Design of a sub-Hz Resolution Fully Digital RF Frequency Synthesizer

Panagiotis Zarkos
Vladimir Stojanovic, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2020-17
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-17.html

January 21, 2020

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my advisor, Vladimir Stojanović, for his valuable guidance and insight during this work, as well as his patience with me. Also, many thanks to the people in the ISG group for numerous useful conversations: Christos Adamopoulos, Sidney Buchbinder, Taehwan Kim, and Pavan Bhargava. I would especially like to thank Sajjad Moazeni for guiding me through the group's digital chip making flow. On a personal level, I would like to thank my friends: Iason, Iason, Alex and Konstantinos for their constant cross-atlantic encouragement over the past 4.5 years; the Greeks of Berkeley: Fotis Iliopoulos, George Patsakis and Katerina Papadopoulou for being my family here; Anna Giannakou for her love and mental support in the write-up of this thesis and the chip testing; and finally, my parents for standing by my side and believing in me.

Abstract

Optical frequency synthesis (OFS) based on self-referenced optical combs has enabled a variety of applications including absolute optical frequency measurements, optical spectroscopy, gas sensing, light detection and ranging (LiDAR), and optical frequency metrology. Arbitrary OFS can be achieved by locking a Continuous-Wave (CW) tunable laser to a programmable offset across multiple comb teeth. This report describes the design of a Radio Frequency (RF) frequency synthesizer that will generate this programmable offset, enabling the optical synthesizer to span frequencies between the comb lines with 1Hz tuning resolution and $^{10^{-13}}/_{\tau}$ Allan deviation. The RF-synth tuning range requirement is 1GHz as dictated by the distance of adjacent optical comb teeth. To satisfy these specs, a Flying Adder frequency synthesis topology is selected. A simulation framework that evaluates the effect of different synthesizer variants on the spectral content and stability of the synthesized output has been implemented in MATLAB Simulink. Simulations of this model are used as a guide for actual circuit design choices. Finally, measurement results of the synthesizer that has been built in silicon are presented.

Chapter 1

Introduction

Since the first demonstration of optical frequency synthesis using self-referenced combs in the early 2000's [1]-[3] a lot of effort has been placed in the development and integration of optical frequency synthesizers (OFS) over the past decade. This technology is expected to pave the way to numerous applications ranging from optical spectroscopy, gas sensing, and metrology to LiDAR, atomic clocks, and high bandwidth coherent communications [4]-[8]. However, integration of optical synthesizers is quintessential to their widespread adaptation in these applications since their current cost, size and power dissipation has been an effective showstopper. This can be understood when looking at commercial products that report power consumption of 0.5kW with a form factor of 0.14m^3 [9]. The aim of the Direct On-Chip Digital Optical Synthesizer (DODOS) project is to integrate all of the separate key components (non-linear optics, mode-lock lasers, tunable lasers, optical modulators and CMOS RF frequency synthesizers) into a functional synthesizer system with power dissipation of 1W and volume of 1cm^3. The efforts for this project have so far, have led to the integration of multiple photonic devices [10], [11].

The approach taken to implement this integrated OFS is to lock a Continuous-Wave (CW) tunable laser to an arbitrary offset from a self-referenced, octave spanning comb. Generation of this comb is achieved by pumping a Kerr medium, in this case Si, with an integrated mode-lock laser

(MLL), and its specifics are beyond the scope of this report. The focus of this work has been on selecting and modifying an RF frequency synthesizer topology that will create this arbitrary offset in a programmable fashion. Once this offset is available, the RF-synth output is mixed with the CW laser through an optical Single-Sideband (SSB) or Serrodyne modulator and the up-converted modulator output is locked to the comb line of interest. That way, all frequencies in-between the comb lines are generated, while the locking loop can be accounted upon to suppress potential harmonics and spurs, originating in the modulator and/or the RF synthesizer, from showing up at the CW laser output. The rest of this chapter will outline the modeling and design approach followed for the RF-synth.

1.1 Scope of Work

Given the context in which this RF-synth will be operating, we derived the resolution, tuning range and stability values that our design had to meet. Specifically, the resolution and tuning range requirements led to the selection of a Flying-Adder (FA) frequency synthesizer topology, a kind of RF-synth very similar in principle to a DDS, but with several additional intricacies that complicate its behavior. From that point on we wanted to first create a simulation framework in which behavioral models would allow us to evaluate standalone synthesizer performance as well as synthesizer impact on the output of the overall locking loop, before proceeding to the actual implementation of the synthesizer in silicon.

This framework was built in MATLAB Simulink, and it consisted of different types of digital delta-sigma ($\Delta\Sigma$) modulator models, which are a basic building block of the FA-synth, complete FA-synth models, as well as CW-locking loop models. Time-domain simulations of these models helped verify theoretical expectations for synthesizer spur locations and design $\Delta\Sigma$ modulators that would decrease spur level. Furthermore, using these simulation results we were able to quantify the impact of spurious tones on the CW laser output in terms of the stability metric of interest, which in our case was Allan deviation. Once we were satisfied with the simulation results, we went on to build the synthesizer chip. The core of the synthesizer was fully digital and occupied an active area of 100μm x 140μm. Given that different $\Delta\Sigma$ modulator variants proved to be advantageous for certain operating frequency ranges, and taking into account the small size of the FA-synth block we implemented three different FA variants. The synthesized outputs of each variant were brought out to a pad and evaluated separately. This concluded the modeling, design and measurement of the FA-synth block to be used in the CW-laser locking loop of an integrated OFS.

1.2 Organization

The rest of this work is structured as follows: first we give a more thorough description of the CW locking loop system, and derive the exact specs for the RF-synth in chapter 2. There, we also establish basic relationships between the given stability metric, Allan deviation, and stability metrics that are more familiar to circuit designers, namely phase noise and jitter. Then, in chapter 3 we go into the details of the chosen synthesizer topology, explaining the spur locations in the RF-synth output spectrum and proving the need for modifications in the simple architecture. Chapter 4 discusses some key properties of digital $\Delta\Sigma$ modulators to provide some insight into the design of the modulators and then presents simulation results from the synthesizers employing higher order modulators. Throughout this report we attempt to refer back to the system level specifications for resolution, tuning range and stability and show how our modifications improve initial results. Finally, chapter 5 provides some design guidelines, implementation details, and post place and route simulations, while chapter 6 presents preliminary measurement results from the taped-out synthesizer chip. Chapter 7 concludes this report.

Chapter 2

System Overview and Design Specifications

An concise overview of the OFS system is shown in Fig. 2.1. Octave spanning comb generation through a pulse generator (MLL) pumping a Kerr material is depicted in the "Frequency Standard Generation" block of Fig. 2.1, while self-referencing of the octave spanning comb is achieved by the Second Harmonic Generation (SHG) block. The SHG loop ensures locking of the so called carrier-envelope offset (f_{ceo}) which guarantees that the absolute frequency of generated comb teeth is known and equal to $f_{comb} = f_{ceo} + nf_{rep}$, where f_{rep} is the MLL repetition rate. The OFS also includes the "Synthesizer" block of Fig. 2.1, which consists of a CW tunable laser source that is locked to the comb, an optical SSB modulator (or a serrodyne modulator) and the RF synthesizer, which is not shown in this figure. Here the arbitrary offset we talked about is depicted as f_{SSB}. In the rest of this work, f_{SSB} will be denoted mostly as f_{out}. In the following, the operation of the CW locking loop will be detailed along with useful derivations of the RF-synth specs. A conversion between the required stability metric (Allan deviation) and the most frequently used ones in RF frequency synthesis (phase noise and jitter) will also be provided.

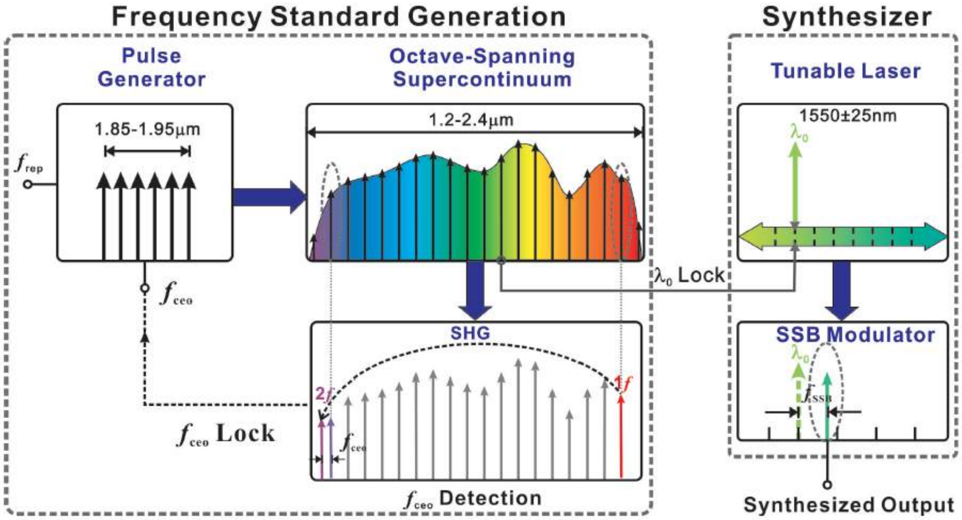


Figure 2.1: OFS System Level Block Diagram

2.1 CW Laser Locking Loop Operation

An overview of the CW laser locking loop is shown in Fig. 2.2. Here, the CW laser is taken as the final OFS output. Compared to a scheme that would take the SSB output as the final OFS output (Fig. 2.1), this scheme (Fig. 2.2) has the advantage of utilizing the proportional integral (PI) control block, which in practice is a thermal tuner, to perform filtering of the RF-synth and SSB harmonics that persist in the output spectrum. Furthermore, in the first scheme in the case that we attempt to synthesize frequencies close to $f_o + f_{rep}/2$ there is an ambiguity regarding the combline to which we are locked (i.e. n or n+1 combline of the octave-spanning supercontinuum), which in turn is translated in ambiguity of the synthesized frequency itself. By locking the output of the optical SSB to a constant offset of $f_{rep}/4$ from the comb tooth of interest we ensure that the beat notes of the CW with the comb teeth that appear at the output of the photodetector (PD) are equal to $(2n + 1)\frac{f_{rep}}{4}$. A band-pass filter (BPF) following the PD is used to reject the beat notes for which $n \neq 0$.

