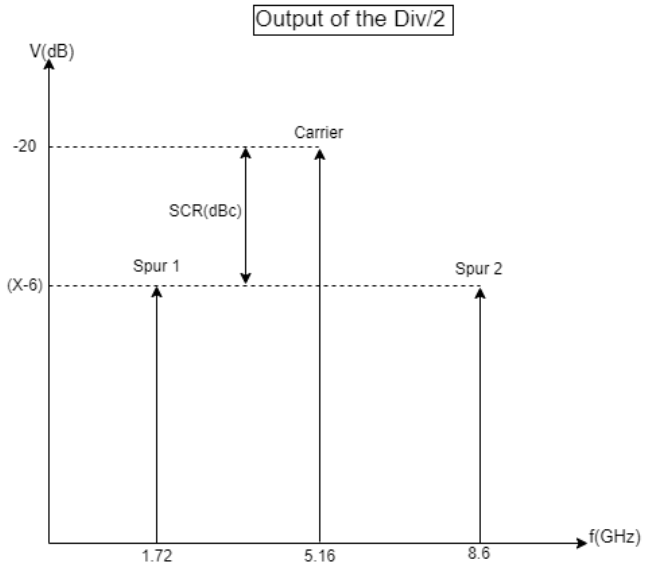

Figure 3.6: Output of the Div/2 Block

$$V_{int,Mixer} = X - 90.13 + G_{conv} \tag{3.11}$$

$$SNR = V_{c,Mixer} - V_{int,Mixer} \tag{3.12}$$

By substituting the values of $V_{c,Mixer}$ and $V_{intMixer}$ in equation 3.13, we get:

$$5dB = -115.68 + G_{conv} - (X - 90.13 + G_{conv}) => -25.55 - X \tag{3.13}$$

$$X = -30.55 \tag{3.14}$$

The Spur-to-Carrier Ratio(SCR) can be expressed as follows:

$$SCR(dBc) = \frac{Spur}{Carrier} => Spur - Carrier \tag{3.15}$$

By substituting the values of $V_{c,LO}$ for the Carrier Voltage(dB) and the value of X(dB) for the Spurs in 3.16, we get:

$$SCR(dBc) = X - (-20) => -30.55 + 20 \tag{3.16}$$

$$SCR(dBc) = -10.55dBc \tag{3.17}$$

P_{TX_w}	+20 dBm
$P_{TX_{int}}$	+23 dBm
f_w	5.16 GHz
f_{int}	1.715 GHz
r_w	2 m
r_{int}	1 m
S_w	-32.744 dBm
S_{int}	-14.12 dBm

Table 3.4: Values for Near Case Scenario

3.1.2 Desired signal "close" to receiver

Consider the scenario when the receiver is just 2 metres away from the base station, which in this case is the transmitter whereas the interferer just 1 metre away from the receiver. The interferer signal is considered to be the same as previous scenario.

Carrier and interferer signals:

By considering the distance between the transmitter and the receiver(r_w) to be really close, the path loss is comparatively lesser at the ARP in this scenario according to the formula used in 3.1:

In this case, we choose the same values for (P_{TX_w}) and ($P_{TX_{int}}$), (f_w), and (f_{int}) as the previous scenario. The values for (r_w) and (r_{int}) are chosen as 2m and 1m respectively.

The transmit power(in dBm) at the ARP for the carrier signal(S_w) and the interferer signal(S_{int}) are written as follows:

$$S_c = -32.74\,dBm \tag{3.18}$$

$$S_{int} = -14.12\,dBm \tag{3.19}$$

The frequency spectrum at the input of the Diplexer is shown below with the amplitude levels of the carrier signal and the interferer signal in dBm at the respective frequencies.

MCS Index	Modulation	Required SNR(dB)
7	64-QAM	22.5

Table 3.5: SNR Wifi Table for MCS Index '7'

Since we deal with a scenario where the desired signal is 'near' from the receiver, the 64-QAM modulation scheme with an MCS Index of '7' can be chosen. The SNR Wi-Fi table below shows the SNR required for each MCS index as well as the modulation, specified in [19]. The minimum sensitivity(20 MHz channel spacing) for the 64-QAM modulation scheme is represented by the red dotted line in the above graph.

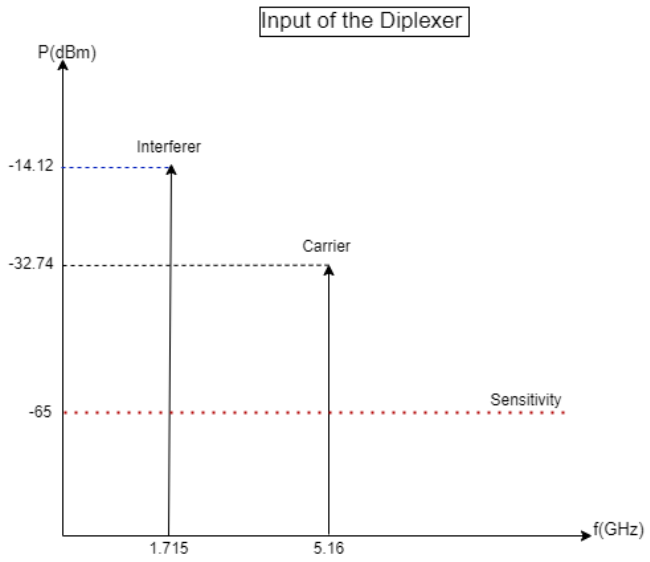

Figure 3.7: Input of the Diplexer in Near Case Scenario

Similar to the previous scenario, the attenuation of the diplexer at the f_{int} is approximately 25 dB. The amplitude level of the wanted signal stays the same since there is no attenuation at the f_w whereas the amplitude level of the interferer signal reduces by 25 dB to -39.1 dBm as shown in the Figure 3.8 below.

The interferer signal at f_{int}(1.715 GHz) undergoes attenuation by -32 dB similar to previous scenario as shown in Figure 3.3.

Since the amplitude levels at the input of the LNA are translated from Power(dBm) unit to Voltage(dB) unit, which in turn is represented for both the signals, the

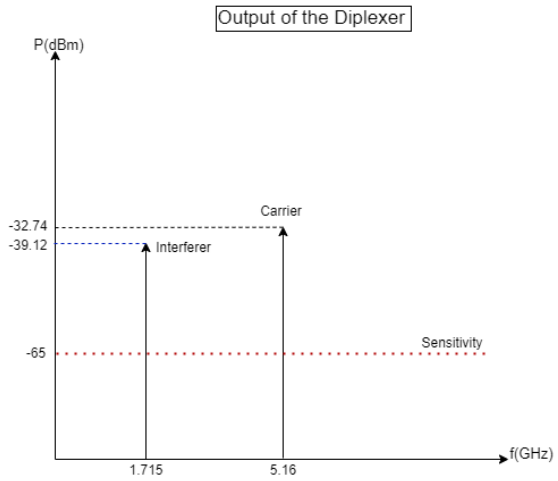

Figure 3.8: Output of the Diplexer in Near Case Scenario

interferer signal amplitude and the wanted signal's amplitude in dB at the input of the LNA as -52.13 dB and -45.75 dB respectively. From Figure 3.9 below, it can be seen that the interferer signal at the output of the LNA has a voltage level of -84.13 dB since it undergoes an attenuation of -32 dB whereas the wanted signal stays at the same voltage level in dB since the signal does not attenuate.

At the output of the LNA,

$$V_w = -45.75 + A \tag{3.20}$$

$$V_{int} = -52.13 - 32 + A => -84.13 + A \tag{3.21}$$

where A represents the LNA gain. But for the derivation of the specifications, the LNA gain is neglected.

As shown in Figure 3.5, the voltage level of the spurious tones are represented as X(in dB) since its value has to be computed in order to estimate the required SCR(in dBc) for the entire system. The frequency offset between the carrier signal(f_c) and the spurious signals(f_{spur1} and f_{spur2}) is equal to the $f_{fundamental}$ = 3.44 GHz.

Similarly, the voltage level in dB of the spurious tones reduces by 6 dB at the output of the Divider-by-2 block whereas the carrier voltage($V_{c,LO}$) stays at the same level of -20 dB. The voltage levels of the signals at the output of the Mixer are written in the form of equations below as follows:

$$V_{c,Mixer} = V_{c,LO} + V_w + G_{conv} => -20 - 45.75 + G_{conv} \tag{3.22}$$

$$V_{int,Mixer} = (X - 6) + V_{int} + G_{conv} => X - 6 - 84.13 + G_{conv} \tag{3.23}$$

where G_{conv} is the conversion gain of the Mixer.

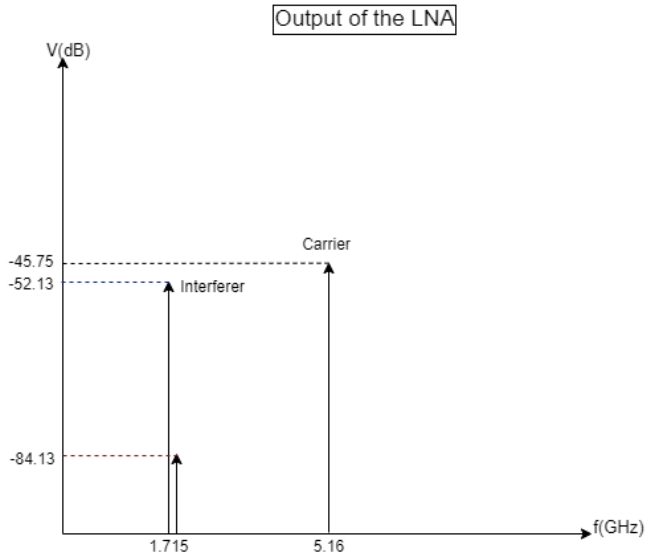

Figure 3.9: Output of the LNA in Near Case Scenario

$$V_{c,Mixer} = -65.75 + G_{conv} \tag{3.24}$$

$$V_{int,Mixer} = X - 90.13 + G_{conv} \tag{3.25}$$

Since the 64-QAM modulation scheme is used for the MCS-Type-9, the SNR for this scenario is given as $22.5dB$ as per [21]. The SNR can be expressed as follows:

$$SNR = V_{c,Mixer} - V_{int,Mixer} \tag{3.26}$$

By substituting the values of $V_{c,Mixer}$ and $V_{intMixer}$ in equation 3.27, we get:

$$22.5dB = -65.75 + G_{conv} - (X - 90.13 + G_{conv}) => 24.38 - X \tag{3.27}$$

$$X = 1.88 \tag{3.28}$$

The Spur-to-Carrier Ratio(SCR) can be expressed as follows:

$$SCR(dBc) = \frac{Spur}{Carrier} => Spur - Carrier \tag{3.29}$$

By substituting the values of $V_{c,LO}$ for the Carrier Voltage(dB) and the value of X(dB) for the Spurs in 3.29, we get:

$$SCR(dBc) = X - (-20) => 1.88 + 20 \tag{3.30}$$

SCR(dBc)	Without Link Budget Margin	With 10 dB Link Margin Budget	With 10 dB Lower Sensitivity
SCR_{Far}	-10.55	-20.55	-30.55
SCR_{Near}	21.88	11.88	1.88

Table 3.6: SCR Values for multiple scenarios

$$SCR(dBc) = 21.88dBc \tag{3.31}$$

Where the Link Margin Budget represents a margin of error, for the specification of all blocks in the Receiver chain, which accounts for cumulative effects from not only interferers, but IM2, IM3 and other noise source. A larger margin of errors ensures the proper operation of the Wi-Fi Receiver even for the worst possible case, however it does result in rather pessimistic values for the specifications of all of the designed receiver blocks. Thus, an observation can be made that even if a block does not meet the specifications derived in this chapter, the final product might still be functional for most use cases.

At the same time, the "10 dB lower sensitivity" section in the Table 3.6 shows what specification values would be needed for the Frequency Synthesizer block in case the sensitivity of the final transceiver device would be lowered in order to achieve typical competitive performance values taken into account the existent commercial solutions available today.

Chapter 4

Self-calibration algorithms

As discussed in the previous chapters, the EC DLL-based Frequency Synthesizer explored in this thesis can be influenced by multiple possible sources of errors, which affect the overall operation of the circuit in different ways, not all of which can be calibrated from the same sub-blocks. As such, this chapter is aimed to explore and explain in great detail the different calibration ideas and mechanisms which can be used to minimise the effect of spurious tones at the synthesizer's output.

This thesis aims to try and correct only the delay SD and the SPE generated in the VCDL, respectively in the PD and CP sub-blocks. The DCD is considered to be negligible for the purpose of testing the two self-calibration mechanisms employed in this project.

Even so, taking into account the previously stated fact that all of the possible sources of errors (DCD, SPE and delay SD) have a shared effect over the magnitude of the spurs, the impact of DCD over the final SCR of the circuit post-calibration will be tackled in chapter 6 during the "Discussion" sub-section.

4.1 SPE self-calibration system

As stated in section 2.2.1, mismatches that appear during manufacturing in the PD and CP sub-blocks of the Frequency Synthesizer can lead to the generation of an improper steady-state coarse calibration voltage "V_{cal}", which in turn generates an Static Phase Offset between the reference signal and the output of the VCDL sub-block which cannot be corrected by the DLL without additional calibration systems.

Remembering the way the coarse calibration voltage is generated, the amplitude of the "V_{cal}" signal is decided based on the interval that the current sources in the CP are turned ON, which is turn controlled by the individual pulse width of the "UP" and "DN" signals from the output of the PD. As the pulse width of the "UP" and "DN" signals controls the "V_{cal}" generation mechanism, and each of the two outputs of the PD are determined by the rising edge of the reference and feedback (VCDL output) signals, then any imbalances in the CP or between the "UP" and "DN" signals can always be compensated by delaying, by a proper amount of time, the rising edge of the signals on the input of the PD.

There are multiple ways of implementing a circuit which delays a signal. Usual

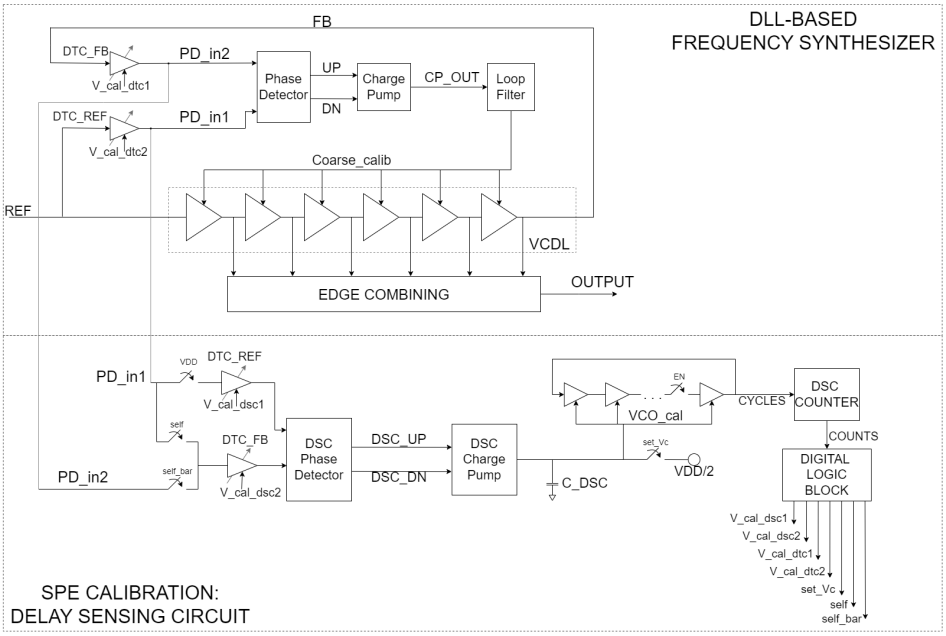

Figure 4.1: Block diagram of SPE self-calibration system

methods are employing a Digital-to-Time Converter component, using transmission lines with adjustable capacitance values or using current-starved inverter circuits with a tunable analog voltage as the control signal for the amount of delay added to the circuit. However, for the purposes of this thesis, a system similar to the one mentioned in [9] has been used, which implements a secondary DLL loop equipped with two sets of DTC devices, and which can be observed in the block diagram presented in Figure 4.1.

In the following subsection, a SPE calibration system will be explained which uses DTCs to delay the reference and the feedback signals on the input of the PD in order to compensate for any SPE generated by mismatches in it or in the CP block.

4.1.1 Previous work

Three other papers that implement different SPE calibration approaches have been found by the time this thesis has been written.

The work done by [22] employs a modified PD component in order to reduce the effect of UP and DN misalignments. Meanwhile, the approach of [23] is to utilise a digital calibration circuit to calculate and add a certain amount of delay at the output of each Delay Cell in their VCDL to minimise final SPE values (as well as delay SD, to some extent).

The main disadvantage of both calibration systems mentioned above is the necessity of keeping the calibration sub-blocks turned ON during the entire operation time of the Frequency Synthesizer circuit, leading to significantly increased

	[22]	[23]	[9]
Process	130 nm	0.18μm	130 nm
Year	2019	2006	2021
Operational Frequency Range	16 GHz	0.7-2 GHz	250 MHz
Maximum SPE	1.68 ps	$1.26^o = 3.5$ ps @1GHz	0.9 ps
Power	-	81 mW @2GHz	2.89 mW

Table 4.1: Comparison among previous works

power consumption needs.

Finally, the approach detailed in [9] represents the foundation for the SPE correction technique applied in this thesis as well. Compared with the other approaches in previous literature, this SPE calibration loop can run completely separated by the main Frequency Synthesizer circuit, and most of its sub-blocks can be disabled post-calibration, which leads to significantly reduced added power consumption by the SPE calibration mechanism.

The results obtained by these previous papers are detailed in Table 4.1.

4.1.2 SPE calibration mechanism

The self-calibration system undergoes two routines to correctly compensate for the SPE in the synthesizer, which are run after steady-state has been achieved in the Frequency Synthesizer. The first routine short-circuits the two inputs of the Delay-Sensing Circuit (DSC) to compensate any imbalances and mismatches existent in the actual Delay Sensing component, by finding appropriate states for DTC_DSC1 and DTC_DSC2. Afterwards, the short-circuit switch is turned OFF and the routine is repeated during the second stage using the proper reference and feedback signals on each DSC input paths (coming from DTC_REF and DTC_FB in the main DLL system). The newly found states will then be applied to DTC_REF and DTC_FB which will properly compensate the mismatches in the DLL PD and CP components, suppressing or minimising the SPE to a large extent.