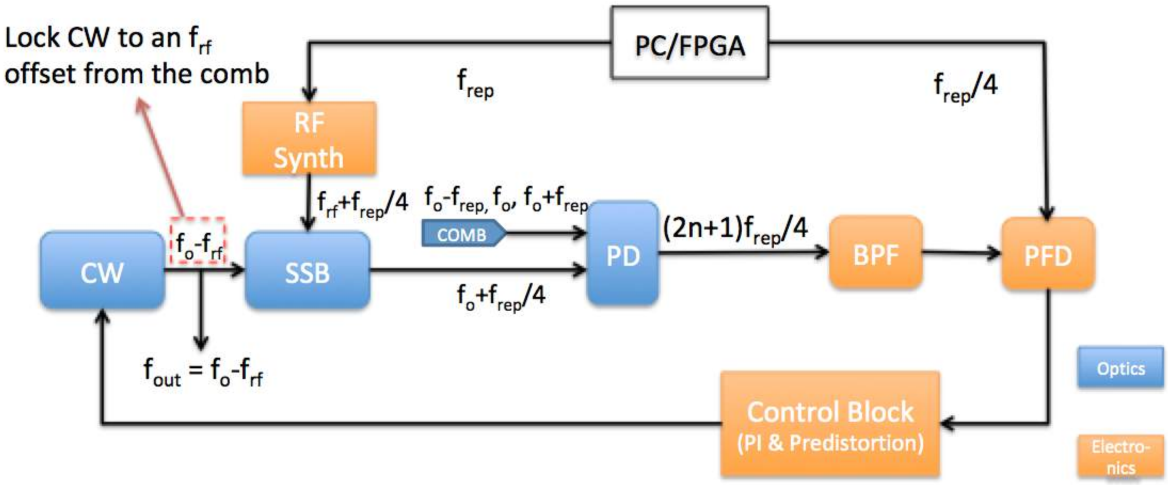


Figure 2.2: CW Laser Locking Loop

2.2 Resolution and Tuning Range

Resolution and tuning range specifications of the RF-synth are derived from the system requirement that the CW laser spans the entire C-band with relative accuracy of one part in 10^{15}. This is a band spanning 1530-1565nm or equivalently 192-196THz. Thus the required frequency resolution is on the order of:

$$f_{res} = \frac{f_{out}}{10^{15}} \approx 0.2\text{Hz} \tag{2.1}$$

Assuming a CW source with sufficient tuning range to span the C-band is available, the optical comb lines can be used as a coarse locking mechanism. In order to generate all in-between frequencies with the desired precision, the **RF-synth tuning range has to be equal to the comb line spacing**. As mentioned above, in our system that is equal to the MLL repetition rate, which is expected to be **1GHz**.

2.3 Stability

Typical stability and noise metrics used in circuit design include phase noise and RMS cycle-to-cycle jitter. On the other hand, people in metrology specify the long-term stability of a synthesizer in terms of Allan deviation or Allan variance. The purpose of this section is to derive the Allan deviation requirement for the RF-synth and bridge the gap between that and circuit design stability metrics.

Allan deviation was introduced as an oscillator stability metric by D. Allan [12] in order to address non-convergence of classical statistical tools such as the standard deviation, when the oscillator under test exhibited white or flicker noise in the **frequency** rather than the **phase** domain. At this point it seems necessary to introduce some fundamental functions regarding Allan deviation. Assuming we have an oscillator described by: $V_{out} = Asin(\Phi(t))$, we can define the following:

Phase Function:

$$\Phi(t) = 2\pi f_{nom} t + \phi_n(t) \tag{2.2}$$

Frequency Function:

$$f(t) = \frac{1}{2\pi} \frac{d\Phi(t)}{dt} \tag{2.3}$$

Time-error Function:

$$x(t) = \frac{\phi_n(t)}{2\pi f_{nom}} \tag{2.4}$$

Fractional Frequency Function:

$$y(t) = \frac{f(t) - f_{nom}}{f_{nom}} = \frac{1}{2\pi f_{nom}} \frac{d\phi_n(t)}{dt} = \frac{dx(t)}{dt} \tag{2.5}$$

With these functions at hand we have the tools to calculate Allan deviation and express it in many algebraic forms. Before we do that though, we need to define the N-sample variance of the fractional frequency function, which is a measure of the oscillator frequency fluctuations from the nominal value:

$$\begin{aligned}\sigma_y^2(N,T,\tau) &= \frac{1}{N-1}\left\{\sum_{i=0}^{N-1} y_i^2 - \frac{1}{N}\left[\sum_{i=0}^{N-1} y_i\right]^2\right\} \\ &= \frac{1}{N-1}\left\{\sum_{i=0}^{N-1}\left[\frac{x(iT+\tau)-x(iT)}{\tau}\right]^2 - \frac{1}{N}\left[\sum_{i=0}^{N-1}\frac{x(iT+\tau)-x(iT)}{\tau}\right]^2\right\}\end{aligned} \tag{2.6}$$

In eq. (2.6) it is assumed that we have N samples of the oscillator under test taken at time intervals T seconds apart. Each sample consists of two measurements acquired at time instances iT and $iT+\tau$, where τ is said to be the observation period of the measurement. If counters with no dead-time between measurements are used we can allow $T=\tau$. Using the notation of eq. (2.6) **Allan variance** is defined as the 2-sample variance of the fractional frequency function derived using a counter with no dead-time:

$$\sigma_y^2(\tau) = \sigma_y^2(2,\tau,\tau) \tag{2.7}$$

And **Allan deviation** as its square root. Essentially, this reduces to one-half the squared average of the first finite-difference of the fractional frequency function and one-half the squared time average of the second finite-difference of the time-error function, so we can write:

$$\sigma_y^2(\tau) = \frac{1}{2}\left\langle (y_{i+1}-y_i)^2\right\rangle = \frac{1}{2\tau^2}\left\langle \left(x_{i+2}-2x_{i+1}+x_i\right)^2\right\rangle \tag{2.8}$$

It is apparent from eq. 2.8 that Allan deviation is a function of the observation period τ. The DODOS project has a goal of producing optical outputs with Allan deviation equal to $\frac{10^{-13}}{\tau}$ at $\tau = 1$sec. This number however, refers to the optical output and needs to be down-converted to RF. Since Allan deviation is defined as the two-sample variance of the fractional frequency of an oscillator we can expect this spec to be relaxed proportionally with the carrier frequency value. Hence, going down from 100THz to 1GHz the Allan deviation spec for the RF-synth will be $\frac{10^{-8}}{\tau}$. That would be in the case that the RF-synth was accountable for all of the noise showing up at the optical output. A 20% margin will be imposed to our design bringing the Allan deviation target for the RF-synth down to $\frac{0.2\cdot 10^{-8}}{\tau}$

2.3.1 Allan Deviation to Phase Noise

The basic equation that allows to convert from the time to the frequency domain was introduced in [13] and simplified in [14]:

$$\sigma_y^2(\tau) = \int_0^\infty 2S_y(f)\frac{sin^4(\pi f\tau)}{(\pi f\tau)^2} \tag{2.9}$$

Where $S_y(f)$ is the power spectral density (PSD) of the fractional frequency function. Combining eqs. (2.2)-(2.5) we can derive the following relationships regarding the PSDs of the functions that we defined above:

$$\begin{aligned} S_\phi(f) &= \left(\frac{f_{nom}}{f}\right)^2 S_y(f) \\ S_x(f) &= \left(\frac{1}{2\pi f}\right)^2 S_y(f) \\ S_\phi(f) &= \left(2\pi f_{nom}\right)^2 S_x(f) \end{aligned} \tag{2.10}$$

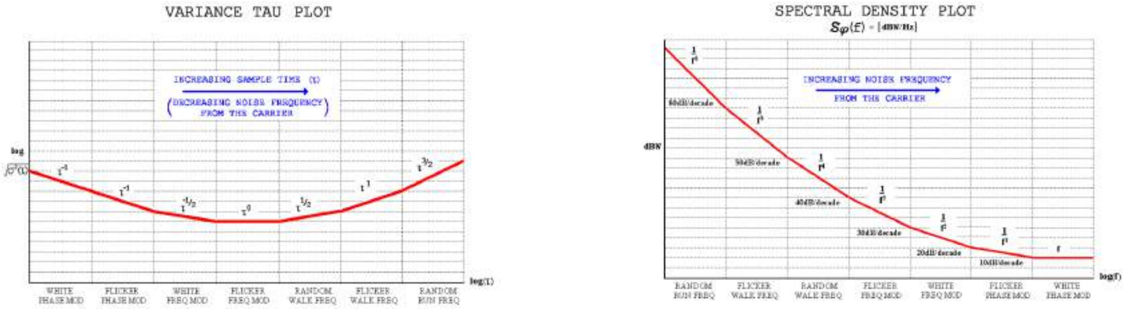
By definition, phase noise is equal to the single-sided PSD of phase fluctuations so:

$$L(f) = \frac{1}{2}S_\phi(f) \tag{2.11}$$

The above equations provide a link between Allan variance and phase noise, however in most cases it is not possible to analytically calculate the integral of eq. (2.9). The practical solution used when a quick, approximate conversion is needed is to assume that one type of noise source dominates at a certain offset from the carrier. Knowing the power-law model describing different types of noise according to which $S_\phi(f) \propto f^\alpha$ and using (2.9), (2.10) we can derive expressions linking Allan deviation directly to phase noise. These expressions have been tabulated in [14] and are also repeated in table 2.1 here for convenience. Inspection of table 2.1 reveals that white and flicker noise in the phase domain have the same dependency on τ and thus cannot be distinguished from one another using Allan deviation. To resolve this ambiguity a modification known as modified Allan variance has been introduced. The interested reader is pointed to [15]. Power-law relationships for all types of oscillator noise are illustrated in the log-log plots of Fig. 2.3 [16]. We can see that for increasing observation time the noise sources dominant are the ones that we typically find close to the carrier such as white and flicker noise in the phase domain. It is also worth noting that