Thus, the states identified in the second routine will delay one of the signals

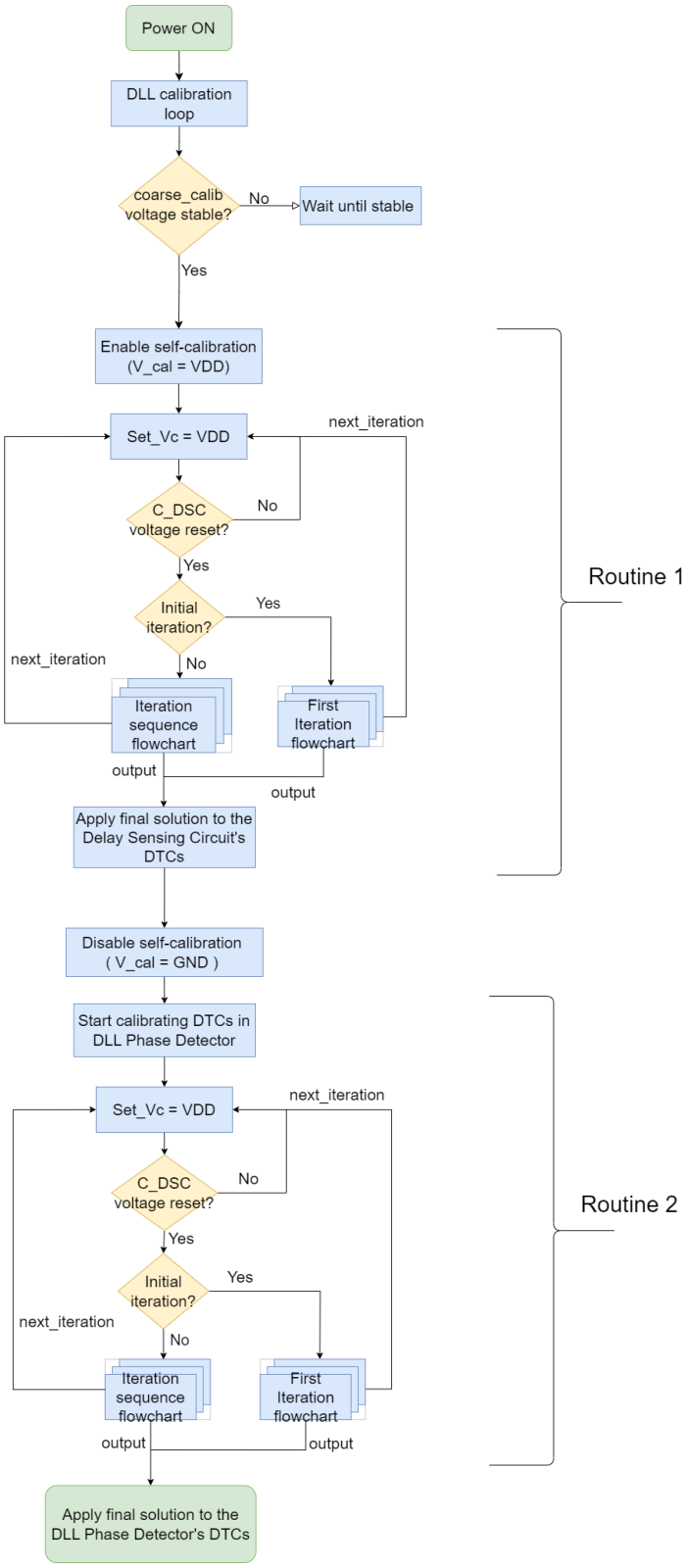

Figure 4.2: Main flowchart of SPE self-calibration system

at the input of the PD in the main system, bringing the reference and the VCDL output signals back in-phase.

The way the self-calibration algorithm functions is illustrated in the flowchart in Figure 4.2. From the presented flowchart, it can be observed that each routine is made up of a set of multiple iterations. The first iteration is slightly different than the rest of them, however both routines function in identical ways, the main difference being that the self-calibration routine identifies states for the DSC DTC components, while the second routine identifies states for the DTCs in the main synthesizer circuit (denoted as DTC_REF and DTC_FB).

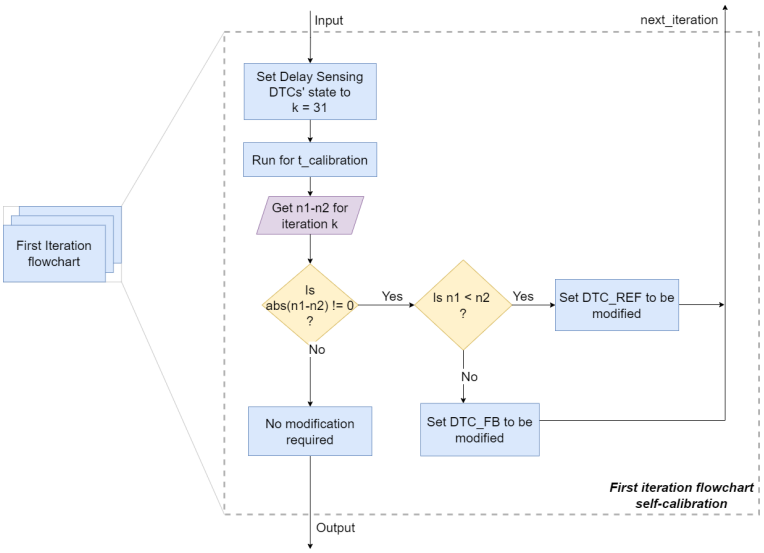

Figure 4.3: Flowchart of the first iteration in the SPE calibration routine

During any of the routines, the signals are passed through the Delay Sensing Circuit's DTCs and fed to the PD in the DSC which acts similar to a secondary DLL loop. The output of the DSC PD outputs an similar set of UP and DN signals similar in nature to the ones generated from the DLL circuit, which are then passed to a CP component to generate a voltage "VCO_CAL" based on the difference in rising edges in the signals coming from the DTCs in the main DLL circuit. This voltage will then charge or discharge the capacitor connected between the output of the CP and the Voltage-Controlled Oscillator (VCO) in the Delay Sensing block.

At first, the capacitor on the output line of the DSC CP is pre-charged to a certain voltage level. That level is the same level that the CP output during a steady-state situation, when both UP and DN have the same pulse width. Then, depending on the difference in phase between the "REF" and the "FB" signals from the DLL, the voltage across the capacitor changes, which leads to a change in the VCO's frequency. Then, the numbers of cycles running through the VCO is counted for a pre-determined calibration time of "t_{spe}".

The algorithm divides the total counts registered during the SPE calibration time t_{calib_SPE} into two equal intervals, denoted as "n_1" for the first $\frac{t_{calib\_SPE}}{2}$ time period and "n_2" for the latter half of the interval, each being counted during the first half of the calibration period (from 0 to N/2 reference cycles) and, respectively, the latter half of it (from N/2 to N reference cycles). Because the frequency of the VCO is constantly changing as long as the signals from the main DLL loop are not properly aligned, then "n_1" and "n_2" will be different from each other, as long as the calibration time for the SPE correction procedure is chosen properly. ($t_{calib\_SPE} = N * T_{REF}$, where N is a number of cycle appropriately chosen)

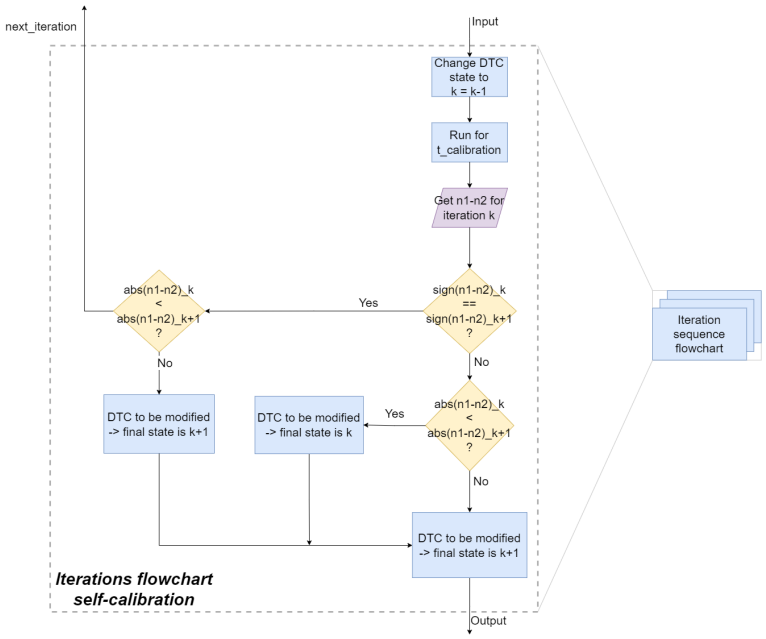

Figure 4.4: Flowchart of the common iterations in the SPE calibration routine

The difference in "n_2 - n_1" is then used to estimate the relative phase difference between the two signals from the main DLL. The count difference indicates how far "apart" the signals are, while the polarity indicates which signals is predominating (the dominating signal influences whether the pre-charge voltage over the cap is being discharged or charged towards one of the rails). Based on those two values, the corresponding DTC in the main loop will be re-calibrated to the following state and the procedure is re-run. This whole process encompasses "an iteration", as it is represented in the flowchart diagram from Figure 4.2.

A switch connected to a voltage supply is being employed on the output of the DSC CP circuit to "reset" the capacitor on the line back to the desired initial pre-charged voltage value before each iteration. This ensures that each calibration iteration starts from the same initial conditions.

The information about how the first and the rest of the iterations work can be

visualised in the flowcharts presented in Figures 4.3 and 4.4.

The main difference is that during the first iteration, it is determined whether any calibration needs to be done, and if yes, for which of the two DTC components (either in the DSC or in the main loop) need to be adjusted to properly compensate for the phase difference of the two signals. While for the rest of the iterations, the procedure continues until either the polarity of the "n_2" - "n_1" parameter reverses from previous iterations, or the magnitude does not change compared with the previous iteration in the procedure. Then, the routine is considered finished and the final calculated DTC states are used in the final circuit.

The main advantage of this type of self-calibration circuit is that it can run parallel and independent from the main DLL feedback routine, assuming no coupling between the two circuits. This is valuable, as most of the self-calibration circuit can be disabled after the final DTC states have been identified, maintaining the power consumption of the Frequency Tripler close to that of the non-calibrated version of the component.

4.1.3 Mathematical support

In [9], and the explanations presented in the previous subsection, an expression can be derived for the maximum calibration time necessary for DSC to minimise the SPE from the main DLL, as well as for the dead-zone of this circuit, depending on multiple design parameters.

As a start, the amount of charging or discharging over the capacitor on the output of the CP component should be estimated based on the amount of delay between the reference and the feedback signals in the main DLL:

$$\Delta V_{cap} = \frac{I_{CP}}{C_{DSC}} * \delta_{in} \tag{4.1}$$

Where ΔV_{cap} is the amount that the capacitor the rate of change in the capacitor voltage, I_{CP} is the current from the DSC CP sub-block, C_{DSC} is the value of the capacitor and δ_{in} is the amount of delay offset between the signals from the main DLL loop of the synthesizer.

Deriving the slope of the previous expression provides a way of calculating the charging or discharging rate during one signal cycle from the synthesizer's VCDL sub-block, as follows:

$$S_v = \frac{I_{CP}}{C_{DSC} * T_{REF}} * \delta_{in}, \quad T_{REF} = reference\ clock\ period \tag{4.2}$$

It can be observed that as long as S_v is a non-zero value, the capacitor on the output of the DSC CP will eventually charge or discharge towards one of the rails, given enough calibration time. This ensures that the calibration mechanism can always find the proper states to offset the SPE in the main DLL as long as enough time and DTC resolution are provided.

Representing the LSB of the DTC block as the smallest possible added delay, and considering the steps between DTC states to be constant for each of the two DTC pairs (either the DLL's pair or the DSC's pair) the total time for a calibration routine can be estimated as:

$$t_{total} = t_{cal\_DSC} * round(\frac{SPE_{DSC}}{LSB_{DSC}}) + t_{cal\_DLL} * round(\frac{SPE_{DLL}}{LSB_{DLL}}) \quad (4.3)$$

Where t_{cal_DSC} is the calibration time for an iteration during the first routine of the SPE self-calibration system, t_{cal_DLL} is the calibration time for a similar iteration for compensating the actual SPE in the main DLL component and round is a function which rounds the value inside to the closes integer value.

Next, in order to estimate the dead-zone (or the limit for the SPE calibration system's precision), an expression needs to be derived for "n_2 - n_1" based on the other parameters of the circuit. The first step towards that goal is to derive an expression for the amount of change in the VCO's frequency for a delay offset of δ_{in} between input signals. Assuming K_{VCO} to be a parameter that represents the "frequency-to-voltage" gain based on the output of the CP sub-block, then the slope of the ring oscillator's change in frequency is:

$$S_f = \frac{I_{CP}K_{VCO}}{C_{DSC}T_{REF}} * \delta_{in} \quad (4.4)$$

Considering the initial value for the VCO's frequency to be "f_0", then the number of cycles being counted from the VCO per reference cycle can be represented as:

- for the first reference cycle, COUNTS = $T_{REF} * f_0$
- for the second reference cycle, COUNTS = $T_{REF} * (f_0 + S_f T_{REF})$
- for the third reference cycle, COUNTS = $T_{REF} * (f_0 + 2 * S_f T_{REF})$
- ...
- for the N^{th} reference cycle, COUNTS = $T_{REF} * (f_0 + (N-1) * S_f T_{REF})$

The total count value for the VCO's cycles at the end of the N^{th} reference cycle during a complete iteration is the sum of the individual number of cycles for each step of the iteration, as follows:

$$COUNT_N = N * T_{REF} f_0 + \frac{N(N-1)}{2} * S_f T_{REF}^2 \quad (4.5)$$

Then, as n_1 is the total cycles counted in the first $\frac{N}{2}$ reference cycles and n_2 is the total cycles counted in the following $\frac{N}{2}$ reference cycles, then:

$$n_2 - n_1 = \frac{N^2}{4} S_f T_{REF}^2 = \frac{N^2}{4} \frac{I_{CP}K_{VCO}T_{REF}}{C_{DSC}} \delta_{in} \quad (4.6)$$

$$Gain_{DSC} = \frac{N^2}{4} \frac{I_{CP}K_{VCO}T_{REF}}{C_{DSC}} \quad (4.7)$$

Thus, an estimation of the dead-zone of the calibration mechanism can be derived, assuming the condition for the dead-zone is that $n_2 - n_1 < 1$:

$$|\delta_{deadzone}| = |n_2 - n_1| < \frac{4 * C_{DSC}}{N^2 * I_{CP}K_{VCO}T_{REF}} \quad (4.8)$$

4.2 Delay SD self-calibration system

At the beginning of the circuit operation, the VCDL is not calibrated at all (output of VCDL is not in-phase or locked to the reference signal), so all of the delays are different from a desired average μ delay, leading to spurious tones in the spectrum of the synthesizer's output signal.

After the lock state is achieved, the reference signal and the output of the VCDL are in-phase, however, the delay added by the in-between Delay Stages might still be skewed in such a way that on average, the total delay added by them is "0 ps".

There are multiple ways of designing a calibration system that might correct this issue, and some of these ideas will be explained in the next subsection before exploring in more detail the system that has been chosen to be designed and refined for this thesis project.

4.2.1 Previous versions of delay SD calibrations

Two other approaches have been tried previously to regulate delay mismatches between different Delay Cell outputs, which are described in papers [24] and [25].

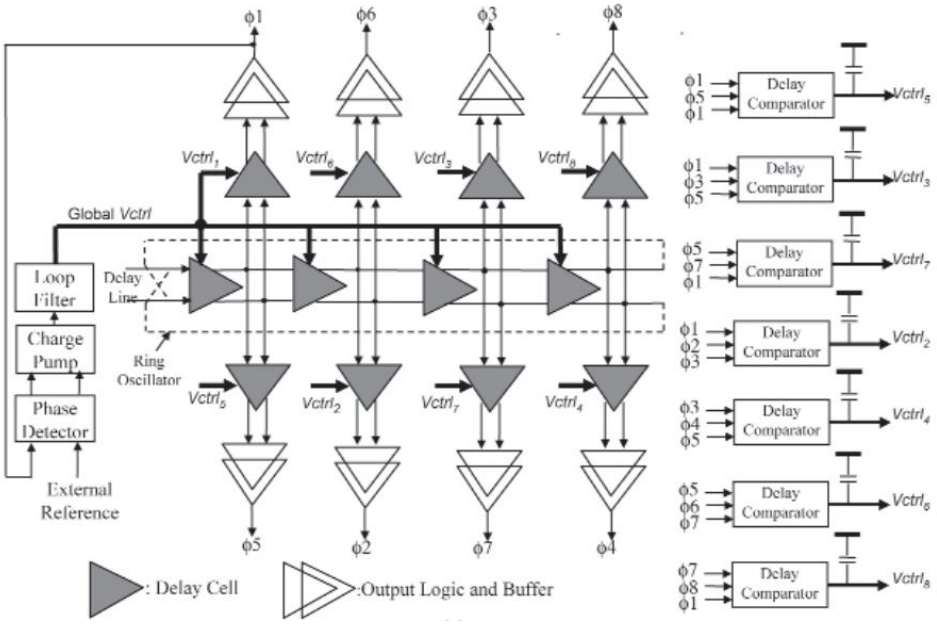

Figure 4.5: Self-calibration method with Delay Comparators, referenced from [24]

The first approach, described in [24], employs a differential VCDL setup where the differential Delay Cell outputs are connected to multiple variable-delay voltage buffer blocks to obtain the desired delayed outputs. Specific combinations of these non-calibrated outputs are sent to multiple three-input delay comparators

to estimate the delay mismatches. The output of these comparators is then used as the delay adjusting voltage signal for the voltage buffers in the delay line. The schematic implementation of this work can be observed in Figure 4.5.

Although this approach can result into better calibrated delays, and thus into reduced spurious tones at the synthesizer's output, its approach necessitates the use of multiple added circuit block which need to be kept turned ON throughout the operation time of the transceiver, resulting in a bigger total power consumption for the final circuit.

Another way of ensuring that all Delay Cells are calibrated for the same amount of delay is the approach suggested in [25]. Here, a two-by-two comparison approach is suggested, where reference Delay Cell "D_X" is employed in combination with a very precise PD to compare outputs of successive pair of Delay Cells from the VCDL. An N-to-2 multiplexer circuit that can select which two Delay Cells from the synthesizer's VCDL to be compared and re-calibrated, similar to Figure 4.6. The VCDL block will be composed of the necessary N Delay Cells, an input buffer Delay Cell, and an output buffer Delay Cell.

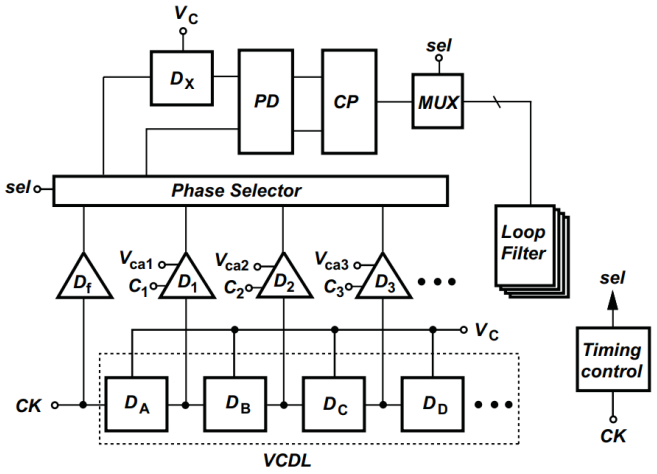

Figure 4.6: Self-calibration method with Phase Comparator setup, referenced from [25]

The Multiplexer would select the outputs of two successive Delay Cells based on a series of control signals from a digital logic block. For every pair of two outputs, the first output will be directed through the reference Delay Cell before being used as an input for the PD, while the second, already delayed signal, output of the multiplexer will be used as the second input. In this way, the delay added by the second Delay Cell selected from the VCDL will be aligned to the delay offset added by D_X. By repeating this process for the whole VCDL, every Delay Cell in the delay line will be adjusted to replicate the offset of D_X. Even if D_X also suffers from mismatches, as long as every cell in the VCDL adds the same offset to their respective input signals, the DLL feedback loop can readjust the signals to fit within a T_{ref} period.

	[25]	[24]
Process	0.13μm	0.25μm
Year	2008	2001
Operational Frequency Range	833 MHz	100-250 MHz
Maximum delay SD	4.47 LSB	10.2 ps
Power	67.2 mW	80 mW

Table 4.2: Comparison among previous works

The output of the PD is then fed to a secondary CP block, which gives a new, individual calibration signal for each of the delay cells. Each Delay Cell can be equipped with a second current starving inverter and transistor system for "fine delay calibration" can be added to the Delay Cells to enable the offset compensation necessary to align the offsets of the VCDL cells with that of the reference cell D_X.

However, this calibration system necessitates multiple loop filters, equipped with capacitors big enough to retain the proper calibration voltage for the duration of a reference period, besides the components already existent in the main DLL circuit. All of the components in this calibration mechanism would also need to be constantly turned ON, adding a significant amount to the total power consumption of the Frequency Synthesizer circuit. Moreover, the multiplier component needs to be designed in a way that adds a symmetrical amount of loading on each Delay Stage in the VCDL and the total precision of this calibration mechanism is dependent on the precision of the PD employed in the secondary calibration loop.

The results for the previous works are illustrated in Table 4.2

Because of the reasons indicated previously for each of the previous works, a different approach has been chosen for this thesis, which is explained in the following subsection.