Table 2.1: Phase noise to Allan deviation per Noise Type [14]

Noise Type	$\mathbf{S}_\phi(\mathbf{f})$
White Phase	$\frac{(2\pi)^2}{3f_h}(\tau\sigma_y)^2 f_{nom}^2$
Flicker Phase	$\frac{(2\pi)^2}{A}(\tau\sigma_y)^2 \frac{f_{nom}^2}{f}$
White Freq.	$2\tau\sigma_y^2\left(\frac{f_{nom}}{f}\right)^2$
Flicker Freq.	$\frac{6}{(2\pi)^2}\tau^{-1}\sigma_y^2\frac{f_{nom}^2}{f^3}$
f_h: measurement system bandwidth $A = 1.038 + 3\ln(2\pi f_h)$	



(a)

(b)

Figure 2.3: Typical log-log plots for (a) Allan deviation, and (b) $S_\phi(f)$ [16]

for noise sources such as flicker walk in the frequency domain, for which $S_\phi \propto f^\alpha$ with $\alpha < -4$ Allan deviation diverges.

2.3.2 Allan Deviation to Jitter

The first jitter quantity we need to define before making the connection between these time-domain stability metrics is timing jitter. If we denote time instances of zero-crossings as $t_k, k \in \mathbb{N}$ in a clock under test, with nominal period T_{nom}, then timing jitter is:

$$J_k = t_k - kT_{nom} \tag{2.12}$$

Which is the same as the time-error function defined in eq. (2.4). However, there is a significant difference in the sense that the time-error function is continuous, while timing jitter is a discrete sequence. Specifically, it is:

$$J_k = -x(kT_{nom}) \tag{2.13}$$

Where $x(t)$ is the time-error function. Moving on, we can define period and cycle-to-cycle jitter as the first and second finite-difference of timing-jitter [17]:

$$P_k = J_{k+1} - J_k \tag{2.14}$$

$$j_{c2c,k} = P_{k+1} - P_k = J_{k+2} - 2J_{k+1} + J_k \tag{2.15}$$

From eq. 2.15 and remembering that Allan variance is one-half the squared time average of the second finite-difference of the time-error function (eq. (2.8)) we can deduce the following, which concludes this Allan deviation to jitter conversion:

$$\sigma_y(T_{nom}) = \frac{1}{\sqrt{2}T_{nom}} j_{c2c,RMS} \tag{2.16}$$

This section has provided a quick review of the definition of Allan deviation. It also has attempted to bridge the gap between Allan deviation, phase noise, and jitter by rederiving conversion formulas from one to another. Ability to perform such conversions will prove useful in the course of this work, since it will allow us to evaluate our RF-synth design and compare it to existing architectures using familiar metrics such as phase noise, jitter and SFDR, while at the same time ensuring we are meeting project spec. Another useful reference outlining conversion from phase noise to jitter can be found in [18].

Perhaps as a final note we should mention that it is common practice to use jitter and phase noise in short timescales on the order of ms and below. Allan deviation on the other hand, is used for larger timescales, on the order of seconds, which is the range of interest in metrology applications. It is a metric used to evaluate long-term stability rather than fast fluctuations of the oscillator phase.

Chapter 3

The Flying Adder Frequency Synthesizer

In this chapter, the circuit architecture along with the fundamental equations describing the principle of operation of the synthesizer are initially presented. This mathematical analysis shows the advantages (large tuning range and fine frequency resolution) that make this architecture suitable for our application, but also brings forth the main drawback of this architecture, namely the dense spurious content of the output spectrum. The analysis is supported with simulation results, which point out the pathological cases of this type of synthesizer. To address this issue, higher order delta-sigma ($\Delta\Sigma$) modulation in the phase accumulator block is proposed.

3.1 Synthesizer Architecture

A system level block diagram of the Flying Adder frequency synthesizer first proposed in [19] is shown in Fig. 3.1. It consists of a phase generation a phase accumulation and a phase selection unit.

The phase generation can be implemented as a DLL, a PLL, or even as a simple divider network, which is what is used in this work and is shown in Fig. 3.2a. The purpose of this block is to

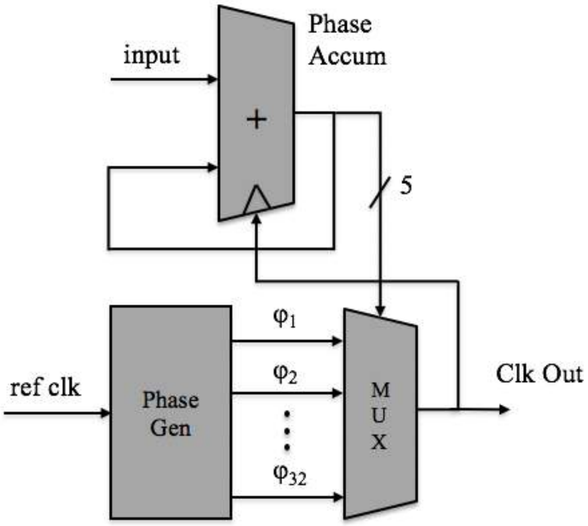
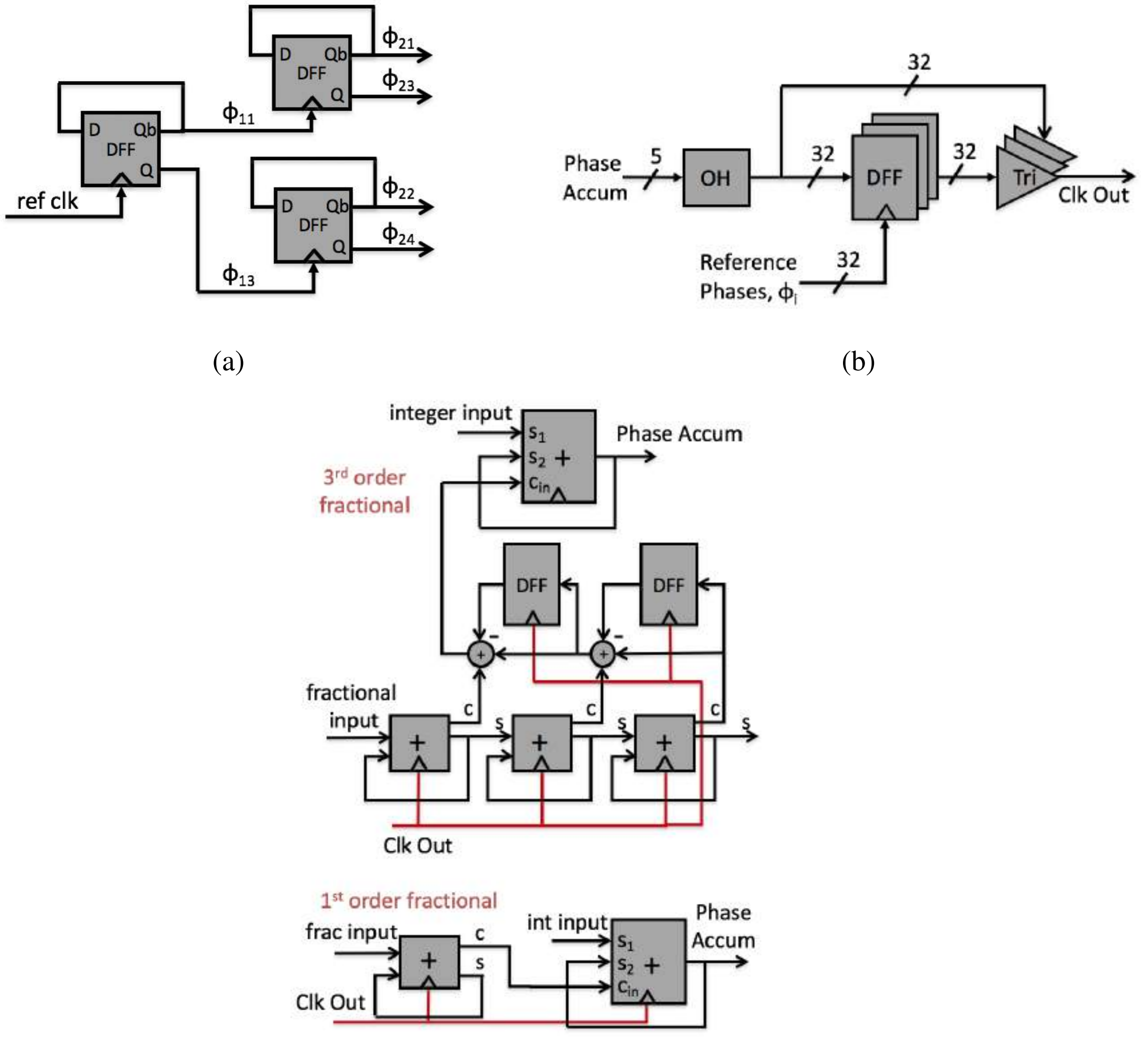