4.2.2 System of equations

Another approach for a self-calibration system would be to modify the circuit to permit multiple fine calibration states for each individual Delay Cell used in the VCDL, as well as modifying the VCDL block to be able to run as a ring

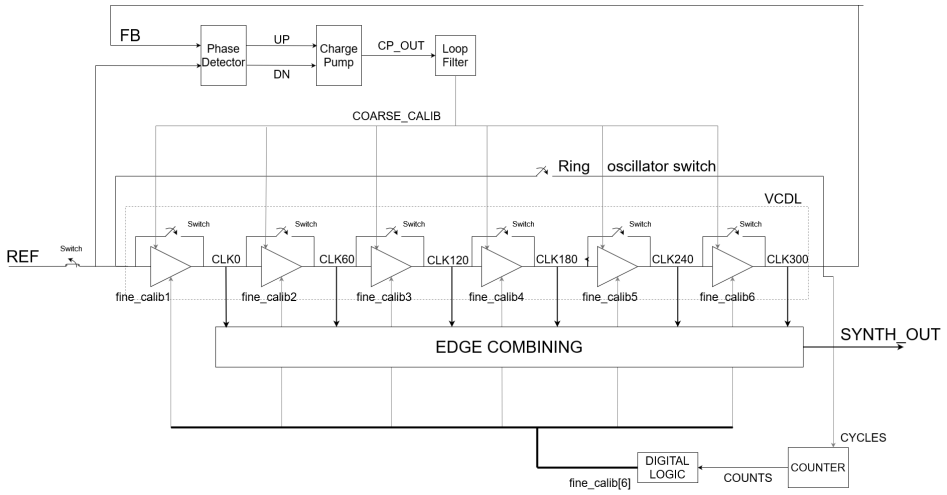

Figure 4.7: Self-calibration method with Ring oscillator setup

oscillator for a pre-determined period of time, only for calibration purposes. The fine calibration would add a smaller quantity of delay compared with the coarse calibration provided by the main DLL loop calibration. In that case, if a simplified mathematical representation of the signal's path through the VCDL during a T_{ref} period could be derived, then a system of equations could be constructed, where the unknowns are represented by the effect of mismatches from each individual Delay Cell.

Running the VCDL in a ring oscillator setup would allow the system to count the number of cycles repeating inside the ring oscillator's signal path for a pre-defined calibration time t_{cal} which could then be used to calculate an average cycle time. The averaging effect would apply in equal measure to the delay added by the unbalanced Delay Cells (which have been previously calibrated by the DLL to add only a T_{ref} amount of cumulative delay from input to the output of the VCDL block) as well as to the delay offset added by the mismatches in each of the cells.

A system of switches, together with a large enough number of individual fine calibration states, could ensure that a number of different and unique settings could be used for different iterations of the ring oscillator, where each iteration represents running the VCDL as a ring oscillator for t_{cal} time units. Thus, a large enough set of independent equations - meaning individual iterations of the ring oscillator with different sets of settings - can be obtained, which would lead to a determined system of equations as long as the individual sets of settings are chosen properly.

Thus, if the mismatches existent in the N Delay Cells of the VCDL block are represented as mismatches occurring in the added fine calibration circuit - denoted as fine mismatches - and mismatches occurring in the rest of the Delay Cell structure - denoted as coarse mismatches -, then a set of $2 * N$ equations is needed to construct a determined system of equations. Such a system provides only

one unique solution for its unknowns, which could result in a precise estimation of the mismatch effects in the VCDL structure, together with a set of fine calibration states which could properly compensate for it.

The number of equations needed, together with the predicted maximum value for an individual delay offset caused by such mismatches and the number of Delay Cells, will determine the resolution for the fine calibration system and the total number of individual "fine calibration states" (or the complexity of the fine calibration system).

All calculations for obtaining the proper estimations of delay offsets and solutions for compensating them in the final set of delay line settings can be executed in a dedicated Digital Logic block, the design of which is not covered in the scope of this thesis.

The advantages are that this method doesn't add any new mismatches to the existent circuit which cannot be estimated by the calibration system, that the method is time-tested and its calibration resolution is only dependant on the calibration time used, and that the power consumption amount added by this calibration routine is only needed during calibration and not throughout the operation of the whole receiver chain.

4.2.3 Mathematical support

If the behaviour of the Delay Stages of the VCDL can be approximated by a set of unknown variables for the delay added by each Delay Stage, then an expression for the total "cycle" time for the reference signal to arrive at the output of the VCDL can be expressed as:

$$cycleTime_k = delayStage_1 + delayStage_2 + ... + delayStage_n \tag{4.9}$$

Where "k" is the number of the current cycle, $cycleTime_k$ is the total time for the reference signal to pass through the VCDL once, and $delayStage_{1-n}$ is the delay added by each Delay Stage (along with their individual time skews from mismatches).

This formula applies to assume that SPE would be negligible in the final system. Otherwise, SPE could be modelled by a constant "T_{SPE}" added at the end of the equation.

Furthermore, the delay added by each Delay Stage can be expressed as the desired delay, indicated by the voltage control from the PD and CP section of the DLL, modified by a finite time value "tm_n" that represents the effect of the mismatch.

Mismatches that appear during production affect the physical properties of transistors, which in turn affect how the transistor responds to input signals in different analogue configurations, inducing unwanted deviations of a circuit's response from the desired values, in this case from μ:

$$delayStage_n = \mu + tm_n \tag{4.10}$$

Where μ is the desired delay that should be added by the Delay Stage and tm_n is the representation of the delay skews caused by the mismatches in the Delay Stage structure.

The formula 4.10 applies assuming that the effect of the mismatches added by the manufacturing process in one Delay Stage structure doesn't change dynamically (for example: with temperature or operation time). This assumption will be denoted in the following sections as the static mismatch effect assumption.

However, the mismatch effect can be different between Delay stages in the same EC DLL circuit, which is why each delay cell has its own mismatch effect variable in the formula below, obtained by introducing equation (4.10) in (4.9):

$$cycleTime_k = (\mu + tm_1) + (\mu + tm_2) + ... + (\mu + tm_n) \tag{4.11}$$

$$cycleTime_k = N * \mu + tm_1 + tm_2 + ... + tm_n$$

Where "N" is the total number of Delay Stages in the VCDL.

The method of self-calibration employed in this thesis is using additional fine delay control on each Delay Stage, in the form of a DTC system together with a ring oscillator and counter setup.

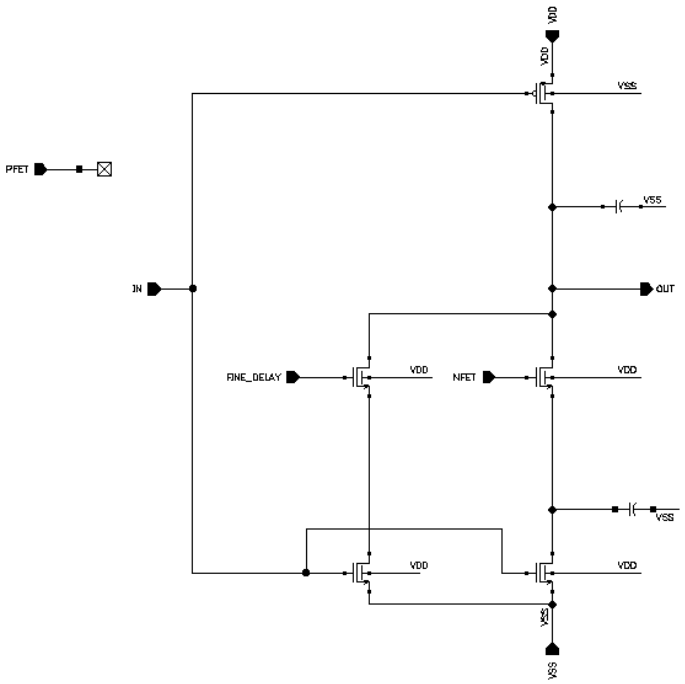

Figure 4.8: Delay Cell modified for both coarse and fine delay calibration

The DTC system consists of a series of digitally controlled binary-weighted transistors added to the Delay Stage structure, which can be turned on or off to add a set amount of fine delay over the initial delay added by the DLL control voltage setting.

Because the calibration mechanism in each Delay Stage is made out of binary-weighted transistors, the fine delay added to each stage can be viewed as a multiple of the smallest delay obtainable, namely a multiple of the time-step of the system tc_n (where "n" is the number of the current Delay Stage). Introducing this notion in the formulas 4.10 and 4.11 results in:

$$delayStage_n = (\mu + tm_n) + tc_n \tag{4.12}$$

$$cycleTime_k = (\mu + tm_1 + tc_1) + (\mu + tm_2 + tc_2) + ... + (\mu + tm_n + tc_n) \tag{4.13}$$

Where:

$$tc_n = bitState * (time\_step + fine\_mismatch) = k * c_n \tag{4.14}$$

$$bitState = (2^{bitConfiguration} + 1);$$

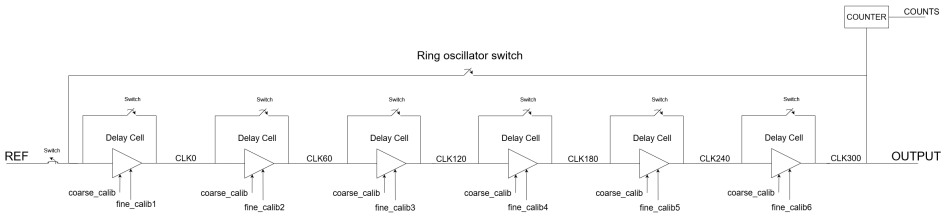

Figure 4.9: Self-calibration modification of VCDL

In order to obtain the time necessary for the reference signal to pass from the input to the output of the VCDL component (or to complete one "cycle") an optional ring oscillator mechanism can be implemented for the delay line, similar to Figure 4.9. The output and the input of the VCDL component are connected through an optional branch (commuted by a switch) controlled by a separate logic control unit.

When the ring oscillator branch is turned on, the DLL delay control line is disconnected from the CP and LF and connected to a steady voltage source with a voltage amplitude equal to the common-mode of the CP circuit to maintain all of the Delay Stages at the same delay setting.

Afterwards, the ring oscillator is left to run until the signal (a time period of $t_{stabilization}$). Then, the output of the ring oscillator is fed to a binary counter through a multiplexer component, to maintain an equal load across all Delay Stages.

Inside a ring oscillator, because the output of the component is fed as the input, the mismatches from each Delay Stage, together with the effect of the fine calibration settings used, add up over multiple cycles, resulting in a slightly different number of total cycles being counted during a large enough interval of time $t_{calibration}$. The total number of cycles is divided by the calibration time to get the amount of time per cycle for a certain fine calibration scheme, as follows:

$$cycleTime_k = \frac{totalCycles}{t_{calibration}} \tag{4.15}$$

Where "k" is the number of the calibration scheme / the calibration iteration in use at the moment.

Combining equations 4.13 with 4.15, a connection can be defined between the time of a cycle and a certain fine calibration scheme being used, irrespective of the control signal from the DLL. This expression has a number of six unknown elements or variables defined by the delay skews added by a Delay Stage's structure (without accounting for mismatches from the fine calibration mechanism).

Considering the above simplified expression of the behaviour of the delay line as an independent equation, then using a different setup, a system of equations could be obtained such that the values of the delay skews added by structural mismatches in the Delay Stages could be uniquely identified for a given production of the circuit.

Each equation would represent the value obtained for a given fine delay control scheme, run in a ring oscillator setup for the same amount of calibration time $t_{calibration}$. Where a fine delay control scheme represents a set of digital bits configuration for each Delay Stage in the VCDL.

An observation can be made that if the fine calibration schemes can be proven to be independent across different calibration runs, then the system of equations obtained from these different runs is determinate, meaning that unique values for the unknown mismatches effects in each Delay Stage can be obtained.

As the system of equations is always determinate, following the rules for choosing coefficients explained above, then a unique solution can be calculated for the mismatches introduced by the structure of the Delay Stages. Constructing on the mathematical basis presented in Section 2.3, the following set of formulas can be derived to represent the behaviour of the calibration mechanism:

$$delayStage_n = (t\_coarseCalib + t\_delayMismatch) + t\_fineCalib = (\mu + tm_n) + tc_n \tag{4.16}$$

where $\mu + tm_n$ can be expressed as s_n.

$$tc_n = k * c_n \tag{4.17}$$

$$cycleTime_k = (\mu + tm_1 + tc_1) + (\mu + tm_2 + tc_2) + ... + (\mu + tm_n + tc_n) \tag{4.18}$$

$$\begin{cases} k_{i\omega}/k_{p\omega} = 2\pi \times 10 \\ \left| \frac{k_{p\omega}s + k_{i\omega}}{s} \cdot \frac{1}{Ts+1} \right|_{S=\mathrm{j}\cdot 2\pi} = 1 \end{cases} . \tag{4.19}$$

$$\begin{cases} cycleTime_1 = (\mu + tm_1 + tc_1 1) + (\mu + tm_2 + tc_2 1) + (\mu + tm_3 + tc_3 1) \\ cycleTime_2 = (\mu + tm_1 + tc_1 2) + (\mu + tm_2 + tc_2 2) + (\mu + tm_3 + tc_3 2) \\ cycleTime_3 = (\mu + tm_1 + tc_1 3) + (\mu + tm_2 + tc_2 3) + (\mu + tm_3 + tc_3 3) \\ cycleTime_4 = (\mu + tm_1 + tc_1 4) \\ cycleTime_5 = (\mu + tm_2 + tc_2 5) \\ cycleTime_6 = (\mu + tm_3 + tc_3 6) \end{cases} \tag{4.20}$$

$$\begin{cases} cycleTime_1 = (s_1 + k * c_1) + (s_2 + (k+1) * c_2) + (s_3 + (k+2) * c_3) \\ cycleTime_2 = (s_1 + (k+1) * c_1) + (s_2 + (k+2) * c_2) + (s_3 + (k+3) * c_3) \\ cycleTime_3 = (s_1 + (k+2) * c_1) + (s_2 + (k+3) * c_2) + (s_3 + (k+4) * c_3) \\ cycleTime_4 = (s_1 + (k+5) * c_1) \\ cycleTime_5 = (s_2 + (k+6) * c_2) \\ cycleTime_6 = (s_3 + (k+7) * c_3) \end{cases} \tag{4.21}$$

$$\begin{bmatrix} s_1 & s_2 & s_3 & k * c_1 & (k+1) * c_2 & (k+2) * c_3 \\ s_1 & s_2 & s_3 & (k+2) * c_1 & (k+3) * c_2 & (k+4) * c_3 \\ s_1 & s_2 & s_3 & (k+3) * c_1 & (k+4) * c_2 & (k+5) * c_3 \\ s_1 & 0 & 0 & (k+5) * c_1 & 0 & 0 \\ 0 & s_2 & 0 & 0 & (k+6) * c_2 & 0 \\ 0 & 0 & s_3 & 0 & 0 & (k+7) * c_3 \end{bmatrix} = \begin{bmatrix} cycleTime_1 \\ cycleTime_2 \\ cycleTime_3 \\ cycleTime_4 \\ cycleTime_5 \\ cycleTime_6 \end{bmatrix} \tag{4.22}$$

$$\begin{bmatrix} 1 & 1 & 1 & k & (k+1) & (k+2) \\ 1 & 1 & 1 & (k+2) & (k+3) & (k+4) \\ 1 & 1 & 1 & (k+3) & (k+4) & (k+5) \\ 1 & 0 & 0 & (k+5) & 0 & 0 \\ 0 & 1 & 0 & 0 & (k+6) & 0 \\ 0 & 0 & 1 & 0 & 0 & (k+7) \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} cycleTime_1 \\ cycleTime_2 \\ cycleTime_3 \\ cycleTime_4 \\ cycleTime_5 \\ cycleTime_6 \end{bmatrix} \tag{4.23}$$

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & k & (k+1) & (k+2) \\ 1 & 1 & 1 & (k+2) & (k+3) & (k+4) \\ 1 & 1 & 1 & (k+3) & (k+4) & (k+5) \\ 1 & 0 & 0 & (k+5) & 0 & 0 \\ 0 & 1 & 0 & 0 & (k+6) & 0 \\ 0 & 0 & 1 & 0 & 0 & (k+7) \end{bmatrix}^{-1} \begin{bmatrix} cycleTime_1 \\ cycleTime_2 \\ cycleTime_3 \\ cycleTime_4 \\ cycleTime_5 \\ cycleTime_6 \end{bmatrix} \tag{4.24}$$

4.2.4 Multi-order mismatch effects over delay

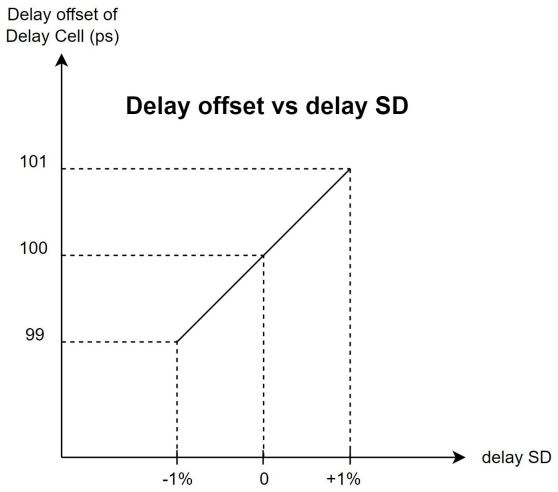

Figure 4.10: First order mismatch behaviour over delay of Delay Stage

A previous assumption: was that the expected effect of the mismatches created during manufacturing can be approximated by a linear function similar to Figure 4.10. The effect of the mismatches induced to the transistors in a classic Delay Stage structure (no added digital calibration) was previously represented as an added term tm_n to the expression of a Delay Stage's delay.

However, the effect of any source of mismatches in the Delay Stage component might actually manifest as a second or higher order function, similar to Figure 4.11. For ease of explanation, only the second-order behaviour of the classic Delay Stage structure will be assumed in this sub-section.

If we consider a second-order behaviour, then the effect of the total mismatch can be modelled by using a second mismatch variable $tm2_n$ which represents the

modification of the linearity of the slope across the X-axis. Of course, this approximation applies only as long as the static mismatch effect assumption holds true.

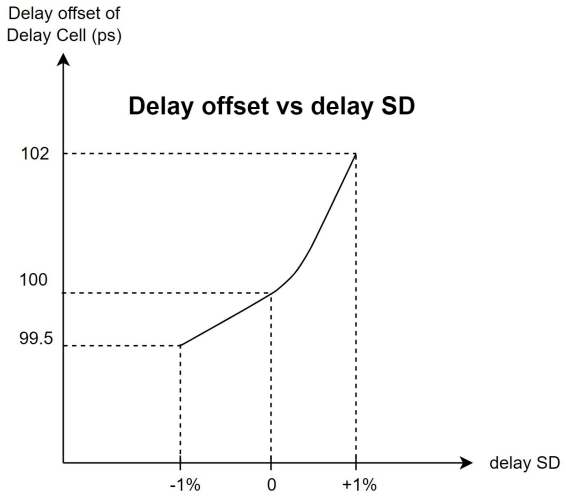

Figure 4.11: Second order mismatch behaviour over delay of Delay Stage

This effect results in the modification of equations (4.16) and (4.18):

$$delayStage_n = (\mu + tm_n + tm2_n) + tc_n \tag{4.25}$$

$$cycleTime_k = (\mu+tm_1+tm2_1+tc_1)+(\mu+tm_2+tm2_2+tc_2)+...+(\mu+tm_n+tm2_n+tc_n) \tag{4.26}$$

Thus, the new approximation of the "$cycleTime$" of calibration iteration can be expressed as an equation with 2 unknowns per Delay Stage, resulting in $2 * n$ total individual unknown variables. The new system can, of course, be uniquely resolved by employing the coefficient matrix model derived above which ensures the system equations are independent and the system can be solved by using a number of $2 * n$ equations.

4.2.5 Fine calibration mismatches

Besides multi-order mismatch effects, another problem arises if the calibration system also introduces its own mismatches into the system, affecting the behaviour of delay calibration. Similarly to the previous sub-section, this new mismatch source can be linear in nature or multi-ordered. However, as the logic from the previous chapter can be employed for any multi-order mismatch behaviour as long as all the previous assumptions hold true, then only a linear behaviour will be taken into account.