Figure 3.1: Flying Adder Block Diagram

generate N equally spaced phases of a reference clock. These reference phases are appropriately combined to produce the desired output frequency. Selection of the next phase is controlled by the phase accumulator. Phase accumulation is broken down to an MSB and an LSB sub-accumulator each of which is responsible for integer and fractional synthesis as will be shown later on. The carry-out of the LSB sub-accumulator is fed into the carry-in of the MSB thus implementing $\Delta\Sigma$ modulation. The above are depicted in Fig. 3.2c , which shows 1st order and 3rd order **M**ulti St**A**ge Noise **Sh**aping (MASH) $\Delta\Sigma$ modulation, where the carry-in essentially modulates is the phase increment of the synthesized clock. The modulator order used affects the "depth" of phase increment modulation as well as the periodicity of the accumulator output sequence, a quantity very important for the synthesizer spurs. It should be noted that the first implementations of this architecture [19], [20] did not use any sub-accumulators or $\Delta\Sigma$ modulation, but simply truncated the $m = \log_2 N$ MSB's of a single n-bit accumulator. (It is suitable to select N to be a power of two since this simplifies decoder and MUX design).

Finally, the phase selection operation is described in Fig. 3.2b. The phase selection unit consists of a OH-decoder controlling a number of DFFs followed by tri-state buffers. The reference phases clock a DFF each, while the DFF outputs are fed into the tri-state buffers. The outputs



(a)

(b)

(c)

Figure 3.2: Details of Synthesizer Building Blocks. (a) Divider network based phase generation, (b) phase multiplexer, and (c) phase accumulator (1^{st} and 3^{rd} order modulation)

of the OH-decoder control the tri-state buffers enable port. This implementation of the phase selection unit prevents any glitching to propagate to the output while the phase accumulator is still calculating the next reference phase to be selected. Many implementations of this architecture use an additional divide-by-2 flop at the end to transform the irregular pulse train to square waveform with a duty cycle closer to 50% .

The main advantage of this synthesizer topology is that it is self clocked and fully digital, while at the same time allowing for fast frequency hopping and wide tuning range. It is similar to a DDS with the subtle difference that it does not require a LUT or an external clock. However, it suffers from dense spurs in its output spectrum, due to inherent timing irregularities of the output waveform. The design of the phase accumulator is key to spur minimization at the synthesizer output, since it acts as a $\Delta\Sigma$ modulator as will be explained in the following sections and is shown in Figs. 3.1, 3.2. Several accumulator variants from 1st order to 3rd order, have been designed and evaluated both in behavioral models and post place and route layout simulation. It is shown in this work and in [21] that higher order modulation randomizes the accumulation operation for fractional inputs and helps distribute the quantization noise over a higher portion of the spectrum.

3.2 Principle of Operation - Governing Equations

As it has already been mentioned frequency synthesis in the FA architecture is performed by proper selection of the next phase, through the accumulator block. The digital codeword fed into the accumulator dictates it's output sequence and thus controls next phase selection, setting the output frequency to:

$$f_{out} = \frac{2^n}{w} f_{ref} \tag{3.1}$$

Where we define w as the digital codeword with n number of bits, and f_{ref} as the frequency of all reference phases. We repeat at this point that we have defined $m = \log_2 N \in \mathbb{Z}$ the number of bits of the integer sub-accumulator. It is also useful to define the time increment between two adjacent reference phases as $\Delta = \frac{1}{N f_{ref}}$ (also being the period of the highest theoretically synthesizable frequency).

It might have been already obvious that this synthesizer can only produce frequencies higher than the frequency of the reference phases. A metric of interest for our application of CW laser locking is the range of synthesizable frequencies. In this architecture, the minimum and maximum frequencies that can be synthesized are:

$$\begin{aligned} f_{out,max} &= 2^m f_{ref} \\ f_{out,min} &= \frac{2^n}{2^n - 1} f_{ref} \end{aligned} \tag{3.2}$$

Putting this into perspective, using a reference frequency of 312.5MHz and 32 reference phases (i.e. $N = 32, m = 5$), we get that $f_{min} \approx 312.5$MHz and $f_{max} = 10$GHz, which easily meets our 1GHz tuning range requirement.

Another important advantage of the Flying Adder frequency synthesizer is the high frequency resolution that it provides. This resolution is at its minimum (best) for the lowest synthesizable frequencies and at its maximum (worst) for the highest synthesizable frequencies. These values are [21]:

$$\begin{aligned} \delta f_{min} &= \frac{2^n}{2^n - 2} f_{ref} - \frac{2^n}{2^n - 1} f_{ref} \approx \frac{1}{2^n} f_{ref} \\ \delta f_{max} &= \frac{2^n}{2^{n-m}} f_{ref} - \frac{2^n}{2^{n-m} + 1} f_{ref} \approx \frac{1}{2^{(n-2m)}} f_{ref} \end{aligned} \tag{3.3}$$

To gain some more insight about the theoretical limitations of this circuit, we extend the previous numerical example: with the same f_{ref} and N assume we use 32-bit long codewords. Eq. (3.3) yields $\delta f_{min} = 0.07$Hz, $\delta f_{max} = 74.5$Hz. Worst case resolution is an important parameter based on which we will select the number of bits of the input codeword and consequently the size of the integer and fractional sub-accumulators. This design procedure is outlined in more detail in chapter 5 and uses eq. (3.3) to hit the sub-Hz resolution spec at the highest frequency we are interested in generating based on the tuning range requirement.

Having derived the basic equations for output period, tuning range and resolution we will now focus on the output spectrum of the Flying Adder synth. It has been shown in [22] that input codewords with $w < 2^{n-m}$ are redundant, that is, all possible frequencies can be synthesized using codewords $w \geq 2^{n-m}$. Furthermore, inputs with $w < 2^{n-m}$ are impractical since they take too long to overflow the fractional sub-accumulator output and result in rather irregular patterns for the synthesized clock, consequently generating spurs too close to the carrier that are difficult to filter out.

Taking into account only codewords with $w \geq 2^{n-m}$ we can discern two basic cases: $w \bmod 2^{n-m} = 0$, which implies that the n-m LSB's are zero and corresponds to integer synthesis, and $w \bmod 2^{n-m} \neq 0$, in which case fractional synthesis occurs. In the first case the accumulator continuously increments the output phase by the same amount, the output period is an integer multiple of Δ, and the deterministic jitter is ideally zero. That is not true for the second case, in which we get modulation of the number of phase shifts whenever the fractional sub-accumulator overflows and feeds a non-zero carry into the integer sub-accumulator. Given that $w \bmod 2^{n-m} \neq 0$ we can break down the input codeword into an integer and a fractional part:

$$\begin{aligned} \frac{w}{2^{n-m}} &= I + \frac{X}{Y} \\ I &= \lfloor \frac{w}{2^{n-m}} \rfloor \\ \frac{X}{Y} &= \frac{w \bmod 2^{n-m}}{2^{n-m}} \end{aligned} \tag{3.4}$$

With X, Y prime to each other.

In integer synthesis the output period is always $T = I\Delta$. However, in fractional synthesis since the circuit can only produce periods that are multiples of Δ, the output period alternates between $T_1 = I\Delta$ and $T_2 = (I+1)\Delta$ with the time-average period being $T = 1/f_{out}$, f_{out} and given by (3.1). The period of time over which this averaging is taking place, determines the frequency offset at which harmonics appear at the output spectrum. We will be referring to this period of time as the fundamental period of the phase selection sequence. It intuitively makes sense that in the case of 1st order $\Delta\Sigma$ it takes Y cycles for the accumulator to return to its initial state, out of which X are $(I+1)\Delta$ long and $Y - X$ are $I\Delta$ long. Analytically, the fundamental period can be written as [20], [22]:

$$T_{fund} = \big[(I+1)X + I(Y-X)\big]\Delta = (IY + X)\Delta \tag{3.5}$$

Thus we can expect a spur appearing at frequencies:

$$f_{spur} = f_{out} \pm \frac{1}{T_{fund}} = f_{out} \pm f_{fund} \tag{3.6}$$

As an example that clarifies this accumulation operation we can set $I = 2, \frac{X}{Y} = \frac{3}{4}$, (i.e. $w = (2 + \frac{3}{4})2^{n-m}$), with the initial condition being zero. For this input codeword we will get the following accumulation sequence: $\{2\Delta, 3\Delta, 3\Delta, 3\Delta\}$, while the actual accumulation result is: $\{2\frac{3}{4}, 5\frac{1}{2}, 8\frac{1}{4}, 11\}$. We can see that every 4 cycles we get 3 cycles of length 3Δ and 1 cycle of length 2Δ. Simulations of the synthesizer model presented in the next section, demonstrate the above argument more clearly.

3.3 Synthesizer Simulink Model - 1^{st} Order $\Delta\Sigma$ Spur Locations

In order to evaluate the effect of the different synthesizer parameters such as the input codeword, number of fractional and integer accumulator bits, and $\Delta\Sigma$ modulator order on the metrics of interest, namely resolution, tuning range, and Allan deviation (and implicitly spectral clarity) we have built a model in MATLAB Simulink. This model is depicted in Fig. 3.3.

In this model we do not have an actual dll, rather we explicitly multiplex 16 square-wave pulses, running at 312.5MHz, in one Simulink bus. These pulses play the role of the reference phases, with each phase being delayed 200ps from the previous one. Using such a simplified model for the synthesizer allows us to focus on its inherent irregularities and is much easier to

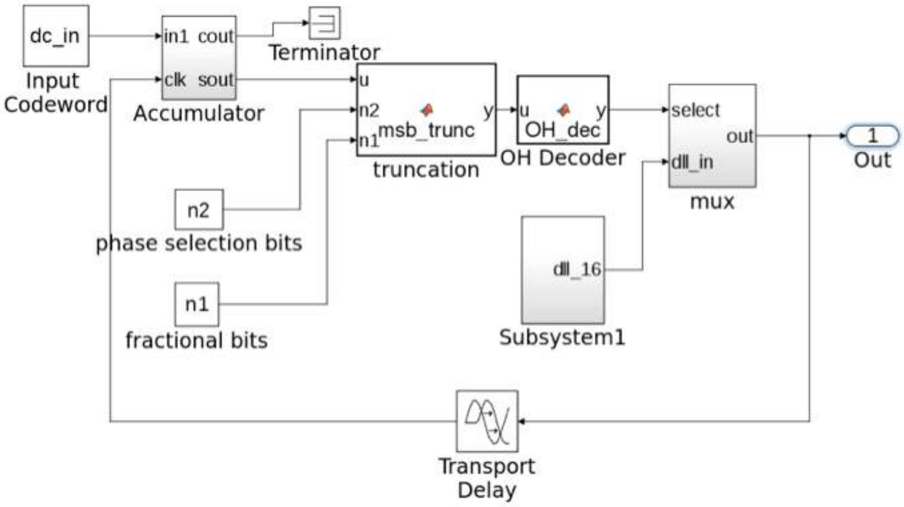


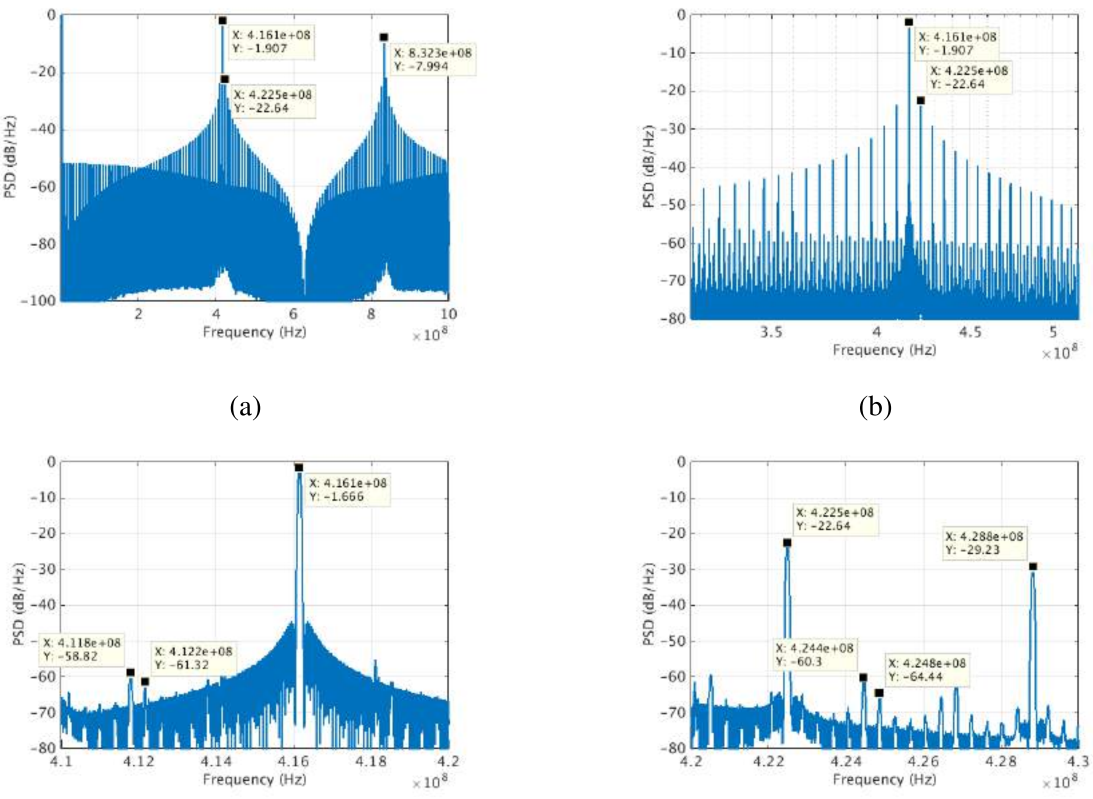
Figure 3.3: Synthesizer Model - 1^{st} order $\Delta\Sigma$ phase accumulator

debug at the same time. Real-life non-idealities such as delay mismatch between the reference phases can be easily added in this model if necessary. Simulation of models incorporating delay mismatch has given very similar results, which indicates that the effect of delay mismatch in the reference phases is not as important (especially for fractional codewords). This point will be made clearer in the following chapters.

The phase multiplexer can be ideally modeled as a dot product between different phases and the output of the OH-decoder following the phase accumulator. Finally, 1st order $\Delta\Sigma$ for the phase accumulator can be implemented by simply truncating the m MSBs of a single accumulator block which is what is shown in Fig. 3.3.

Having explained how the model is put together we now move on to use it in order to validate and expand the conclusions of the previous section. It is obvious from eq. (3.5), (3.6) that larger Y (longer fundamental periods) will result in spurious tones closer to the carrier frequency which are very difficult to be filtered out. In that sense, the most problematic set of input codes is the one for which $Y = 2^{n-m}$. This implies that the $n - m$ LSB's of the input codeword (input of the fractional sub-accumulator) form a number non-divisible by 2^{n-m}. Such a case is shown here for an accumulator with 30 bits, 4 of which are responsible for integer frequency synthesis (since we have a 16 phase "DLL") and the 26 implement fractional frequency synthesis. The input is set equal to $w = 2^{n-1} + 2^{n-2} + \lceil \frac{2^n}{1051} \rceil$, which if we recall the definition of X/Y in (3.4), gives: $\frac{X}{Y} = \frac{w \bmod 2^{n-m}}{2^{n-m}} = \frac{\lceil \frac{2^n}{1051} \rceil}{2^{n-m}} = \lceil \frac{2^m}{1051} \rceil \approx \frac{2^m}{1051}$. This ensures that we get a fundamental period that is long enough to create strong spurs close to the carrier, at an offset where they can be captured by reasonable FFT resolution (which can be achieved through increasing simulation time or through zero padding). For comparison a non-pathological case where $w = 2^{n-1} + 2^{n-2}$ is also shown here. Finally, for completeness we show the case where $w = (2 + \frac{3}{4})2^{n-m}$ which we used as an example previously. The simulated spectra and their zoomed in versions are presented below in Figs. 3.4 - 3.5

With respect to Fig. 3.4 we can first validate eq. (3.1) which gives the output (carrier) frequency and in this case is: $f_{out} = \frac{2^n}{w} f_{ref} = 416.138$MHz for this choice of w. The strong spur at a



(a) (b)

(c) (d)

Figure 3.4: (a) Simulated spectrum for $w = 2^{n-1} + 2^{n-2} + \lceil\frac{2^n}{1051}\rceil$, (b)-(d) zoomed in versions revealing the effect of periodicity in spur locations

harmonic frequency of the output (here 832.3MHz), shown in Fig. 3.4a, is to be expected since our output is a square wave. However, we can also see in Fig. 3.4b that for the case of $w = 2^{n-1} + 2^{n-2} + \lceil\frac{2^n}{1051}\rceil$ the output spectrum is very dense with spurs of relatively high magnitude (about -20dB worst case) very close to the carrier. By extending eq. (3.5), which gives the periodicity of the phase accumulation sequence, we can predict the spur locations:

$$\begin{aligned} T_{fund} = (IY+X)\Delta \overset{(3.4)}{\Longrightarrow} \frac{T_{fund}}{Y} = \frac{w}{2^{n-m}}\Delta = \frac{w}{2^n}T_{ref} \\ \Longrightarrow \quad f_{fund}Y = \frac{2^n}{w}f_{ref} \overset{(3.1)}{\Longrightarrow} Y = \frac{f_{out}}{f_{fund}} \end{aligned} \tag{3.7}$$

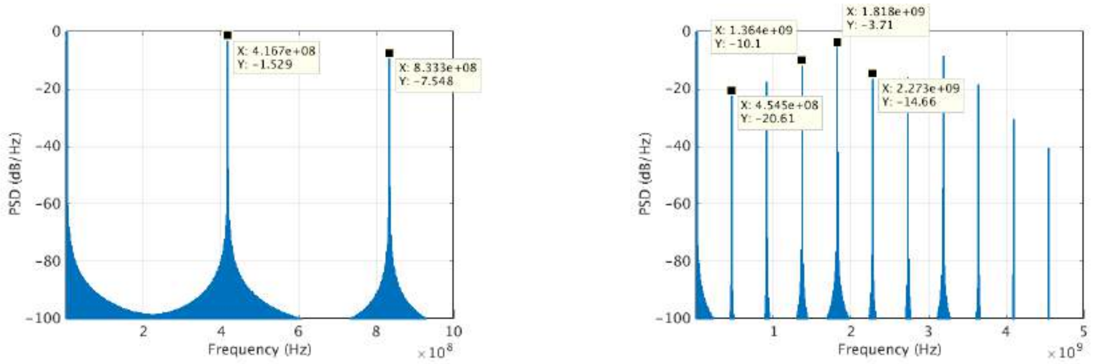
Eq. (3.7) can be validated from the simulated spectrum in Fig. 3.4c in which case we get:

$$f_{fund} = (412.2 - 411.8)\text{MHz} = 0.4\text{MHz},\, f_{out} = 416.1MHz$$
$$\implies Y = \frac{f_{out}}{f_{fund}} = 1040.25$$

A result close to $Y = 1051$ which we are expecting for the given input. Another observation at this point is that every 16 spurs we obtain a strong one, which is directly related to the fact that our fractional input is approximately $\frac{2^m}{1051}$, with $m = 4$. Under close examination we can also see that there are 15 intermediate spurs between the ones at 422.5MHz and 428.8MHz.