As the calibration mechanism can also add a certain amount of current draw over the effect of the DLL current-starving technique, the effect of any mismatches from this system can be modelled as additional delay skew terms "tcm_n" (calibration mismatch delay for Delay Stage number n) in the final equations of "$cycleTime_k$" and "$delayStage_n$" calculations, resulting in:

$$delayStage_n = (\mu + tm_n + tm2_n) + (tc_n + tcm_n) \tag{4.27}$$

$$cycleTime_k = (\mu + tm_1 + tm2_1 + tc_1 + tcm_1) + ... + (\mu + tm_n + tm2_n + tc_n + tcm_n) \tag{4.28}$$

As per the previous explanations, this model translates into a new unknown variable in the final system of equations which can be covered by employing a proper determinate coefficient matrix structure, needing "n" additional equations added and fine calibration states added.

4.2.6 Script and errors to be accounted for proper Delay cell mismatch estimation

In an ideal differential 2 delay chains with no mismatches and no fine calibration used, a perfect phase difference of 120 °between output signals of each delay cell will be obtained as per 4.29. However, in a delay chain with mismatches, there will be minute differences in the phase difference between clock signals at 400 mV($V_{DD}/2$) based on which delay cells have a mismatch.

$$Phase\,Difference(\phi) = 360^o / Number\,of\,Delay\,Cells(N) \tag{4.29}$$

For the setup with mismatches in delay cells, simulations are run for a certain calibration time say, 4μs to compute the cycle counts for each iteration i.e., by enabling and disabling delay cells based on the switch control states. For example, let's say that the switch control state is decimal '5' which can be represented in binary format as '101', the first and the third delay cells are enabled and the second delay cell is disabled which means that the mismatches come only from the delay cells that are enabled and not from the ones that are disabled. Similarly for the switch control state '7', all the delay cells in the delay chain are enabled.

The number of fine calibration states is defined by the bit resolution. In this case, the bit resolution of '5' is chosen which in turn gives 32 states. Since each delay chain uses 3 delay cells, the matrix computation is also divided into 2 (6*6) matrices, one for each delay chain. In our case, we have a square n-by-n(6-by-6) matrix where the rows of the matrix represent iterations whereas, the first 3 columns represent switch control values in binary format(0 or 1) and the next 3 columns represent fine calibration states for each iteration.

The first half (6*3) matrix in the overall (6*6) matrix can be written as follows:

$$\left[D\right] = \begin{bmatrix} 5 \\ 7 \\ 6 \\ 2 \\ 1 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.30}$$

The second half (6*3) matrix in the overall (6*6) matrix can be written as follows:

$$\left[Fine\ Calibration\ States\right] = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix} \tag{4.31}$$

However, the above matrix would have certain values as '0' since disabling some delay cells based on the switch control iterations means that we disable the fine calibration control for those delay cells as well. Therefore, the above matrix[Fine Calibration States] when multiplied with [D] can be written as follows:

$$\left[Fine\ Calibration\ States\right] = \begin{bmatrix} 1 & 0 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \\ 0 & 11 & 0 \\ 0 & 0 & 15 \\ 16 & 17 & 18 \end{bmatrix} \tag{4.32}$$

The final (6*6) matrix can be written as follows:

$$\left[A\right] = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 3 \\ 1 & 1 & 1 & 4 & 5 & 6 \\ 1 & 1 & 0 & 7 & 8 & 0 \\ 0 & 1 & 0 & 0 & 11 & 0 \\ 0 & 0 & 1 & 0 & 0 & 15 \\ 1 & 1 & 1 & 16 & 17 & 18 \end{bmatrix} \tag{4.33}$$

Based on the configuration of the switch control states('0' and '1') in the rows and columns of the matrix, there are only certain matrix configurations that follow the properties of the matrix to be an invertible matrix.

Since the above matrix [A] satisfies the properties of an invertible matrix as mentioned in the section 2.5 for which the determinant of [A] = -729, this matrix was chosen as the final [A] to solve the system of equations.

Based on the switch control states and the fine calibration states in each iteration, the cycle counts obtained for a certain calibration time is different between each iteration, which in turn makes the *cycleTime* different for each iteration.

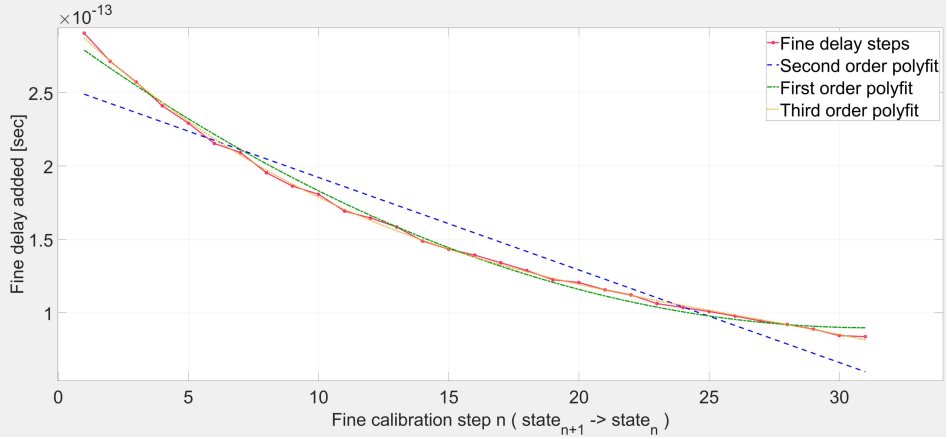

Figure 4.12: Polynomial approximations of fine calibration behaviour across states

The first step added by the fine calibration states from 0 to 1 is approximately 290.5 fs. However, the relative step size between fine calibration states does not increase, nor decrease, linearly which is represented as the red line in the Fine Delay steps plot, presented in Figure 4.12. Therefore, a second-order poly fit is

used in MATLAB to approximate the fine delay steps as shown by the green curve in the figure below along with the non-linear curve of fine delay step sizes to find the offset for each fine state.

The compounded offsets are computed as the difference between fine delay steps and the values from the second-order poly fit. The Second order poly fit for the fine steps estimation can be written in the form of a second-order polynomial as shown below:

$$f(x) = (0.207e^{-15} * x^2) - (12.925e^{-15} * x) + (291.704e^{-15}) \tag{4.34}$$

The compounded offsets from the 1st state to the 18th state can be written as:

$$\left[Compounded\ Offsets\right] = \left[10^{-11}\right] * \begin{bmatrix} 0 & 0.0012 & 0.0035 \\ 0.0071 & 0.0118 & 0.0177 \\ 0.0246 & 0.0325 & 0.0414 \\ 0.0512 & 0.0620 & 0.0736 \\ 0.0860 & 0.0992 & 0.1131 \\ 0.1277 & 0.1430 & 0.1589 \end{bmatrix} \tag{4.35}$$

Since few delay cells are disabled based on the switch control iterations, the above matrix when multiplied with [D] from equation 4.30 can be rewritten as follows:

$$\left[Compounded\ Offsets\right] = \left[10^{-11}\right] * \begin{bmatrix} 0 & 0 & 0.0035 \\ 0.0071 & 0.0118 & 0.0177 \\ 0.0246 & 0.0325 & 0 \\ 0 & 0.0620 & 0 \\ 0 & 0 & 0.1131 \\ 0.1277 & 0.1430 & 0.1589 \end{bmatrix} \tag{4.36}$$

The compounded offsets for each fine calibration state from the above matrix

is added for each iteration and written as follows:

$$\begin{bmatrix} Fine\ Calib\ Comp\ Offsets_5 \\ Fine\ Calib\ Comp\ Offsets_7 \\ Fine\ Calib\ Comp\ Offsets_6 \\ Fine\ Calib\ Comp\ Offsets_2 \\ Fine\ Calib\ Comp\ Offsets_1 \\ Fine\ Calib\ Comp\ Offsets_7 \end{bmatrix} = \begin{bmatrix} 10^{-11} \end{bmatrix} * \begin{bmatrix} 0.0035 \\ 0.0366 \\ 0.0571 \\ 0.0621 \\ 0.1131 \\ 0.2707 \end{bmatrix} \tag{4.37}$$

$t_{simulation}$ matrix can be written as the $cycleTime$ matrix for all 6 iterations with fine calibration as shown in equation 4.20 whereas x_{script} matrix is the matrix that consists of the SPE for the 3 delay cells in the delay chain that the script expects, which are obtained from the simulations as well as the fine calibration step added by the fine calibration states. Using $t_{simulation}$ and A, we can write $x_{simulation}$ as follows:

$$x_{simulation} = A/t_{simulation} \tag{4.38}$$

Using the x_{script} matrix, we can compute t_{script} matrix as follows:

$$t_{script} = A/x_{script} \tag{4.39}$$

By comparing the matrices $t_{simulation}$ and, t_{script}, it can be seen that there are differences in the values of both the (6*1) matrices. Since the values from $t_{simulation}$ matrix and t_{script} matrix have some offset, a method is proposed in order to compute the offsets between the two matrices.

To compute the offsets added by the multiplexers, a transient simulation is run for the same calibration time of $4\mu s$ without the effect of fine calibration. The cycle counts obtained using the matrix values(iterations) gives the desired $cycleTime$ for each iteration considering only the effect of multiplexers in both the delay chains.

The offset added by the multiplexer is different based on the switch control state. For the switch control state '1' where the signal passes through the delay cell and the multiplexer, the case of iteration '111' where all the 3 delay cells in the delay chain are enabled, the $cycleTime$ obtained is 316.5 ps and hence, the offset added by the multiplexer is 316.5 ps - 300 ps => 16.5 ps.

However, the offset added by the multiplexer for the switch control state of '0' is slightly different from the one that was obtained for switch control state '1'. The difference in Multiplexer offsets for switch control states '1' and '0' is explained through a set of equations below:

Since the $cycleTime_{111}$ for the iteration '7' is approximately 316.5 ps where the switch control states for all the delay cells in the delay chain are '1', $cycleTime_{111}$ can be written as follows:

$$cycleTime_{111} = ((3 * \mu) + Multiplexer_{Offset_1}) \tag{4.40}$$

From the above equation, $Multiplexer_{Offset_1}$ can be written as:

$$Multiplexer_{Offset_1} = cycleTime_{111} - (3 * \mu) \tag{4.41}$$

where μ in the above equation is defined as the desired phase difference between the delay cells of a delay chain which is equal to 100 ps.

Similarly for example if the $cycleTime_{101}$ for the iteration '5' is approximately 231 ps where the switch control states for the delay cells '1' and '3' in the delay chain are '1' and the switch control state for delay cell '2' is '0' and by substituting the value of $Multiplexer_{Offset_1}$ from equation 4.41 in equation 4.42, $cycleTime_{101}$ can be written as follows:

$$cycleTime_{101} = ((2 * \mu) + Multiplexer_{Offset_{101}} + Multiplexer_{Offset_1}) \tag{4.42}$$

From the above equation, $Multiplexer_{Offset_{101}}$ for $cycleTime_{101}$ can be written as:

$$Multiplexer_{Offset_{101}} = \frac{(cycleTime_{101} - (2 * \mu) - Multiplexer_{Offset_1})}{2} \tag{4.43}$$

However, the $Multiplexer_{Offset_0}$ can be slightly different for different iterations based on the switch control states.

Similarly, $Multiplexer_{Offset_{001}}$ for $cycleTime_{001}$ can be written as:

$$Multiplexer_{Offset_{001}} = \frac{(cycleTime_{001} - \mu - Multiplexer_{Offset_{111}})}{2} \tag{4.44}$$

Since the switch control iterations from 4.30 is represented as,

$$\begin{bmatrix} '5' =' 101' \\ '7' =' 111' \\ '6' =' 110' \\ '2' =' 010' \\ '1' =' 001' \\ '7' =' 111' \end{bmatrix} \tag{4.45}$$

The $Multiplexer_{Offset_1}$ and $Multiplexer_{Offset_0}$ values computed for respective iterations based on the switch control states ('0' and '1') are added along with

the existing values in the t_{script} matrix and is defined as $t_{scriptNew}$. $t_{scriptNew}$ can be written as follows:

$$\begin{bmatrix} t_{scriptNew_{101}} \\ t_{scriptNew_{111}} \\ t_{scriptNew_{110}} \\ t_{scriptNew_{010}} \\ t_{scriptNew_{001}} \\ t_{scriptNew_{111}} \end{bmatrix} = \begin{bmatrix} cycleTime_{101} - Multiplexer_{Offset_{101}} - Multiplexer_{Offset_1} \\ cycleTime_{111} - Multiplexer_{Offset_1} \\ cycleTime_{110} - Multiplexer_{Offset_{110}} - Multiplexer_{Offset_1} \\ cycleTime_{010} - (2 * Multiplexer_{Offset_{010}}) - Multiplexer_{Offset_1} \\ cycleTime_{001} - (2 * Multiplexer_{Offset_{001}}) - Multiplexer_{Offset_1} \\ cycleTime_{110} - Multiplexer_{Offset_{110}} - Multiplexer_{Offset_1} \end{bmatrix} \tag{4.46}$$

It is to be noted that the above matrix, $t_{scriptNew}$ is computed only with Multiplexer Offsets based on the switch control iterations and without fine calibration. The $x_{scriptNew}$ matrix can be computed as follows:

$$x_{scriptNew} = A/t_{scriptNew} \tag{4.47}$$

However, the values of $t_{scriptNew_{FineCalib}}$ matrix when fine calibration is added can be computed as follows:

$$\begin{bmatrix} t_{scriptNew_{FineCalib}} \end{bmatrix} = \begin{bmatrix} t_{scriptNew_{101}} + Fine\ Calib\ Comp\ Offsets_5 \\ t_{scriptNew_{111}} + Fine\ Calib\ Comp\ Offsets_7 \\ t_{scriptNew_{110}} + Fine\ Calib\ Comp\ Offsets_6 \\ t_{scriptNew_{010}} + Fine\ Calib\ Comp\ Offsets_2 \\ t_{scriptNew_{001}} + Fine\ Calib\ Comp\ Offsets_1 \\ t_{scriptNew_{111}} + Fine\ Calib\ Comp\ Offsets_7 \end{bmatrix} \tag{4.48}$$

Using the $t_{scriptNew_{FineCalib}}$ matrix, the $x_{scriptNew_{FineCalib}}$ matrix can be computed as follows:

$$x_{scriptNew_{FineCalib}} = A/t_{scriptNew_{FineCalib}} \tag{4.49}$$

Now comparing the x_{script} matrix with the $x_{scriptNew_{FineCalib}}$ matrix, the offset between them is very minimal in the range of '0.5 ps' compared to the offset before calculating the Multiplexer Offsets for each switch control iteration.

After arriving with the final values of $x_{scriptNew_{FineCalib}}$ matrix, we have the final values for static phase error for the respective delay cells of the delay chain.

$$\left[x_{scriptNew_{FineCalib}}\right] = \begin{bmatrix} Static\ Delay\ for\ Delay\ Cell\ 1 \\ Static\ Delay\ for\ Delay\ Cell\ 2 \\ Static\ Delay\ for\ Delay\ Cell\ 3 \\ Fine\ Calibration\ Delay\ Step\ for\ Delay\ Cell\ 1 \\ Fine\ Calibration\ Delay\ Step\ for\ Delay\ Cell\ 2 \\ Fine\ Calibration\ Delay\ Step\ for\ Delay\ Cell\ 3 \end{bmatrix} \tag{4.50}$$

4.2.7 Methodology to Choose the right Fine Calibration Settings:

This methodology is chosen to be the preferred way of estimating the right fine calibration settings for delay cells in the delay chain based on the Static Delays obtained for each delay cell in the $x_{scriptNew_{FineCalib}}$ matrix. For better understanding, the $x_{scriptNew_{FineCalib}}$ matrix can be rewritten with values such as:

$$\left[x_{scriptNew_{FineCalib}}\right] = \begin{bmatrix} 100.13\ ps \\ 97.68\ ps \\ 102.38\ ps \\ 0.3\ ps \\ 0.3\ ps \\ 0.3\ ps \end{bmatrix} \tag{4.51}$$

1. Firstly, out of the 6 values computed from the above matrix, the first 3 values (3*1) can be represented as follows:

$$\left[x_{scriptNew_{FineCalib_{1,2,3}}}\right] = \begin{bmatrix} 100.13\ ps \\ 97.68\ ps \\ 102.38\ ps \end{bmatrix} = \begin{bmatrix} DC\ 1 \\ DC\ 2 \\ DC\ 3 \end{bmatrix} \tag{4.52}$$

2. Next step is to find the Minimum, Maximum and middle values in the $x_{scriptNew_{FineCalib_{1,2,3}}}$ matrix.

   Here, Minimum = 97.68 ps,

Maximum = 102.38 ps,

Middle = 100.13 ps.

$$\begin{bmatrix} DC_{middle} \\ DC_{min} \\ DC_{max} \end{bmatrix} = \begin{bmatrix} 100.13\ ps \\ 97.68\ ps \\ 102.38\ ps \end{bmatrix} \tag{4.53}$$

3. Target variables are calculated for the minimum and the middle values as follows:

$$target_{min} = Maximum - Minimum \tag{4.54}$$

$$target_{mid} = Maximum - Middle \tag{4.55}$$

4. Firstly, if all the fine calibration states(Fine States) had a linear step delay, we can represent it as follows:

$$Fine\ States\ Delay_{Linear} = Fine\ States * 290.55^{-15} \tag{4.56}$$

Here, the compounded offsets are subtracted from the linear fine delay steps for all the 32 fine calibration states starting from the 1st fine calibration state to the 32nd fine calibration state.

$$Fine\ States\ Delay_{Real} = Fine\ States\ Delay_{Linear} - Compounded\ Offsets \tag{4.57}$$

5. To find the closest value from the $Fine\ States\ Delay_{Real}$ in comparison to the $target_{min}$ value, the minimum value in absolute terms is computed for the difference between the $target_{min}$ and $Fine\ States\ Delay_{Real}$ and substituted in the variable named 'temp' as shown below.

$$temp = abs(target_{min} - Fine\ States\ Delay_{Real})) \tag{4.58}$$

Substituting the temp value obtained in the equation below, we get:

$$[min_{temp}\ \ min_{temp_{idx}}] = min(temp) \tag{4.59}$$

Therefore, we have found the min_{temp} value as well as the index of the min_{temp} value from the $Fine\ States\ Delay_{Real}$ array.

6. If the $target_{min}$ value is lesser than the largest fine calibration step from the $Fine\ States\ Delay_{Real}$ array which is always the last value(31st value), then the Minimum value from the equation 4.52 is added along with the closest value obtained in 4.60.

$$closest = Fine\ States\ Delay_{Real}(temp == min_{temp}) \tag{4.60}$$

$$min_{num} = Minimum + closest \tag{4.61}$$

Otherwise, the Minimum value from the equation 4.52 is added along with the largest fine calibration step from the $Fine\ States\ Delay_{Real}$ array which is always the last value(31st value).

$$min_{num} = Minimum + Fine\ States\ Delay_{Real}(31) \tag{4.62}$$

7. Similarly, the difference in absolute values is computed between the $target_{mid}$ and $Fine\ States\ Delay_{Real}$.

$$temp_{mid} = abs(target_{mid} - Fine\ States\ Delay_{Real})) \tag{4.63}$$

Substituting the $temp_{mid}$ value obtained in the equation below, we get:

$$[mid_{temp}\ \ mid_{temp_{idx}}] = min(temp) \tag{4.64}$$

From the above equation, we have found the $temp_{mid}$ value as well as the index of the $temp_{mid}$ value from the $Fine\ States\ Delay_{Real}$ array.