As far as the integer input is concerned we can see that we get almost ideal waveforms with only harmonic spurs appearing for the special case where $w = 2^{n-1} + 2^{n-2}$ and no spurs close to the carrier (Fig. 6.3a), while for the case where $Y = 4$ (small) the fundamental spur is indeed as predicted by eq. (3.7) as $f_{fund} = f_{out}/4 = 454.5$MHz. Both figures agree with eq. (3.1) regarding the value of the output frequency, which is 416.7MHz and 1.82GHz in Figs. 6.3a and 6.3b respectively.

The above simulations dictate that in order to get acceptable output spectrum over the entire tuning range (including fractional inputs) we need to come up with an architecture that will be able to reduce the spur level. To this direction we will introduce higher order $\Delta\Sigma$ modulation in the fractional sub-accumulator. It will be shown that even though we end up with a much larger



(a)

(b)

Figure 3.5: Simulated spectra for: (a) $w = 2^{n-1} + 2^{n-2}$ (b) $w = (2 + \frac{3}{4})2^{n-m}$

fundamental period (i.e. much denser spur profile) the quantization noise (originating from phase truncation) is spread over a fairly larger portion of the spectrum, essentially improving the SFDR (and the Allan deviation) by a significant amount. The mathematical concepts underlying the operation of higher order modulators are laid out in the following chapter, while good reference points for deeper mathematical analysis of the Flying Adder architecture are [22] - [24].

Chapter 4

Digital $\Delta\Sigma$ Modulators for the FA Frequency Synthesizer

It has already been pointed out in the previous chapter that in order to address the unacceptable spur profile of the FA-synth output when simple 1^{st} order modulation is employed, we need to come-up with solutions that better randomize the accumulation sequence. A possible solution to this problem could be increasing the modulation order. A lot of work has been done in the field of Digital $\Delta\Sigma$ Modulators (DDSMs) since they are an essential building block in numerous applications including fractional-N frequency synthesizers, digital-to-analog and analog-to-digital converters (DACs) and (ADCs) [25] - [27]. Some of the most common types of modulators analyzed in literature are Single-Quantizer (SQ) DDSMs, Error Feedback Modulators (EFMs), and **M**ulti St**A**ge Noise **Sh**aping (MASH) DDSMs. Our focus in this work is mainly on MASH DDSM, which is most commonly used in fractional-N frequency synthesis and whose stable input range is equal to the full scale range of the modulator quantizer. In this chapter we first lay down the theoretical basis of higher order MASH DDSMs, and then incorporate them in the FA synth design in an attempt to improve the spur profile. Finally, the simulation framework that has been developed is used to examine the effect of different variants of $\Delta\Sigma$ modulators (different modulation order) on the output spectrum and the Allan deviation of the FA synth.

4.1 Theoretical Background

It has already been mentioned in chapter 3 that the initial state and the type of input (odd vs even) of a $\Delta\Sigma$ modulator can affect the number of cycles, that a modulator will go through before returning to its initial state. In that sense it is useful to think about $\Delta\Sigma$ modulators (DSMs) as finite state machines (FSMs) that must return to a state they have visited in the past, assuming they are given a constant input [26]. The more states we introduce, the larger the fundamental period becomes, and the more spread out the quantization noise will be. DSMs are deterministic in nature, which makes it possible for us to predict exactly how the noise will be spread out and at which locations spurs will appear depending on the input, initial condition, and modulator order. This motivates the use of higher order modulators even though they are expected to increase the fundamental period, hence bringing the 1^{st} spur *closer* to the carrier. Our bet will be that because the quantization noise is now distributed among many more different spurs (as many as the fundamental sequence length) the power of each individual spur will be pretty low (ideally close to the noise floor level).

4.1.1 1^{st} Order $\Delta\Sigma$ Modulator

The building block of most DDSM's is the Error Feedback Modulator (EFM). The block diagram of a first order digital EFM is shown in Fig. 4.1 . It is comprised of an m-bit digital accumulator (adder and delay cell). The outputs of the digital accumulator are the sum $s[n]$ (also m-bit) and carry $c[n]$ (single bit), where n now denotes discrete time. Assuming the input to the EFM is a constant digital codeword w, the outputs $s[n], c[n]$ are derived as:

$$\begin{aligned} s[n] &= \big(w[n] + s[n-1]\big) \bmod 2^m \\ c[n] &= \lfloor \frac{w[n] + s[n-1]}{2^m} \rfloor \end{aligned} \tag{4.1}$$

What is of interest for spur minimization of DDSM's is the fundamental period of the accumulator carry output for a given input w and initial condition $s[0], c[0]$. It has been shown in [25] that the fundamental period is independent of the initial condition and equal to at most 2^m only when the input w is an odd number. In general, the fundamental period, L_{fund} satisfies the condition [25]:

$$(L_{fund} * w) \bmod 2^m = 0 \tag{4.2}$$

The FA synthesizer using a first order EFM modulator has been theoretically analyzed in 3.2, 3.3, where simulation results have also been provided, motivating the investigation of higher order MASH modulators.

4.1.2 Higher Order MASH $\Delta\Sigma$ Modulators

The general block diagram of a higher order MASH $\Delta\Sigma$ modulator is presented in Fig 4.2 . It consists of a cascade of first-order EFMs, where the sum output of each EFM is an input to the next one. The carry output of each stage is back propagated through a noise cancellation network to produce the final output of the modulator. This noise cancellation network rejects the quantization noise of all the modulators in the chain but the very last one, significantly improving the quantization noise level. However, it also the imposes a constraint on the maximum frequency of the synthesizer by reducing the critical path of the integer accumulator. Notice that the output range of the 2$^{\text{nd}}$ and 3$^{\text{rd}}$ order DDSMs is $\{-1, 0, 1, 2\}$ and $\{-3, -2, ..., 3, 4\}$ respectively, compared to $\{0, 1\}$ in the case of the 1$^{\text{st}}$ order DDSM. This effect will be explained in more detail in chapter 5, and is one of the main drawbacks of using higher order modulators in the fractional part of the FA

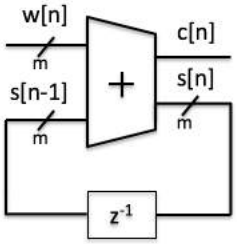


Figure 4.1: 1$^{\text{st}}$ order EFM Block Diagram

RF-synth.

The main advantage of higher order modulators is the increased length of the period of the carry out sequence. It has been shown empirically (through a large number of simulations) in [26] and proved mathematically [25] that the minimum fundamental period of the carry out sequence is guaranteed to be achieved when the initial condition of the first accumulator in the MASH chain is odd. The result is tabulated in table 4.1 and can be compared with the one given by eq. (4.2) for the 1st order DDSM output sequence length, which only reaches a maximum cycle length of 2^m for odd input values.

The effect of longer periodicity on the spur profile of the Flying Adder synthesizer is the reduction of the spurious tones and the subsequent improvement of SFDR, since now the accumulation operation is better randomized and the quantization noise is distributed over a larger portion of the output spectrum. It should be noted however, that this improvement comes at the expense of a slightly increased noise floor. Simulation results using the Simulink models of the FA synthesizer that support this point are provided in the next section.

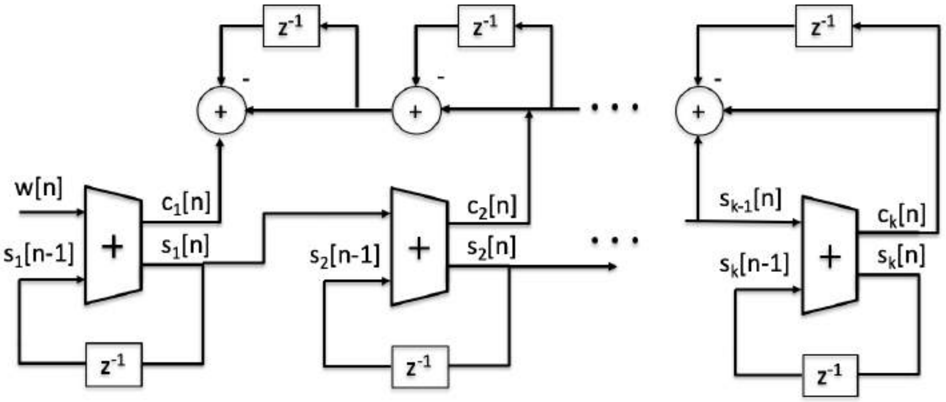


Figure 4.2: Higher order MASH block diagram

Table 4.1: Modulator Output Sequence Lengths

Modulator order	Initial Condition	Guaranteed sequence length	Maximum sequence length
2	$s_1[0]$ odd	2^{m-1}	2^{m+1}
3	$s_1[0]$ odd	2^{m+1}	2^{m+1}
4	$s_1[0]$ odd	2^{m+1}	2^{m+2}

4.2 FA Frequency Synthesizer with Higher Order MASH Modulators

In this part we will first examine how the spectra of 2nd and 3rd order modulators compare against the simple 1st order case when they are used to perform the phase accumulation operation of the FA synthesizer. Subsequently, we will evaluate how this difference in performance is reflected upon our basic metric, Allan Deviation. The model used for these simulations is shown in Fig. 4.3. It is identical with the 1st order model in Fig. 3.3 with the addition of a fractional phase accumulator that is implemented as a higher order MASH $\Delta\Sigma$ modulator according to Fig. 4.2.