8. The same procedure used in Step 6 is used here to compute the closest middle value from the $Fine\ States\ Delay_{Real}$ array. If the $target_{mid}$ value is lesser than the largest fine calibration step from the $Fine\ States\ Delay_{Real}$ array which is always the last value(31st value), then the Middle value from the equation 4.52 is added along with the closest value obtained in 4.60.

$$closest_{mid} = Fine\ States\ Delay_{Real}(temp_{mid} == mid_{temp}) \tag{4.65}$$

$$mid_{num} = Middle + closest_{mid} \tag{4.66}$$

Otherwise, the Middle value from the equation 4.52 is added along with the largest fine calibration step from the $Fine\ States\ Delay_{Real}$ array which is always the last value(31st value).

$$mid_{num} = Middle + Fine\ States\ Delay_{Real}(31) \tag{4.67}$$

9. Since the motive is to bring the relative delays between the delay cells of the delay chain close to each other, the maximum value of the equation 4.52 is not given any additional fine calibration setting, which is basically represented as '0' fine calibration state. Therefore,

$$[max_{temp}\ \ max_{temp_{idx}}] = [102.38ps\ \ 0] \tag{4.68}$$

By performing all the above steps to get the relative delays between the 3 delay cells as close as possible, we acquire the new values of Minimum, Middle and Maximum and sort them in the same array location as written in equation 4.52.

$$\left[x_{scriptNew_{FineCalib_{1,2,3}}}\right] = \begin{bmatrix} mid_{num} \\ min_{num} \\ max_{num} \end{bmatrix} = \begin{bmatrix} 102.33\ ps \\ 102.39\ ps \\ 102.38\ ps \end{bmatrix} = \begin{bmatrix} DC\ 1 \\ DC\ 2 \\ DC\ 3 \end{bmatrix} \tag{4.69}$$

Since the variables such as $min_{temp_{idx}}$, $mid_{temp_{idx}}$ and $max_{temp_{idx}}$ denote the fine calibration states used for the respective delay cells, we can write the matrix in the form as follows:

$$\begin{bmatrix} mid_{temp_{idx}} \\ min_{temp_{idx}} \\ max_{temp_{idx}} \end{bmatrix} = \begin{bmatrix} 9 \\ 28 \\ 0 \end{bmatrix} = \begin{bmatrix} DC\ 1 \\ DC\ 2 \\ DC\ 3 \end{bmatrix} \tag{4.70}$$

Therefore, the new $x_{scriptNew_{FineCalib}}$ matrix can be written as follows:

$$\begin{bmatrix} x_{scriptNew_{FineCalib}} \end{bmatrix} = \begin{bmatrix} 102.33\ ps \\ 102.39\ ps \\ 102.38\ ps \\ 0.3\ ps \\ 0.3\ ps \\ 0.3\ ps \end{bmatrix} \tag{4.71}$$

10. Using a similar approach for the delay chain-2, we get the final fine calibration states as written below:

$$\begin{bmatrix} mid_{temp_{idx}} \\ min_{temp_{idx}} \\ max_{temp_{idx}} \end{bmatrix} = \begin{bmatrix} 23 \\ 0 \\ 20 \end{bmatrix} = \begin{bmatrix} DC\ 1 \\ DC\ 2 \\ DC\ 3 \end{bmatrix} \tag{4.72}$$

Therefore, the new $x_{scriptNew_{FineCalib}}$ matrix for delay chain-2 can be written as follows:

$$\begin{bmatrix} x_{scriptNew_{FineCalib}} \end{bmatrix} = \begin{bmatrix} 102.8\ ps \\ 102.9\ ps \\ 102.8\ ps \\ 0.3\ ps \\ 0.3\ ps \\ 0.3\ ps \end{bmatrix} \tag{4.73}$$

Chapter 5

Circuit implementation

5.1 Delay SD self-calibration system

5.1.1 Delay Cell design

This section aims to make a comparison between the advantages and the disadvantages of the two main Delay Cell architectures that were taken into consideration for this Thesis: a True Single-Phased Clock (TSPC) delay cell design as explained in Section 5.2.2 and a Current Starved Delay Cell(CSDC) design. The advantages and disadvantages of each Delay Cell design are detailed in the paragraphs below.

CSDC vs TSPC

- The CSDC is implemented with the concept of tuning the delay by changing the amount of current needed to charge or discharge the capacitive load of the delay stage. The variation in current is controlled by the voltage applied as an input to the gate of NFET transistor with an active PFET load(the transistor is diode-connected and hence, operates in the active region) since the control voltage modulates the on-resistances of the NMOS and PMOS transistors through current mirrors which in turn changes the current flow and in turn, the delay of the delay stage. The current flow through the circuit increases or decreases in a linear fashion depending on the value of the control voltage, i.e. how high or low the value of the control voltage is.

  Transistors N8 and P7 act as current sink and current source respectively and transistors P6, N9 act as inverters. The drain current flowing through transistors P9 and N7 is controlled by the NFET coarse voltage, which in turn is mirrored to the current source transistor P7.

- Final performance metrics of the CSDC in comparison with TSPC Delay Cell(Phase Noise Performance and Power Consumption): Comparisons between both versions of the Delay Cell were made by performing simulations with Fine calibration settings but for a non-ring-oscillator setup. For the CSDC architecture, the fine calibration settings are set for 64 calibration states since the resolution chosen is 1 bit higher than the TSPC Delay Cell architecture. PSS + PNoise simulation results showed that the Phase Noise

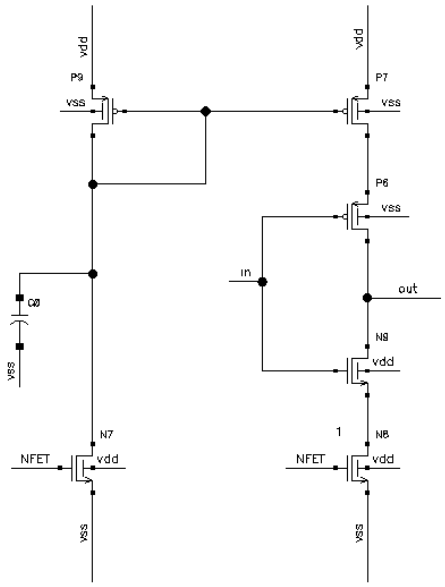

Figure 5.1: Current Starved Delay Cell

performance of the TSPC was comparatively better than the CSDC architecture. Similarly, transient simulations were performed to compute the power consumption for the entire delay cell structure without external blocks such as CP, Edge Combining circuit and PD.

The DC current generated by the biasing transistors could not be reduced beyond a point since it would drastically affect the performance of the delay cell in terms of the delay acquired since reducing the current flow would in turn increase the delay which would contradict the purpose of solving the problem.

Also, another problem with CSDC architecture is that it required an additional delay cell, i.e., 9 delay cells(odd number of delay cells), one more delay cell apart from the existing number of delay cells(8 delay cells comprising 6 delay cells and 2 delay cells - Input Buffer and Output Buffer) when run in a ring-oscillator setup since the oscillation could not be triggered when even number of delay cells were used. On the other side, the performance metrics obtained for the TSPC delay cell were comparatively better than the ones obtained for the CSDC architecture. Therefore, it is because of the above mentioned trade-offs that the TSPC delay cell is the chosen architecture for the delay cell.

TSPC

The custom designed TSPC D Flip-Flop is the building block of the Delay Cells used in the VCDL block of the Frequency Synthesizer. The TSPC design is optimised to replicate the output behaviour of a typical D Flip-Flop cell, using the minimal number of transistors possible. whilst maintaining the stability of a CMOS

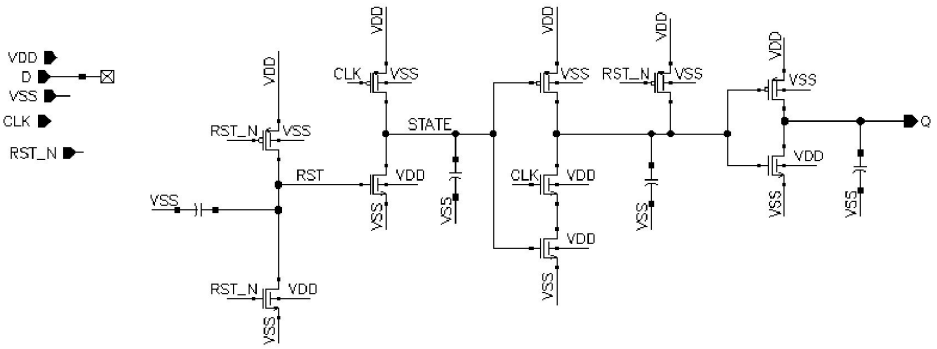

Figure 5.2: TSPC cell schematic

logic structure. As a result, two transistors have a critical role in the generation of the output waveform: the PMOS device in the second stage of the cell, controlled by a CLK signal, and the PMOS transistor in the forth stage, controlled by the RST_N signal.

Thus, the output waveform of the TSPC in the synthesizer presented in this thesis is as follows: the rising edge is generated on the rising edge of the clock signal, while the falling edge appears on the falling edge of the reset signal.

Lastly. by using the output of the TSPC as a delayed input for the RST_N pin, a latching behaviour is obtained, where the pulse width of the TSPC output is controlled by the delay added by a current-staved inverter, which will be expanded upon in Subsection 5.1.4

5.1.2 Switch design

According to the mathematical proof shown in chapter 5, the system of equations structure needed for the calibration procedure to be successful needs the implementation of signal switches throughout the VCDL path, similar to Figure 5.3. Moreover, the switches employed need to be designed in a way which results in as small of an added relative delay to the signal path as possible when choosing to short-circuit one of the delay cells in the VCDL.

Three main switching approaches were considered during the design phase: a T-switch setup, a 2-input-1-output multiplier, and a custom Delay Stage.

The T-switch approach, observable in Figure 5.4, is made out of 2 transmission gates united through a pull-down NMOS which drains any floating voltages during the OFF state of the switch.

The setup for this method is represented by a customized Delay Stage which contains a pull-up PMOS with a tandem pull-down NMOS device between the supply and the rest of the circuit. Moreover, a T-switch component is added between the input and the output of a Delay Stage, which works in tandem with the previous custom mechanism described previously. During normal VCDL operation, the Delay Stage is coupled to the supply through the PMOS device, and the

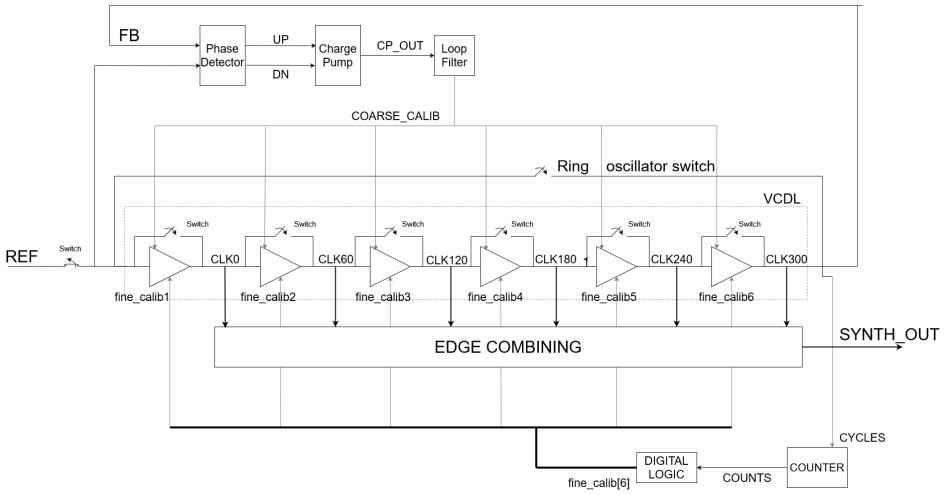

Figure 5.3: Block diagram of the modified VCDL with switches

T-switch is turned OFF. Meanwhile, for specific iterations of the self-calibration routine, the Delay stage is decoupled from the supply voltage, and the pull-down NMOS device is turned ON to drain any floating voltages inside the component. At the same time, the T-switch is turned ON, letting the RF signal bypass the current Delay Stage, whilst adding minimal delay to the desired clock signal.

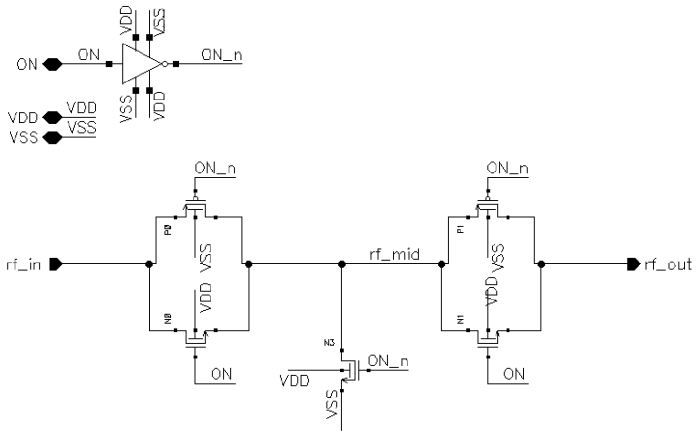

Figure 5.4: Improved switch: T-switch structure

The main disadvantages of this approach are that the added T-switch adds an impedance to the ground during the OFF state, which creates a voltage division at the output of the Delay Stage component, resulting in a significant voltage drop which could affect the output of the Frequency Tripler.

The Multiplier approach consists of a custom circuit, observable in Figure 5.5,

made up of a combination of two T-switches, working in tandem, which has a voltage buffer at their combined output, made of two chained inverters. One T-switch is connected to the output of a Delay Stage, while another one is connected to the input of it. The two switches are managed through a control signal denoted as "SEL", which turns either of the two switches ON or OFF so that only one path is let through to their combined output.

The advantages of this setup are that the voltage division happening at the combined output of the two T-switches is corrected by the voltage buffer on the output of the multiplier and that each of the two inputs is loaded equally by the multiplier, adding the same static delay both during VCDL and calibration operations. Disadvantages of this setup, compared with the other two alternatives, are that the power consumption during operation at 3.33 GHz is higher and that it adds more delay to the clock signal path, which can lead to an output frequency from the EC circuit of less than triple the input frequency.

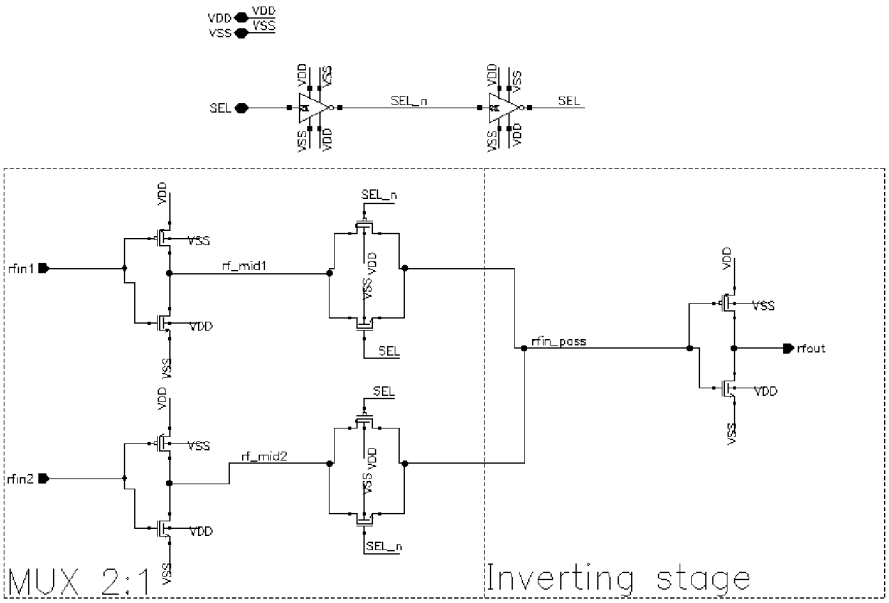

Figure 5.5: Multiplexer setup for the RF signal

The custom Delay Stage approach is represented by combining aspects of the previous two approaches. As the TSPC D Flip-Flop contains an inverter at the output of the cell, a similar, but smaller-sized inverter is added between the input and the output of the TSPC, equipped with a PMOS switch between the cell and the supply voltage and an NMOS switch to ground. A similar set of pull-up and pull-down switches are added to the supply and ground connection of the TSPC cell. Both of these sets of switches are controlled through a control signal denoted as "BYPASS", which ensures the two systems work in tandem. During VCDL operation, the RF signal passes through the TSPC D Flip-Flop, and during

calibration, the Delay Cell can be disconnected from supply and ground while the short-circuiting inverter is connected to them, ensuring that the signal has a minimal added delay.

The disadvantages of this method are that during calibration, the short-circuiting inverter adds a certain amount of delay that is not present during VCDL operation. However, this delay can be taken into consideration as a static offset during the calculation of proper calibration settings of the final product. The advantages are reduced power consumption and lack of voltage division at the output of the cell.

5.1.3 Delay Stage fine calibration

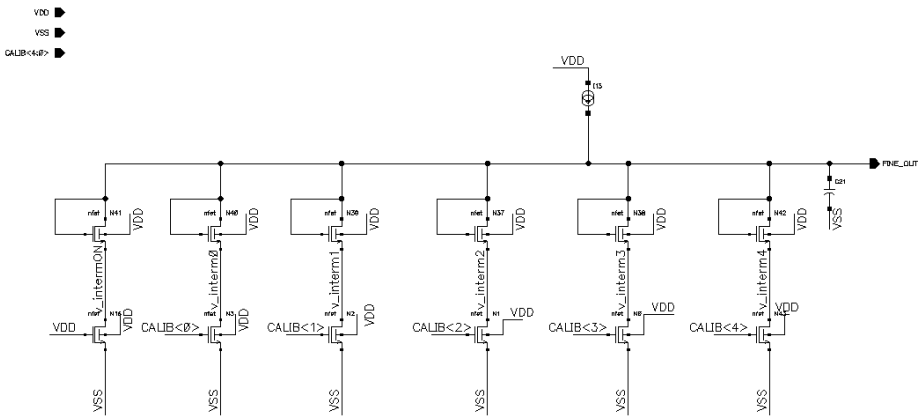

Figure 5.6: Fine calibration circuit

Besides the coarse calibration provided by the DLL components of the circuit, a second fine calibration system needs to be added for the correct VCDL calibration. This is both for covering for the limited precision the coarse calibration of the DLL PD and CP provide, as well as to enable the correction of individual mismatches that can appear between Delay Stages, during the fabrication of the circuit.

The fine calibration mechanism employs a separate fine calibration circuit which can be observed in Figure 5.6. The circuit is made of 5 binary weighted transistors that ca be turned ON by individual control signals "CALIB 0 to 4", which give a number of 32 individual fine calibration states within the dynamic range of the component. The range can be expanded or contracted through the biasing current which is being supplied from a separate part of the receiver. The current affects the amplitude of the fine calibration step unit, where a higher current results in a bigger unit step, but also in less precision for mismatch correction.

In addition, an extra transistor with a similar size to the fourth binary weighted one in the component is always kept turned ON in order to reduce the ripples

caused by rapid changes between extremes of the fine calibration range, which could lead to unwanted behaviour in the Delay Stage it is connected to. Each Delay Stage has an individual fine calibration circuit attached to it for identifying the nominal mismatch correction settings during VCDL calibration. Moreover, the structure of the variable delay inverter in the Delay Stage block has been modified to accommodate the new fine delay calibration by adding a parallel branch equipped with a smaller NMOS transistor which supplies additional current to the inverter, according to the variable voltage supplied by the fine calibration circuit. An image of the modified block is presented in Figure 5.7

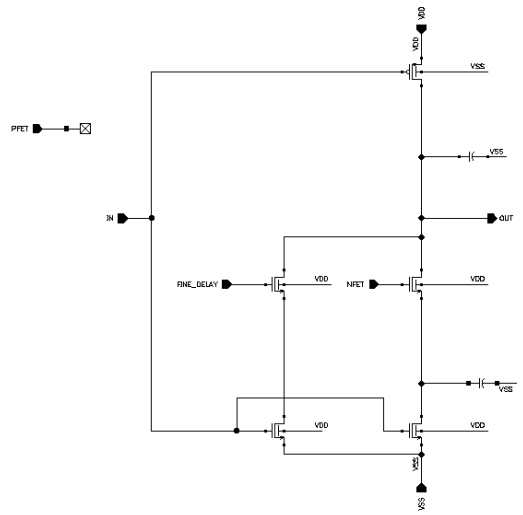

Figure 5.7: Delay Cell modified for both coarse and fine delay calibration

5.1.4 Testbench Setup for simulating mismatches

The $cycleTime_k$ obtained for delay chains with mismatches differs from the ideal 300 ps since, they would have a different static phase delay compared to the other delay cells without mismatch, making the relative static phase delay different for the delay cells in the delay chain. This difference is translated in the form of spurious tones in the Frequency spectrum at the edge-combined output.

Knowing the problems associated with the 60° Phase Difference Delay Cells as mentioned in section 6.2.2, the test bench setup used for simulating mismatches consists of all the delay cells with a phase difference of 120° but along with the mismatches in the form of parasitic capacitors to make the test bench setup more non-ideal. Since we have the setup of 3 delay cells, we combine the 0° Delay Cell, 120° Delay Cell, and the 240° Delay Cells together in a delay chain. As we use the differential input for the delay chains, the 180° Delay Cell, 300° Delay Cell, and the 60° Delay Cells are combined together in another delay chain.

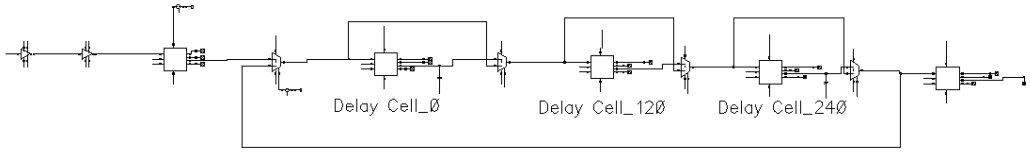

Figure 5.8: Delay-Chain 1

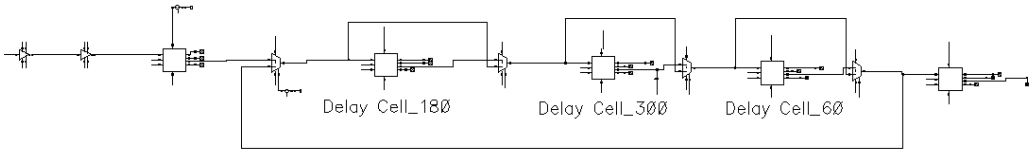

Figure 5.9: Delay-Chain 2

As we see from the above Figure, the 0° Delay Cell and the 240° Delay Cell of the first delay chain have mismatches in the negative output of the delay cells in the form of parasitic capacitors whose values are $2.657fF$ and $4.975fF$ respectively. Whereas in the second delay chain, it's the 300° Delay Cell which has the mismatch in its negative output in the form of PDK capacitor whose value is 3.7294 fF.

5.2 Static Phase Error self-calibration system

5.2.1 DTC

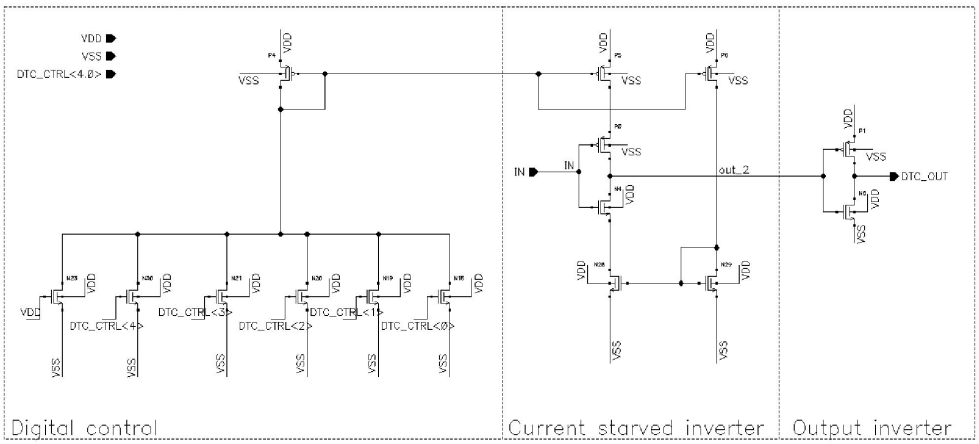

Figure 5.10: Digital-to-Time Converter schematic

The DTC circuit receives a set of five digital control signals "$DTC\_CTRL(4-0)$" which limit the amount of current being drawn by the pull-down and the pull-

up network of a voltage buffer circuit, translating into a certain amount of delay offset to the input signal waveforms. The general schematic of the DTC can be observed in Figure 5.10.

The current starving element of the circuit is composed of five binary-weighted NMOS transistors, together with an extra NMOS transistor which ensures more linear behaviour for the transfer function of the component. Each NMOS device is controlled by a control signal, where the LSB is controlled by $DTC\_CTRL(0)$ and the MSB, which is $2^N = 16$ times larger than the LSB, controlled by $DTC\_CTRL(4)$. The additional NMOS device is sized to be 4 times larger than the LSB, ensuring the maximum difference during operation between digital states is from $25\%I_0$ to $400\%I_0$, where I_0 is the current being drawn by the additional transistor at all times,

The output of the binary-weighted transistors is then replicated to the pull-up and the pull-down network of an inverter using a $1 : 0.7$ PMOS current mirror and a $1 : 0.8$ NMOS current mirror to ensure the desired current starving behaviour. Furthermore, the ratio between the gate widths of NMOS and PMOS devices in the inverters was kept at 2.2 times. This ensures that the rising edge and the falling edge of the input signal are being delayed by the same amount and in a symmetrical fashion, to minimize sources of errors in the Frequency Synthesizer's signal path. The last inverter is sized to provide enough driving strength for the upcoming blocks of the circuit, as well as to ensure that there are no polarity changes between the input and the output signal.

The sizing of the transistor has been achieved using multipliers instead of modifying their gate widths and the width has been divided into groups of 4 or 8 fingers for better linearity and less sensitivity to mismatch effects following manufacturing.

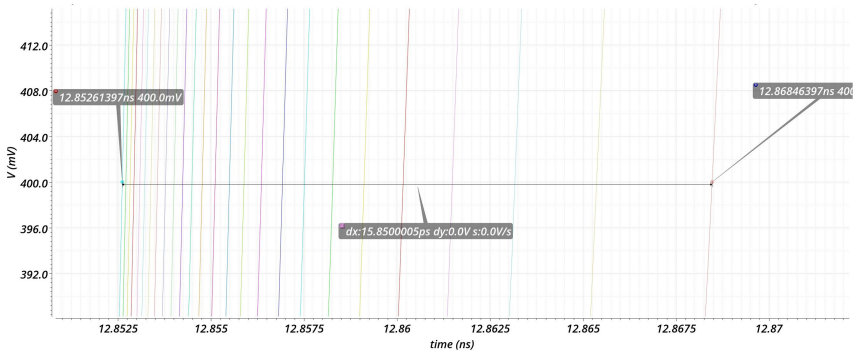

Figure 5.11: DTC Input-Output characteristic

There are two versions of the DTC circuit in use inside the SPE self-calibration system. The DTCs employed on the input of the DLL PD component have a shorter-sized first inverter, leading to the dynamic range of $4.3ps$, while the ones in the Delay Sensing Circuit have a dynamic range of $18ps$ to prevent improper detection of proper compensation for internal mismatches in the DSC. This choice is also motivated by the use of an inverter device on the "UP" input path of the DSC CP block, which creates an imbalance between the two CP inputs even when

the input signals are characterized by equal pulse widths.

The input-output characteristic of a DTC device can be observed in Figure 5.11, where it can be seen that the delay offset added by the DTC to the input signal's rising edge does not evolve in a linear fashion with the digital states represented. The "jump" between different states is not consistent, because the gate voltage mirrored to the current-starving elements of the DTC circuit forces the transistors to enter into the linear region for some of the states, resulting in bigger differences between states compared with their behaviour when in saturation.

This behaviour is an inherent characteristic of transistor operation and, as such, is taken into account by limiting the number of states used during SPE calibration from 32 theoretical calibration states / DTC to 22 actual states.

5.2.2 Phase Detector

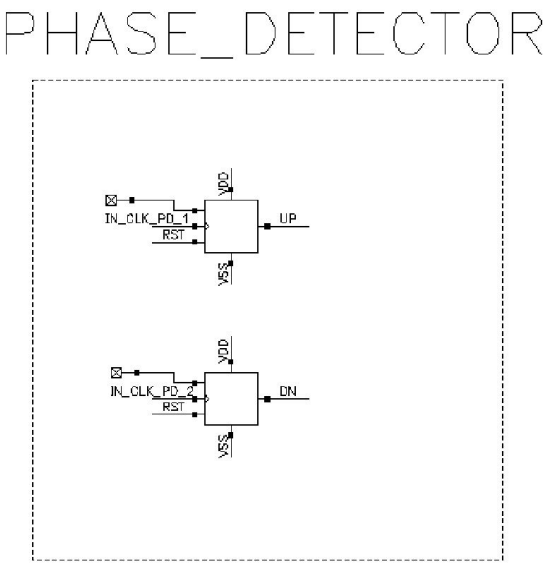

Figure 5.12: PD

The DSC's Phase Detector (PD) circuit, observable in 5.12, works by utilising two TSPC D Flip-Flop devices. The output of each TSPC cell produces a waveform with a pulse width determined by the rising edge of its input signal, which is then used to regulate parts of the calibration system necessary for synchronising the main "REF" and "FB" signals.

The reset signal used for this PD architecture is taken from the negative output of the second to last Delay Cell of the VCDL block, because that signal is equidistant in phase offset from both the reference signal and the signal generated at the output of the VCDL block (at 300^o offset from CLK_0 and CLK_360). Using this identical reset signal on both cells of the PD, the falling edge of the outputs of the PD will happen simultaneously.

The TSPC cell architecture has been employed because of the lower power consumption and reduced number of transistors employed. The architecture can be observed in Figure 5.2, from Section 5.1. Moreover, this type of PD circuit used in the Delay Sensing Circuit is the same as the one utilised in the main DLL loop, due to its reduced power consumption.

5.2.3 Charge Pump

The Charge Pump (CP) shown in Figure 5.13 below is supposed to deliver a charge proportional to the phase error to the loop filter. Phase error refers to the error in phase between the two outputs of the PD which are given as inputs to the CP.

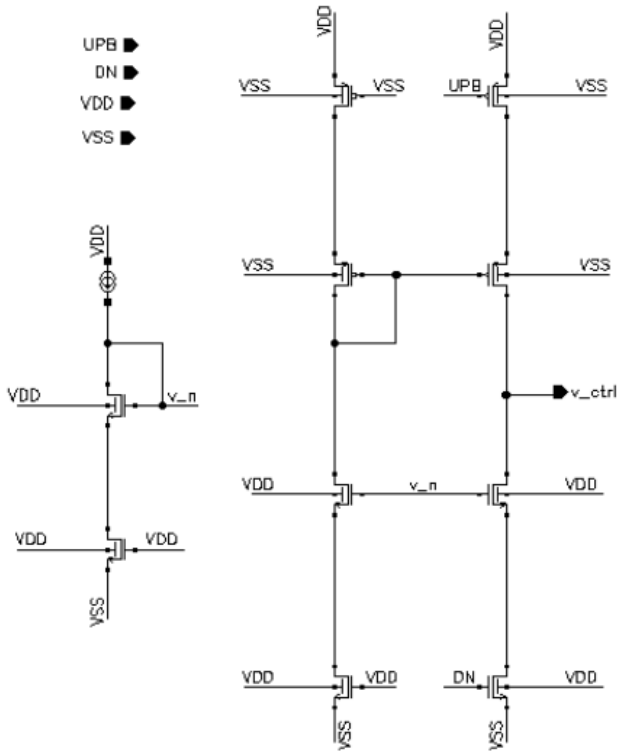

Figure 5.13: CP

The PFET transistor and the NFET transistor in the left branch of the CP act as the current source and current sink respectively since they are always turned ON. A current mirror is used to bias the NFET transistors and the PFET transistors, named by (v_n).

A single-ended version of the charge pump is shown in the above Figure 5.13, where UPB and DN are the input signals applied to the gate terminals of the PFET and the NFET transistors respectively. Transistors are sized in order to obtain a steady state output voltage of $V_{DD}/2$. However, based on the pulse width

of the UPB and DN signals, the OFF-time of the PFET and the ON-time of the NFET transistors vary based on which the current is pulled towards V_{DD} by the PFET transistor or pushed towards GND by the NFET transistor. The output voltage of the CP increases or decreases based on how long the UPB signal is than the DN signal, or vice versa.

This output voltage is then fed to the loop filter which contains a low-pass filter that will convert a series of pulses into a voltage proportional to the average current.

5.2.4 Voltage-Controlled Oscillator

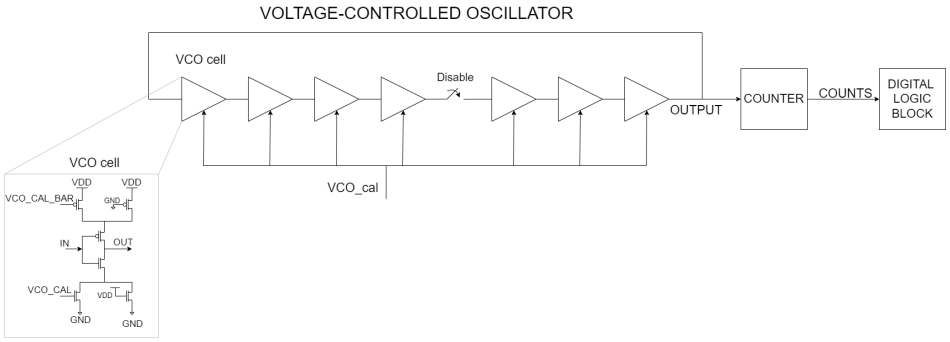

Figure 5.14: DSC Voltage-Controlled Oscillator block diagram

The DSC's Voltage-Controlled Oscillator (VCO) circuit, observable in Figure 5.14 is constructed using custom-designed current-starved inverter cells, designed for a high-frequency dynamic range for a unit of voltage change in the "VCO_CAL" signal. The VCO chain is constructed out of an odd number of custom inverter cells to ensure oscillations from the ring oscillator circuit.

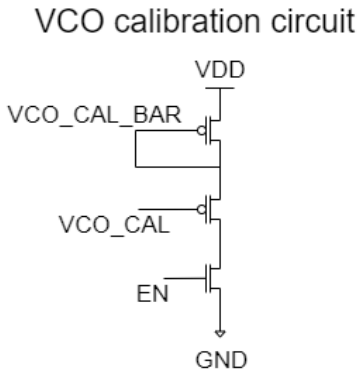

Figure 5.15: VCO calibration circuit block diagram

The custom inverter cell which can be viewed in Figure 5.14 is constructed with two sets of current-controlling transistors in both the pull-up and pull-down network. In each network, one of the transistors is always turned ON, however, it has a reduced size to ensure maximum transition time between "VDD" and "GND" on the output of the cells. The other transistor is controlled by the "VCO_CAL" signal, respectively by the "VCO_CAL_BAR" for the PMOS device, which is provided from the DSC's CP block. To generate the proper biasing for the pull-up network of all VCO cells, a simple calibration voltage has been employed, similar to the one presented in Figure 5.15.

The final dynamic range of the Voltage-Controlled Oscillator is of "626 MHz" between the amplitudes of $300mV$ to $500mV$ in the VCO_CAL calibration signal. Moreover, the whole VCO block can be disabled after the Delay Sensing Circuit has identified to correct DTC states in order to save on power consumption for normal receiver operation.

Chapter 6

Results and discussion

6.1 Results

The SCR results obtained for simulations performed with and without fine calibration and DTC calibration settings for the best case and the worst case scenarios are discussed briefly in this section.

6.1.1 Best Case vs Worst Case Scenarios for the SCR Computation:

SCR for the Best Case Scenario:

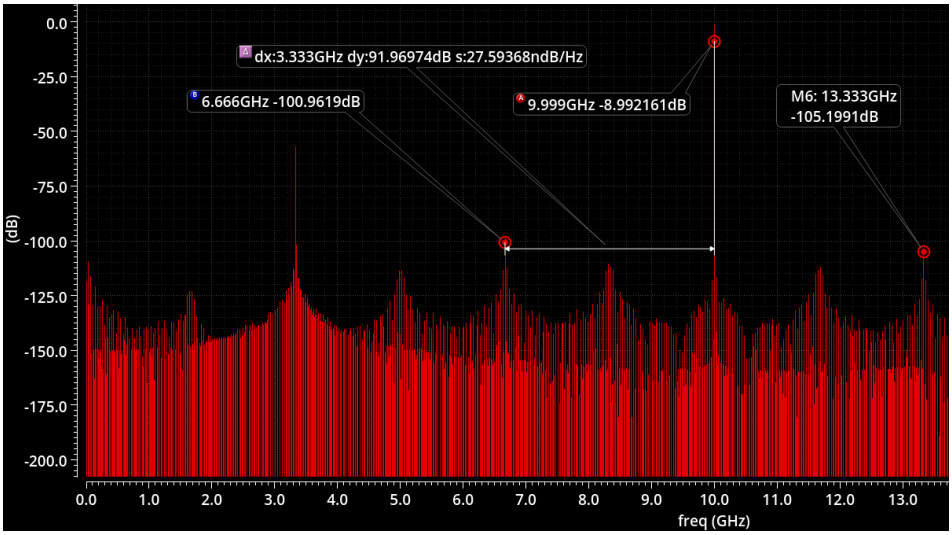

Figure 6.1: Best Case Scenario 1: FFT Plot with no mismatches in the Delay Chains

There are two best case scenarios in the SCR computation out of which one such scenario is when there are no mismatches from both the delay chains, the EC circuit basically does not see any difference in terms of the output signals of the delay cells between both the delay chains in which, the delay SD is approximately

negligible. This can be explained by considering a scenario when there are no mismatches arising from the delay cells in both the delay chains, the EC circuit which combines the outputs of 3 delay cells in both the delay chains does not see a difference in terms of the rising edge or the falling edge between signals from the 2 delay chains.

Since the EC circuit produces differential outputs at the end which are the same, the FFT output produced has spurious tones(6.666 GHz and 13.333 GHz) at amplitude levels of -100.961 dB and -105.199 dB respectively as shown in Figure 6.1. As we see from the Figure 6.1, the $SCR = -91.969dBc$, which turns out to be one of the best case scenarios for the SCR computation. Other scenario is when the delay chains have minute differences in the mismatches in terms of parasitic capacitances for which the FFT output is shown in Figure 6.2.

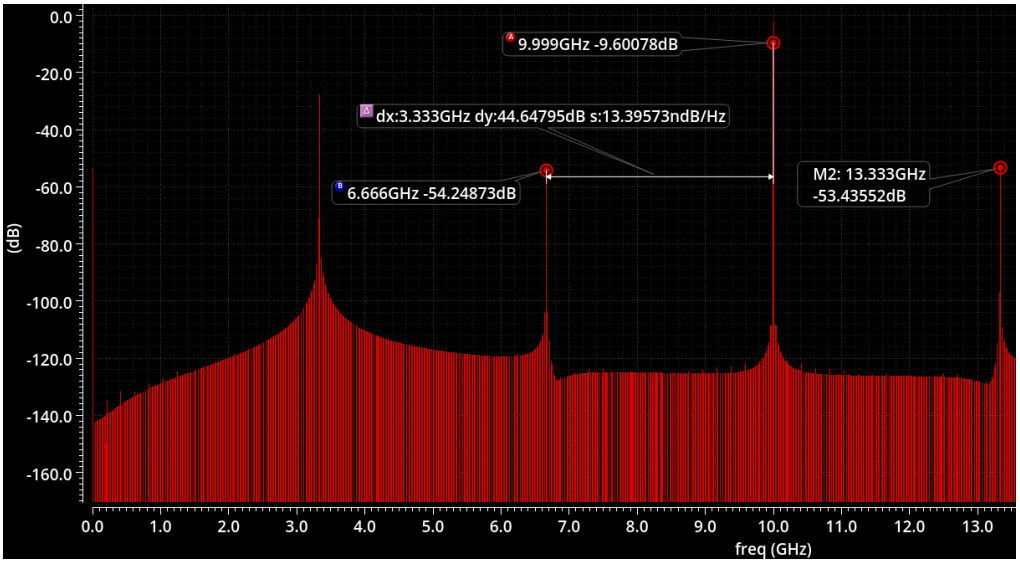

Figure 6.2: Best Case Scenario 2: FFT Plot with minute mismatch differences in the Delay Chains before Fine calibration

SCR for the Worst Case Scenario:

From the above explanations, it suggests us that the spurs originate predominantly only if the mismatches of the delay chains are different, thereby, the differential output signals of the edge-combining circuit have a frequency peak-peak of a higher value in MHz. The effect of mismatches in the delay cells is carried over to the succeeding delay cell as the negative output of the preceding delay cell with the mismatch is sent as an input to the next delay cell. Therefore, if the mismatches from the delay chains are different, the FFT output produced contains spurious tones in the even-order harmonic frequencies of the fundamental frequency, which is explained in the upcoming sections.