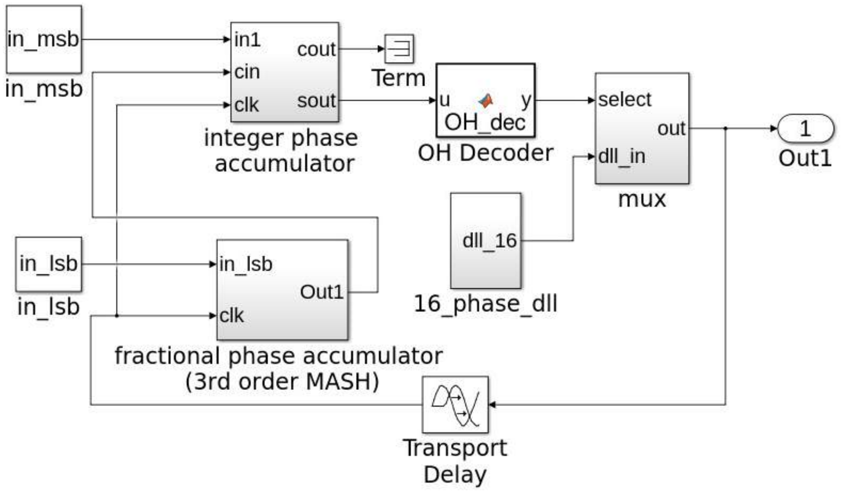


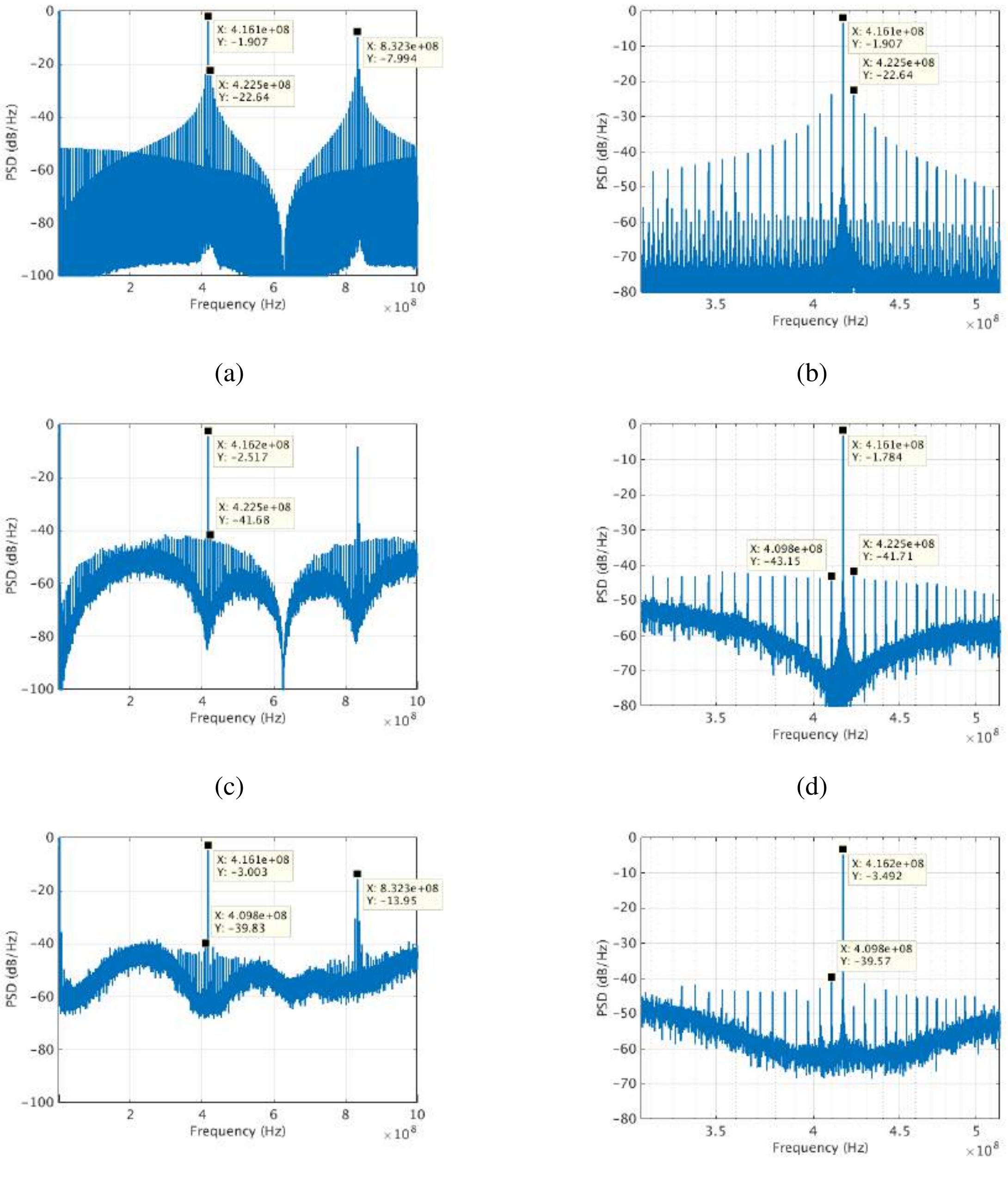
Figure 4.3: Synthesizer Model - 3rd order $\Delta\Sigma$ phase accumulator

4.2.1 Improved Spectra

The comparison will be made for the test input $w = 2^{n-1} + 2^{n-2} + \lceil \frac{2^n}{1051} \rceil$, which was used in chapter 3 to illustrate the problematic spur profile of the 1st order EFM based FA synthesizer. As it can be seen in Fig. 4.4, using higher order modulator to randomize the spectrum does indeed lower the power of spurious tones, increasing the overall SFDR dramatically. Modulators of 2nd and 3rd order provide a 20dB and 17dB improvement in SFDR respectively! The spurs still appear at the theoretically expected frequency since the integer input is identical. Notice though, that the spectrum in between those main spurs looks much smoother. This indicates that the $\Sigma\Delta$ modulator has essentially randomized the fractional sub-accumulator output. In both cases using a higher order modulator increases the noise floor as expected. Also of interest and to be expected is the noise shaping characteristic of the MASH modulators. In both higher order spectra the out-of-band noise floor increases as we move further away from the carrier, while previously it was decreasing. This is a well known behavior of high order MASH modulators, that push the quantization noise out of the carrier band. Perhaps one last thing to point out is that even though we noticed a huge improvement from 1st to 2nd order modulator, there is little difference in the spectra of 2nd and 3rd order.

4.2.2 Allan Deviation Comparison

Fig. 4.5 shows the corresponding Allan Deviation for the same input codeword, for all synthesizer variants and the same input codeword, $w = 2^{n-1} + 2^{n-2} + \lceil \frac{2^n}{1051} \rceil$. Simulation indicated that increasing the modulator order provided an improvement from 1st to 2nd order, while the 3rd order was slightly worst than the 2nd. The simulation was run up to only 0.5ms to avoid unnecessarily long runtimes. However, when extrapolated to 1sec all synthesizer variants easily meet the $\sigma_y = \frac{0.2*10^{-8}}{\tau}$ spec, which ensures the synthesizer contributes only 20% of the total noise budget. Note that since Simulink models run in the time-domain it is fairly straightforward to derive the Allan Deviation directly using eq. 2.8.