6.1.2 Initial SPE and delay SD pre-calibration

Delay SD before Fine Calibration:

As mentioned in Section 5.1.4, the initial mismatch offsets at the rising edges of the (0°, 120°, 240°) signals between delay cells without mismatch and (0°, 120°, 240°) with mismatches before any fine calibration settings is shown in the Figure 6.3 below. The value of 4.907 ps in the Figure 6.3 represents the maximum Delay SD for which we use the Fine Calibration settings to minimise the delay SD.

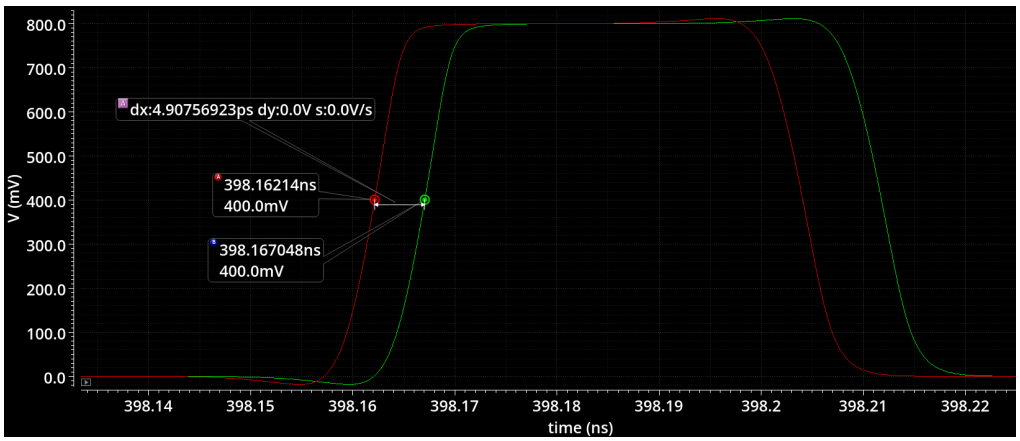

Figure 6.3: Delay SD pre-Fine Calibration

As shown in the equation 4.52, the static phase delays for (0° Delay Cell, 120° Delay Cell and 240° Delay Cell) before Fine Calibration settings for the delay cells with mismatch errors are written below:

$$\left[x_{scriptNew_{FineCalib_{1,2,3}}}\right] = \begin{bmatrix} 100.13\ ps \\ 97.68\ ps \\ 102.38\ ps \end{bmatrix} = \begin{bmatrix} DC\ 1 \\ DC\ 2 \\ DC\ 3 \end{bmatrix} \tag{6.1}$$

SPE before DTC Calibration:

The initial SPE before the DTC calibration is measured to be 4.564 ps between the rising edges of the REF and FB as shown in the Figure 6.4 below.

6.1.3 Statistical SCR estimation based on delay SD and SPE values

For mismatch errors in delay cells before the use of Fine calibration settings,the SPE and Delay SD values obtained from the simulations are inserted in the Python Script to estimate the Mean SCR in dBc. By substituting these values with a negligible effect of the DCD, the script estimates the SCR(dBc) to be equal to -22.470 dBc.

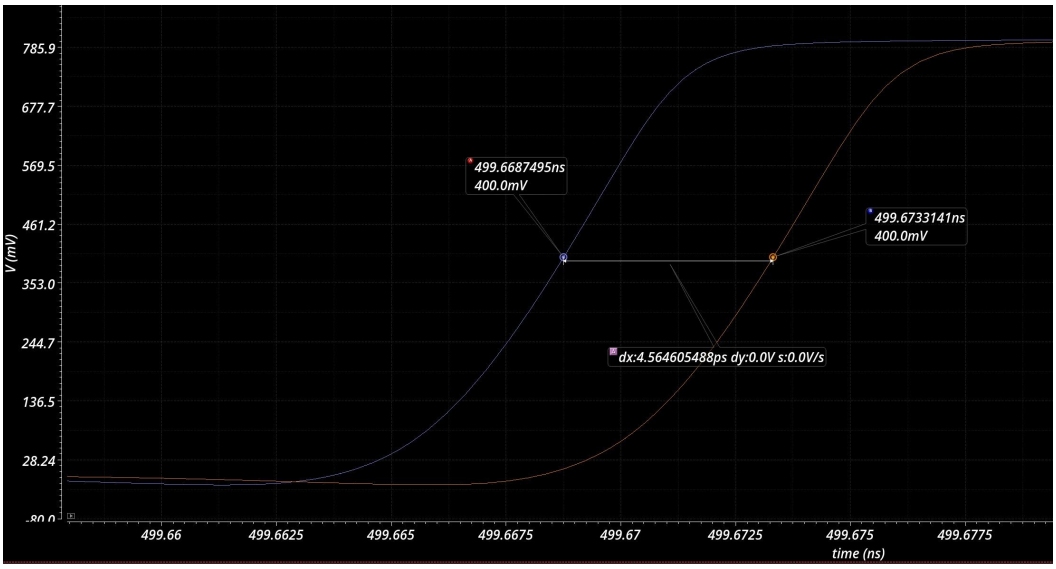

Figure 6.4: SPE pre-calibration

For $T_{SPE} = 4.564ps$, $Delay\ SD = 4.907ps$, and a negligible $T_{DCD} = 10^{-27}ps$,

$$Initial\ Mean\ SCR(dBc) = -22.4707 \tag{6.2}$$

It can be seen that the SCR value estimated by the script is closer to the value generated by the transient simulation results as shown in the Figure 6.10.

6.1.4 Final delay SD and SPE post-calibration

Delay SD before Fine Calibration:

The static phase delays for (0° Delay Cell, 120° Delay Cell and 240° Delay Cell) after Fine Calibration settings for the delay cells with mismatch errors are written below:

$$\left[x_{scriptNew_{FineCalib_{1,2,3}}}\right] = \begin{bmatrix} 100.16\ ps \\ 99.46\ ps \\ 100.38\ ps \end{bmatrix} = \begin{bmatrix} DC\ 1 \\ DC\ 2 \\ DC\ 3 \end{bmatrix} \tag{6.3}$$

Comparing values from equation 6.1 with the above equation 6.3, we notice that the relative static phase delay between the delay cells for the setup with mismatch errors after implementing the Fine calibration mechanism are closer than before Fine Calibration settings are used.

The Delay SD is reduced by 5.66 times after fine calibration from 4.907 ps as shown in Figure 6.3 to 86.610 fs as shown in Figure 6.5. This is mainly because of the fine calibration settings used in the second delay cell(Fine Calibration: '28')

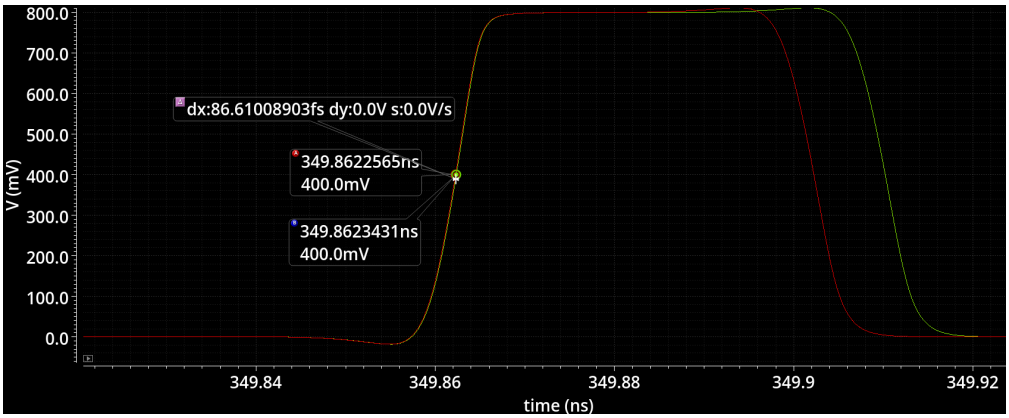

Figure 6.5: Delay SD post Fine Calibration

as mentioned in the Chapter 4 in equation 4.70. The effect of the setting used for Delay Cell 2 is carried over to the Delay cell 3, which in turn minimises the Delay SD to 86.610 fs at the rising edge of the signal.

SPE after DSC calibration:

The final SPE after the DTC calibration is measured to be 1.6 ps between the rising edges of the REF and FB.

6.1.5 SCR improvement impact from SPE only calibration

The FFT generated for the SPE effect before any DTC calibration added is shown in the Figure 6.6 below.

The SCR value as seen in the Figure 6.6 is written as:

$$SCR = -41.331\ dBc \tag{6.4}$$

After the SPE calibration sequence is run, the following set of DTC states are obtained to correct the effect of the PD mismatches:

- DTC_DSC_FB has been modified from state 31 to state 7
- DTC_REF for the first delay chain has been modified from state 31 to state 6
- DTC_REF for the second delay chain has been modified from state 31 to state 10

Thus, a total of 26 + 27 + 23 = 76 SPE calibration iterations were necessary to identify the final calibration settings, according to the flowchart presented in Figure 4.2 from Section 4.1.

The FFT generated for the SPE effect after DTC calibration added is shown in the Figure 6.7.

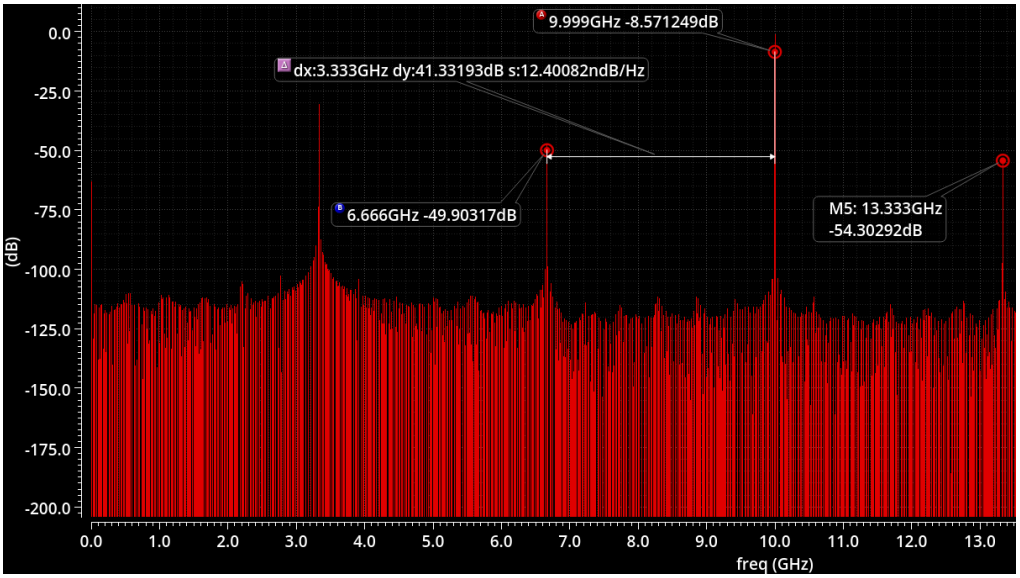

Figure 6.6: SPE only FFT Plot before DTC calibration

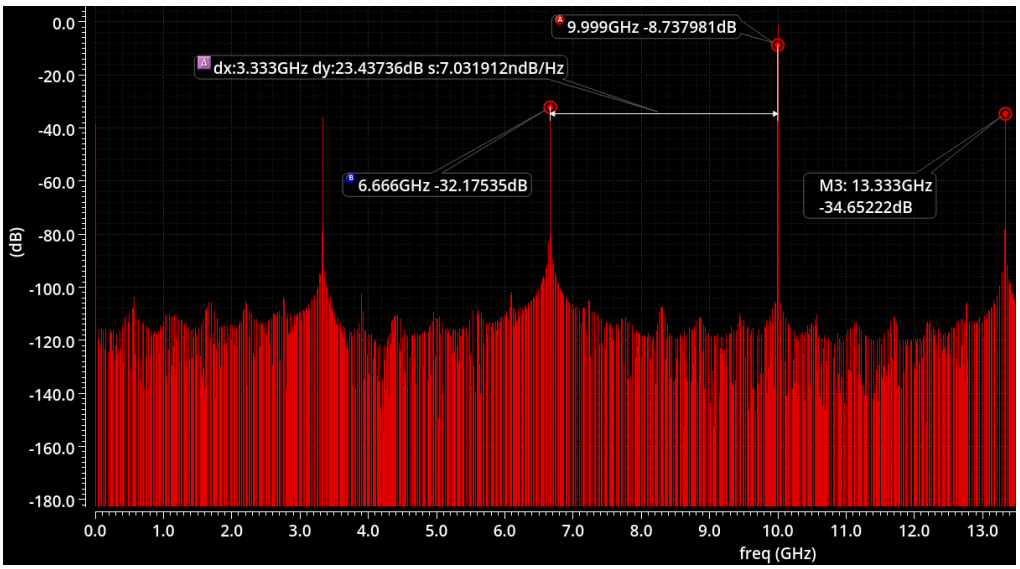

Figure 6.7: SPE only FFT Plot after DTC calibration

The SCR value as seen in the Figure 6.7 is written as:

$$SCR = -23.437\ dBc \tag{6.5}$$

6.1.6 SCR improvement impact from delay SD only calibration

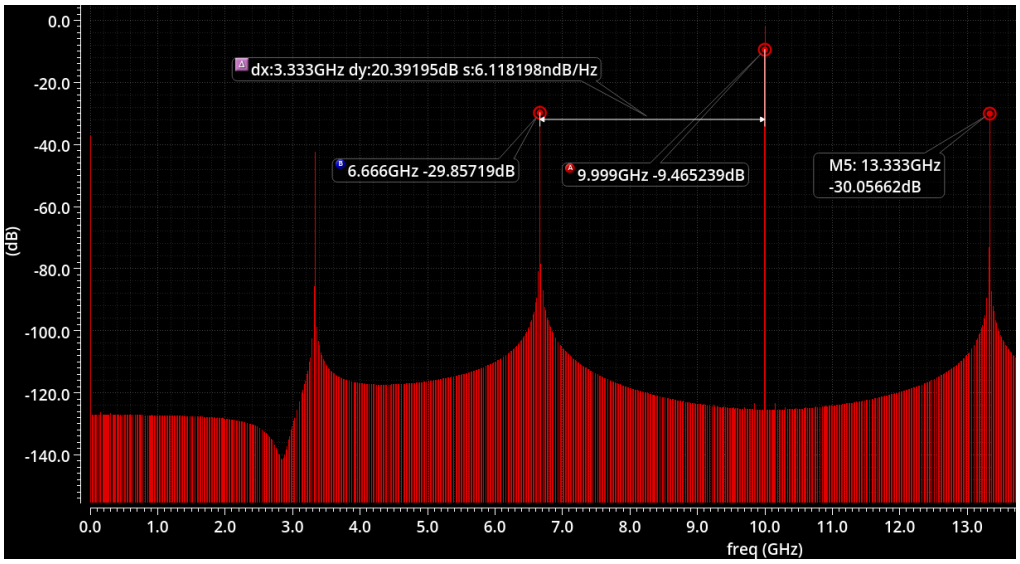

Figure 6.8: Delay SD only FFT Plot before Fine calibration

The FFT generated for the delay SD effect before any fine calibration added is shown in the Figure 6.8 below.

The SCR value as seen in the Figure 6.8 is written as:

$$SCR = -20.391\ dBc \tag{6.6}$$

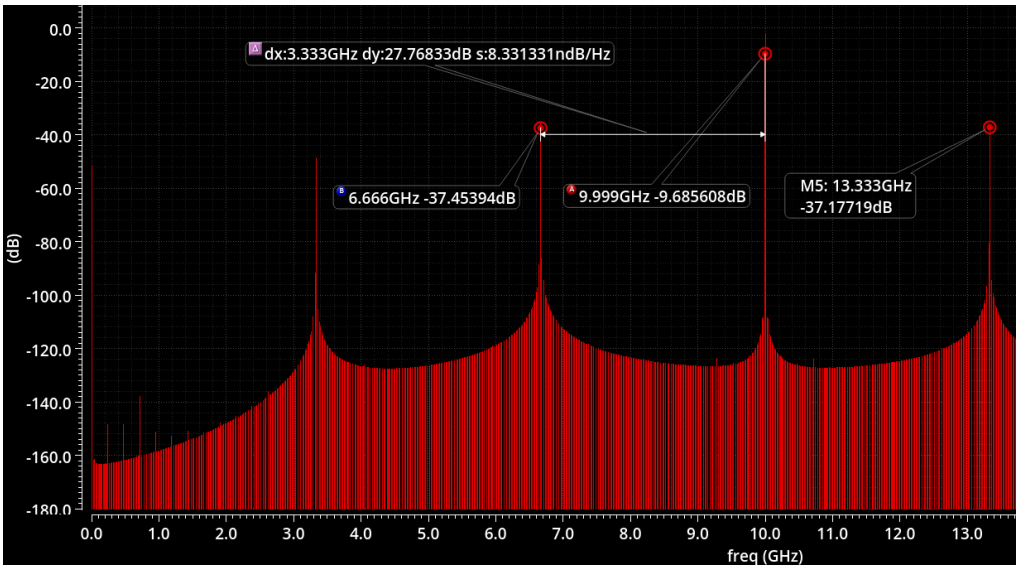

Figure 6.9: Delay SD only FFT Plot after Fine calibration

The FFT generated for the delay SD effect after any fine calibration added is shown in the Figure 6.9 below.

The SCR value as seen in the Figure 6.9 is written as:

$$SCR = -27.7568\ dBc \tag{6.7}$$

As we observe from the equations 6.6 and 6.7, there is a 7.369 dBc improvement in the SCR after the fine calibration mechanism is implemented to suppress the level of the spurious tones.

6.1.7 Combined SCR improvement for both mismatch sources

The FFT generated for the effect of Delay SD and SPE before any fine calibration added is shown in the Figure 6.10.

The SCR value as seen in the Figure 6.10 is written as:

$$SCR = -20.274\ dBc \tag{6.8}$$

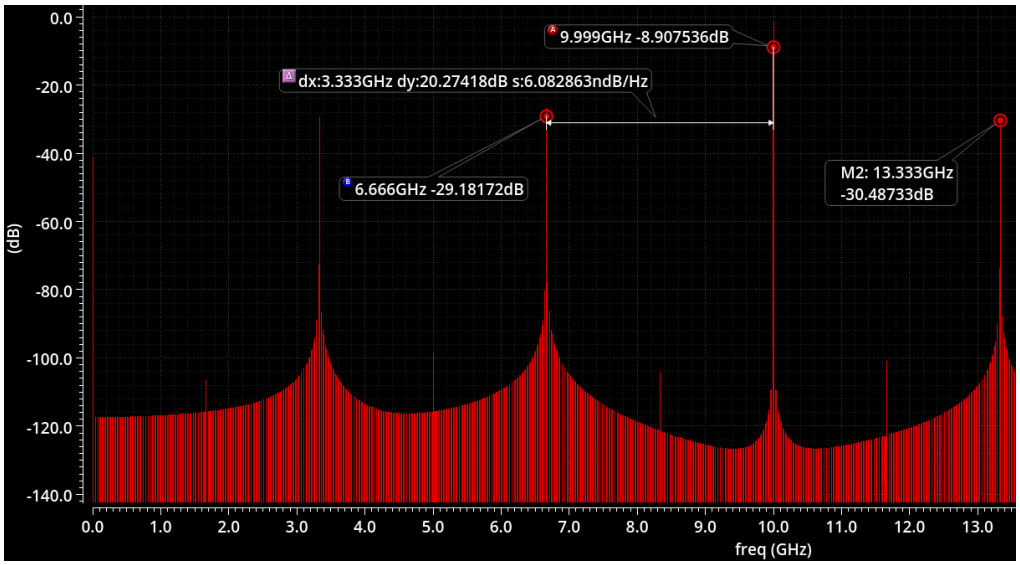

Figure 6.10: SPE and Delay SD FFT Plot before Fine calibration

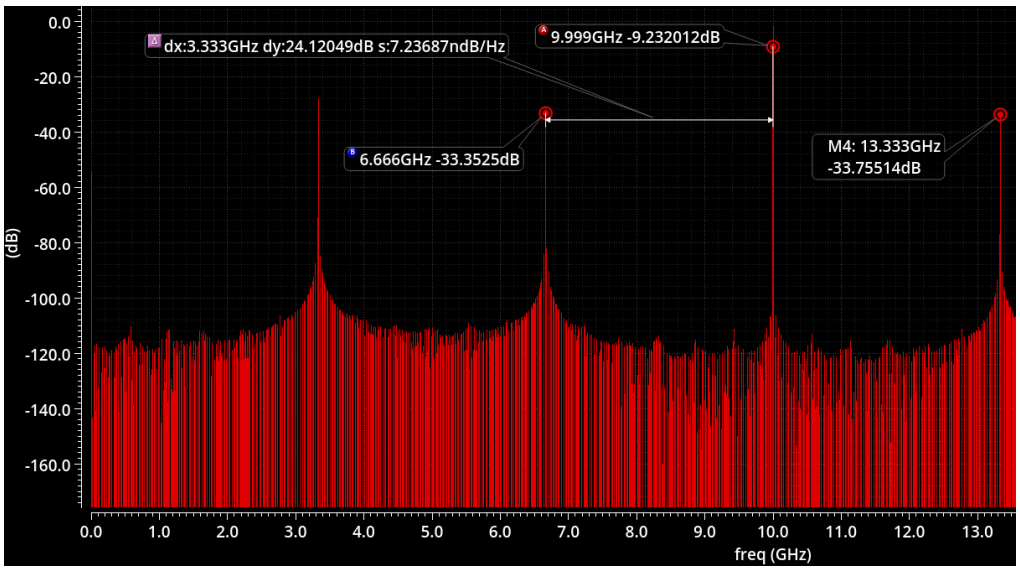

Figure 6.11: SPE and Delay SD FFT Plot after Fine calibration

The SCR value as seen in the Figure 6.11 is written as:

$$SCR = -24.120\ dBc \tag{6.9}$$

6.1.8 Total calibration time for the procedure

The total calibration time for the EC DLL circuit is divided between the following :

- $t_{dll} = 300$ ns, the settling time for the DLL feedback loop
- $t_{iter} = 300$ ns, the calibration time per iteration of the DSC, for each VCDL in the final circuit
- $t_{reset} = 100$ ns, the settling time for resetting the voltage across the VCO capacitor in the DSC
- $t_{vcdl} = 9 * 3.99\mu s = 35.91$ us, the total calibration time for entire delay SD calibration procedure

According to the way the calibration algorithms work and the values specified in the other subsections of this chapter, a total of 76 iterations are necessary for SPE calibration, meaning 76 resetting intervals for the VCO capacitor and 3 settling time intervals for the DLL feedback loop. Thus, the final calibration time can be calculated as:

$$t_{total} = 3t_{dll} + 76t_{iter} + 76t_{reset} + t_{vcdl} \tag{6.10}$$

This amount to a value of $67.21\mu s$ for a complete calibration procedure.

6.1.9 SCR improvement for Best case scenario

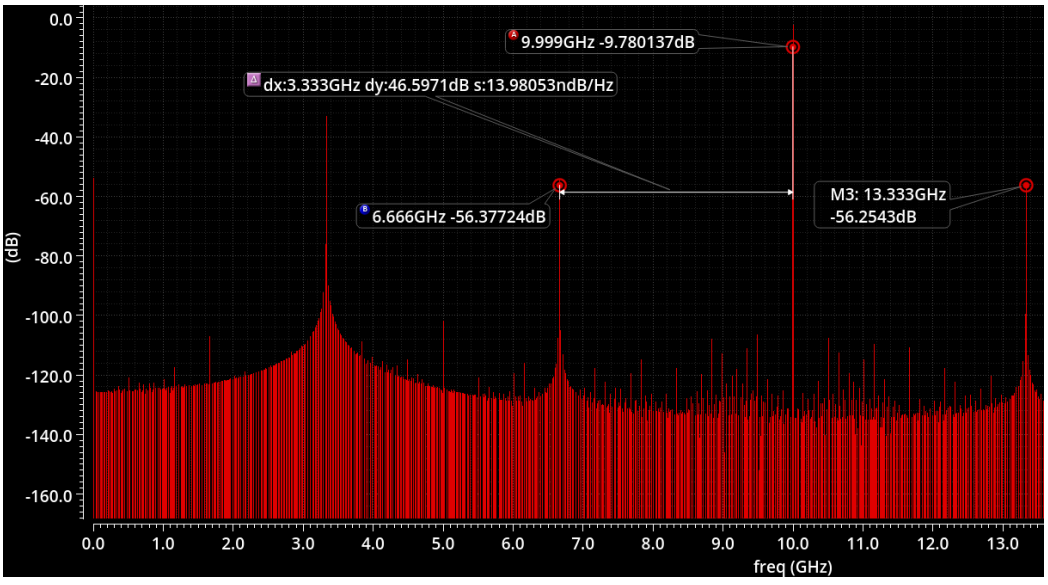

Figure 6.12: Best Case Scenario 2: FFT Plot with minute mismatch differences in the Delay Chains after Fine calibration

The SCR value as seen in the Figure 6.12 is written as:

$$SCR = -46.597\ dBc \tag{6.11}$$

In comparison to the results shown in Figure 6.2 for the Best Case Scenario, the FFT plot shown in the Figure 6.12 has 1.9491 dBc improvement in the SCR value.

6.2 Discussions

- Specify VCDL calibration estimation errors from the script($x_{script_{diff}}$): The VCDL calibration estimation performed in MATLAB has minute errors between the values that the script expects(x_{script} from Equation 4.39) and the values that the script produces with the addition of the Fine calibration settings along with the the Multiplexer offsets($x_{scriptNew_{FineCalib}}$ from Equation 4.53). This is explained with real values below as follows: The x_{script} matrix is written as:

$$\left[x_{script}\right] = \begin{bmatrix} 100.3\ ps \\ 97.3\ ps \\ 102.6\ ps \\ 0.3\ ps \\ 0.3\ ps \\ 0.3\ ps \end{bmatrix} \tag{6.12}$$

The $x_{scriptNew_{FineCalib}}$ matrix is written as:

$$\left[x_{scriptNew_{FineCalib}}\right] = \begin{bmatrix} 100.13\ ps \\ 97.68\ ps \\ 102.38\ ps \\ 0.3\ ps \\ 0.3\ ps \\ 0.3\ ps \end{bmatrix} \tag{6.13}$$

The difference between x_{script} matrix and $x_{scriptNew_{FineCalib}}$ matrix is written as:

$$x_{script_{diff}} = x_{script} - x_{scriptNew_{FineCalib}} \tag{6.14}$$

$$\begin{bmatrix} x_{script_{diff}} \end{bmatrix} = \begin{bmatrix} 0.2078\ ps \\ -0.4066\ ps \\ 0.2272\ ps \\ -0.0305\ ps \\ 0.0131\ ps \\ -0.0026\ ps \end{bmatrix} \tag{6.15}$$

The $x_{script_{diff}}$ matrix represents the errors in the script estimation.

- As shown in equation 4.15, the total cycles obtained from the simulation for a certain calibration time has computation errors which has to be accounted when estimating the $cycleTime_k$ since computation errors would result in (20 fs - 30 fs) difference in the $cycleTime_k$ values.
- Limited step size of the DTC calibration and Fine Calibration systems restricts the amount of mismatch errors that can be solved by the calibration systems.
- Asymmetries in the design blocks lead to a difference in the rise time and fall time of the signals, which in turn would generate computation errors in the cycle counts as mentioned in the first point.
- The calibration mechanism might also be affected by mismatches of its own and its effect over the behaviour of delay changes might be non-linear(second-order mismatch effects).
- There might be other circuit imperfections that need to be accounted by the circuit for a precise calibration procedure? (for example supply ripple effects over the delay, based on the calibration scheme in use at a point in time)

6.2.1 Combined calibration SCR compared with delay SD calibration SCR

As observed from figures 6.11 and 6.9 in the previous section, the spurious performance of the Frequency Synthesizer is lower when the SPE calibration algorithm is run together with the delay SD calibration procedure. This result confirms the formulated mathematical model of the mismatches in Section 2.3, where the magnitude of the spurious tones is affected by all three sources of imperfections.

Thus, the resulting reduction in SCR when calibration systems are combined is a direct result of insufficient SPE compensation from the DSC block. This can be motivated by an insufficient resolution from the DTC components, too big

of a mismatch in the PD to be compensated properly with the maximum DTC state and insufficient optimisations in the DSC design due to time constraints. The minimum SPE calibration value obtained in our worst-case scenario after calibration was of 1.6 ps. However, as this method of calibration has already been proven to obtain even better calibrated SPE values in [9], future improvements upon the SPE calibration circuit could still result in better overall SCR values by combining the two approaches suggested in this thesis.

6.2.2 Problems Associated with 60° Phase Difference Delay Chains

In the initial VCDL setup, 6 delay cells were used in total to produce 50 ps delay for each delay cell i.e., 60° for each delay cell. However, it was difficult to compensate for the added delay along with the desired delay of each delay cell due to which the desired phase difference between each delay cell was not obtained.

As a result, an output frequency of 9.999 GHz could not be produced by the edge-combining circuit since the clock signals from each Delay Stage were not 50 ps apart. i.e. not having 60° phase difference between each other. The only option was to increase the input clock frequency or to increase the number of Delay Stages to bring the output frequency close to 9.999 GHz. Hence, a differential delay chain approach is implemented by having '3 delay cells' in both the delay chains. By implementing delay cells in this way, a 120° phase difference was obtained between each delay cell i.e., 100 ps between each delay cell in an ideal case with no mismatches in the delay chain.

6.2.3 Worst-case total calibration time

Moreover, the values illustrated in the results section shows a possible typical calibration time. However, a scenario can be imagined where the SPE calibration algorithm needs to iterate through all possible DTC states to arrive at a final value in both DSC and main DLL circuit. In this case, the worst case calibration time rises to a value of $3t_{dll} + (32 + 32 + 32)(t_{iter} + t_{reset}) + t_{vcdl} = 75.21\mu$s.

For the scope of this thesis, the calibration time is not an issue. However, for certain time-sensitive applications, this result could be unacceptable, taking into account the calibration procedure might need to be re-run, at certain intervals, throughout the operation of the receiver block.

6.2.4 Power consumption of self-calibration circuits

As stated previously in Chapter 4, the two self-calibration circuits for delay SD and SPE correction do not add a significant amount to the final power consumption of the Frequency Synthesizer. This is because most of the sub-blocks of the calibration circuits can be disabled after the compensating states are found.

The only elements that will remain turned ON, after the calibration routines have been performed, are the circuits generating the fine calibration voltage for the Delay Cells, together with the PD DTCs in the main DLL loop. Provided the circuits are optimised for very low power consumption, the total power budget for

the Frequency Synthesizer should not be heavily influenced by the addition of the analog self-calibration systems.

6.2.5 Theoretical conclusions

Inferring that this approach is valid as long as all of our assumptions are valid, then the problem of calibrating the VCDL setup in an EC DLL circuit can always be uniquely solved given:

1. longer calibration time
2. a bigger number of digital control "bits"
3. The fine calibration step size is small enough to compensate for the effect of any $n-th$ order mismatch behaviour
4. The calibration resolution is small enough to compensate for specified mismatch effects (transistors add a small enough delay to compensate for very small mismatch behaviours)

However, the SCR improvement can be limited by improper calibration from the SPE calibration system, leading to worse SCR performance than without it. This is due to time constraints during the design of the DSC and due to DTC limitations. However, provided the DSC can be further optimised, the final SCR value for the combined calibration approach should be better than the one provided by only delay SD calibration.

6.3 Future improvements

- Since the preference for functionality over performance is given, optimising transistor sizes in all the blocks would be one of the key future improvements.
- All the blocks in the design architecture can be modified considering sustainable performance across temperature and voltage shifts over time.
- The resolution of the fine calibration and the DTC calibration systems can be extended beyond '5' bits such that the increase in the number of fine calibration states would lead to more precise compensation of mismatch errors in the circuit.
- The layout mismatch effects are extrapolated with the help of parasitic capacitors in the test bench setup but practically, a layout for the entire circuit should be implemented.
- Calibration mechanisms are built taking into consideration only the effects of SPE and Delay SD. However, a calibration circuit can be set up to examine the effect of input clock DCD errors.
- A digital logic block for both the self-calibration systems can be executed.
- Optimise the DSC block and refine the resolution of the DTC components in the main DLL loop.

Chapter 7
Conclusion

This thesis has presented two calibration methods for two out of three prominent sources of spurious tones which are commonly found in EC DLL-based Frequency Synthesizer circuits. A statistical model has been derived for the representation and the effect of the sources of spurious tones in an EC DLL circuit. A way of measuring the effect of those sources over the spurious performance of the circuit has been derived in the form of the SCR formula. Similarly, a statistical approach towards estimating an average value for the SCR across a range of mismatch values has been described and employed during the "Results and discussion" section of the thesis. A technical explanation together with a derivation of the mathematical basis behind the calibration algorithms has been detailed. A complete description of the circuit blocks used to construct the self-calibration circuits has been provided a series of results have been presented and discussed.

As a result, it has been shown that the combined effect of the self-calibration circuits provides an SCR improvement of up to $7.7dB$, with no significant addition in power consumption. It has also been shown that limitations for SCR improvement come from both the delay SD calibration in the VCDL component, as well as from improper performance in the SPE calibration system. The delay SD limitations arise from improper compensation of added delays and from limited estimation of circuit imperfections that are unrelated to Delay Cell mismatches. Currently, the SCR performance obtained in this thesis proves that the calibration circuits can fulfil the minimal, commercially viable specification limit for SCR derived in Chapter 3, provided the Diplexer can add $25dB$ of attenuation to any interfering signals. However, even for smaller attenuations, it has been concluded in Section 6.2, based on SCR values collected from the "Best case delay SD scenario" and on the results presented in [9], that a better SCR performance can be obtained provided the limitations previously indicated are rectified.

Thus, provided more work can be done to improve the Frequency Tripler's performance and that chip area and calibration time are not an issue for the functionality of the final Wi-Fi transceiver, the self-calibration systems covered in this thesis can provide a significant enough performance improvement to justify replacing the LO generating block presented in the Wi-Fi receiver of Section 1.2.

Chapter 8

References

[1] Ericsson. "Network modernization – on the quest for net zero." (), [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/articles/eand-network-modernization. Accessed 30-05-2023.

[2] Statista. "Wireless router market revenue worldwide from 2017 to 2027." (2022), [Online]. Available: https://www.statista.com/statistics/947604/worldwide-wireless-router-market-value/. Accessed 30-05-2023.

[3] Statista. "Free wi-fi data traffic volume in the united states from 2018 to 2023 (in million gigabytes per year)." (2022), [Online]. Available: https://www.statista.com/statistics/994889/free-wi-fi-traffic-volumes-in-the-us/. Accessed 30-05-2023.

[4] IEEE. "Demand for wi-fi 6 is growing." (), [Online]. Available: https://innovationatwork.ieee.org/demand-for-wi-fi-6-is-growing/. Accessed 30-05-2023.

[5] Qualcomm. "How wi-fi 6 is so much more than just faster wi-fi." (2019), [Online]. Available: https://developer.qualcomm.com/blog/how-wi-fi-6-so-much-more-just-faster-wi-fi. Accessed 30-05-2023.

[6] E. 360. "Challenges and solutions of wi-fi 6 for design engineers." (), [Online]. Available: https://electronics360.globalspec.com/article/14606/challenges-and-solutions-of-wi-fi-6-for-design-engineers. Accessed 30-05-2023.

[7] S. Adel Sedra, C. Kenneth Smith, C. Tony Charn, and V. Gaudet, *Microelectronic Circuits*, Eighth edition. 198 Madison Avenue, New York, NY, United States of America: Oxford University Pres, 2020.

[8] A. Ojani, B. Mesgarzadeh, and A. Alvandpour, "Monte carlo-free prediction of spurious performance for ecdll-based synthesizers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 1, pp. 273–282, 2015. DOI: 10.1109/TCSI.2014.2347231.

[9] Chithra, A. Narayanan, R. S. A. Kumar, and N. Krishnapura, "Autozeroing static phase offset in dlls using a digitally programmable sensing circuit," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 6, pp. 1788–1792, 2021. DOI: 10.1109/TCSII.2021.3049134.

[10] H. M. Nguyen, J. S. Walling, A. Zhu, and R. B. Staszewski, "A ka-band switched-capacitor rfdac using edge-combining in 22nm fd-soi," in *2021 Symposium on VLSI Circuits*, 2021, pp. 1–2. DOI: 10.23919/VLSICircuits52068.2021.9492451.

[11] A. Khashaba, A. Elkholy, K. M. Megawer, M. G. Ahmed, and P. K. Hanumolu, "A low-noise frequency synthesizer using multiphase generation and combining techniques," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 3, pp. 592–601, 2020. DOI: 10.1109/JSSC.2019.2951384.

[12] J. Yang, Q. Pan, J. Yin, and P.-I. Mak, "A 2.0-to-7.4-ghz 16-phase delay-locked loop with a sub-0.6-ps phase-delay error in 40-nm cmos," *IEEE Transactions on Microwave Theory and Techniques*, pp. 1–9, 2023. DOI: 10.1109/TMTT.2023.3242333.

[13] I. T. Union, "Unwanted emissions in the spurious domain," ITU, standard 329, 2012, Accessed 13.02.2023, ch. 1.1. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/sm/R-REC-SM.329-7-199707-S!!PDF-E.pdf.

[14] A. Ojani, B. Mesgarzadeh, and A. Alvandpour, "Modeling and analysis of harmonic spurs in dll-based frequency synthesizers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 11, pp. 3075–3084, 2014. DOI: 10.1109/TCSI.2014.2321188.

[15] B. Rivet, L. Girin, and C. Jutten, "Log-rayleigh distribution: A simple and efficient statistical representation of log-spectral coefficients," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 796–802, 2007. DOI: 10.1109/TASL.2006.885922.

[16] E. W. Weisstein. "Invertible matrix theorem." (), [Online]. Available: mathworld.wolfram.com. Accessed 26-05-2023.

[17] I. Krzysztof, "Wireless technologies, circuits, systems, and devices," Library of Congress Cataloging-in-Publication Data, International Standard Book, 2008, Accessed 26.05.2023, ch. 3.

[18] W. Stone, *Electromagnetic signal attenuation in construction materials*, en, Oct. 1997. DOI: https://doi.org/10.6028/NIST.IR.6055.

[19] G. Wireless. "Signal to noise ratio (snr)." (), [Online]. Available: https://www.gnswireless.com/info/signal-to-noise-ratio-snr/. Accessed 26-05-2023.

[20] muRata. "Rf diplexer datasheet." (), [Online]. Available: https://www.murata.com/products/productdata/8802291122206/LFD182G45DCHD277.pdf?1674703820000. Accessed 26-05-2023.

[21] IEEE, "Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, pp. 2935–2938, 2020, https://standards.ieee.org/ieee/802.11/7028/.

[22] Chithra and N. Krishnapura, "Static phase offset reduction technique for delay locked loops," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5. DOI: 10.1109/ISCAS.2019.8702613.

[23] H.-H. Chang, J.-Y. Chang, C.-Y. Kuo, and S.-I. Liu, "A 0.7-2-ghz self-calibrated multiphase delay-locked loop," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 5, pp. 1051–1061, 2006. DOI: 10.1109/JSSC.2006.874036.

[24] L. Wu and W. Black, "A low-jitter skew-calibrated multi-phase clock generator for time-interleaved applications," in *2001 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC (Cat. No.01CH37177)*, 2001, pp. 396–397. DOI: 10.1109/ISSCC.2001.912690.

[25] S.-C. Lin and T.-C. Lee, "An 833-mhz 132-phase multiphase clock generator with self-calibration circuits," in *2008 IEEE Asian Solid-State Circuits Conference*, 2008, pp. 437–440. DOI: 10.1109/ASSCC.2008.4708821.

Printed by Tryckeriet i E-huset, Lund 2023

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2023-929
http://www.eit.lth.se