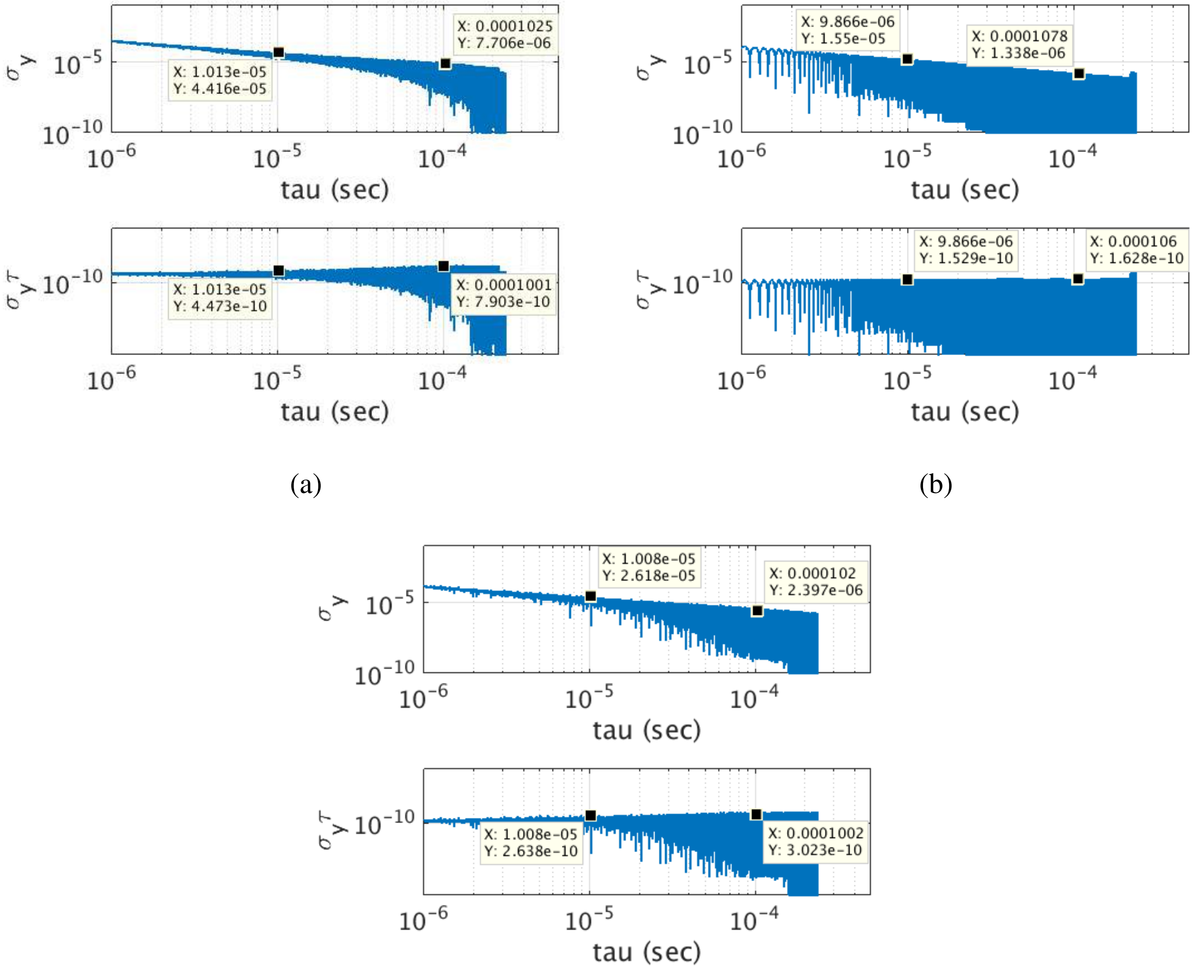


(a) (b)

(c) (d)

(e) (f)

Figure 4.4: Simulated spectrum of different synthesizer variants for $w = 2^{n-1} + 2^{n-2} + \lceil \frac{2^n}{1051} \rceil$. 1st, 2nd, and 3rd order $\Sigma\Delta$ modulators respectively ((a), (c), (e)), along with their corresponding zoom-ins ((b), (d), (f))



(a)

(b)

(c)

Figure 4.5: Simulated Allan Deviation of different synthesizer variants for $w = 2^{n-1} + 2^{n-2} + \lceil \frac{2^n}{1051} \rceil$. 1st, 2nd, and 3rd order $\Sigma\Delta$ modulators respectively shown in (a), (b), and (c)

Chapter 5

Chip Design

Having thoroughly simulated and analyzed this architecture, we finally designed a fully digital Flying Adder (FA) frequency synthesizer in a 45nm CMOS-SOI process. The chip was meant to be co-packaged with an optical SSB modulator in order to enable 1Hz resolution optical frequency synthesis over a BW of 4THz (the entire C-band). Three different modulator variants were implemented, with the modulator order ranging from 1^{st} to 3^{rd} . To bring the clock onto the chip a CML-to-CMOS clock receiver was used. It was either fed as a reference to an on chip PLL or directly to the phase generation divide-by-2 network of the FA. The system level block diagram of

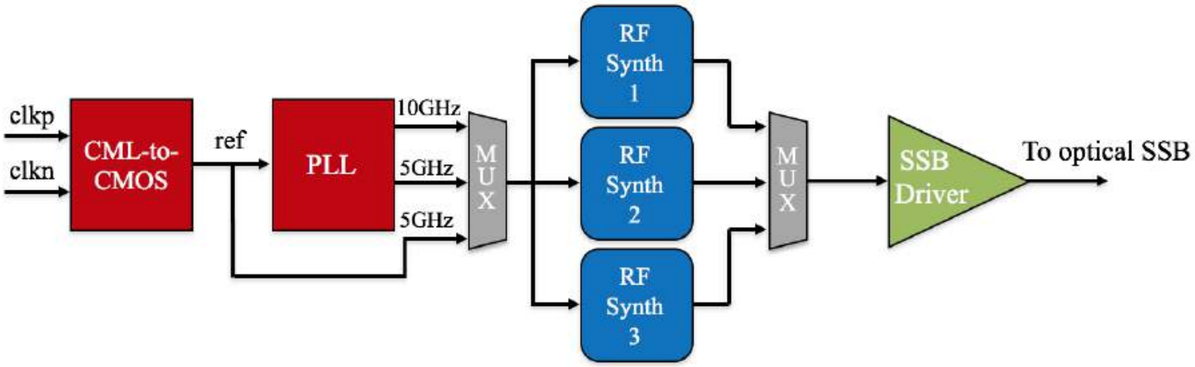


Figure 5.1: Block Diagram of Taped-out System

the chip is shown in Fig. 5.1. In this chapter, design decisions will be explained. Moving on to actual implementation, design intricacies of the different building blocks of the FA will be detailed, followed by post place and route simulations.

5.1 Design Variables

The main design choices that need to be made in this architecture are the following:

(a) *Accumulator bit count*
(b) *Modulator order*
(c) *Reference frequency, f_{ref}*

For our application, the main specs in scope are the frequency range and resolution that can be achieved. On top of that we would like the Allan Deviation, phase noise, and SFDR to also meet certain stability and spectral purity requirements. The easiest specs to satisfy are the frequency range and resolution. As seen in eq. (3.1) the maximum and minimum synthesizable frequencies are $f_{max} = 2^m * f_{ref}$ and $f_{min} \approx f_{ref}$ respectively, where m is the number of integer accumulator bits. Also, the best and worst case resolution equations (3.2), are repeated here for convenience:

$$\delta f_{min} = \frac{2^n}{2^n - 2} f_{ref} - \frac{2^n}{2^n - 1} f_{ref} \approx \frac{1}{2^n} f_{ref}$$
$$\delta f_{max} = \frac{2^n}{2^{n-m}} f_{ref} - \frac{2^n}{2^{n-m} + 1} f_{ref} \approx \frac{1}{2^{(n-2m)}} f_{ref}$$

Where n is the accumulator bit count. Recalling that our locking scheme in Fig. 2.2 locks the CW output to $\frac{f_{rep}}{4}$ we can set $f_{min} = f_{ref} = \frac{f_{rep}}{4} = 250$MHz. However, since we have already implemented a digital LC-PLL that uses an input reference of 625MHz, and which we can reuse, we set $f_{ref} = 312.5$MHz. The LC tank resonates at ≈ 20GHz and it's output is then fed to a CML divider followed by CML-to-CMOS dividers that provide 10GHz and 5GHz outputs. Using the 5GHz output and dividing it down 16 times as shown in next section 5.2 naturally gives us

312.5MHz, which is the selected f_{ref}. Note that the same locking scheme can be used. The choice of this division ratio is justified below.

Next, we need to calculate the number of accumulator bits that give us the required resolution over the entire tuning range of operation. That is 312.5MHz to 1.3125GHz, in which case $\frac{f_{max}}{f_{min}} \approx 4$. Plugging in $m = 2$ in the worst-case resolution formula above, we get that n needs to be at least 32. Note that the accumulator bit count cannot be arbitrarily big. Even though a high bit count provides higher resolution, it also limits the maximum synthesizable frequency, especially if no pipelining is employed in the accumulator data path.

The final choice to be made is the number of integer bits (or equivalently the number of reference phases). Picking more phases than what is actually needed (in this case 4 would be sufficient to get the required tuning range) helps reduce cycle to cycle jitter and improve the spur profile of the synthesized output. However, keeping the phases as uniformly spaced as possible in the time domain, sets an upper limit to their number. Simulation indicated 32 and 16 phases all running at 312.5MHz while being acceptably uniform can be generated without the use of a DLL. The table below summarizes the design variables selected for all three synthesizer variants on this chip, assuming $f_{ref} = 312.5$MHz. We intended the 3rd order to have the highest resolution at the expense of speed, the 2nd order to go faster without a big resolution penalty, and the 1st order to be able to cover any tuning range that the other two could not. The theoretical resolution presented is calculated over the required tuning range using eq. 3.3 with $m = 2$ in the worst case resolution formula.

Table 5.1: Design Choices for Different Synthesizer Variants

Modulator order	Phases	Fractional bits	Theoretical tuning range	Theoretical resolution
1	16	26	5GHz	$0.29 - 4.65$Hz
2	32	26	10GHz	$0.14 - 2.32$Hz
3	32	32	10GHz	$0.002 - 0.34$Hz

5.2 Divide-by-2 Network

The simulink model that was outlined previously in chapter 3 was used again to evaluate the effect of phase skew between the reference phases to the SNR and SFDR of the output. To do so, random skew between the phases was incorporated in the model. The non-ideal spectra had sub-dB degradation in the SFDR and 3-5dB increase in the noise floor, which was considered acceptable. Thus, a digital divide-by-two network was used for phase generation. Further validation of this scheme is provided through the post place and route simulations shown in section 5.5. This solution is much more simple, compact and easier to implement than using a DLL to produce the phases. The divider tree was implemented in structural Verilog using standard cells. Fig. 5.2 depicts the tree structure for the generation of four phases in quadrature. The 32 phases are generated by cascading 3 more divide-by-two FF stages. Special care must be taken to properly enumerate each phase coming out of the tree, such that the accumulator code will indeed select the proper phase. This enumeration is shown in Fig. 5.2 for only 4 phases.

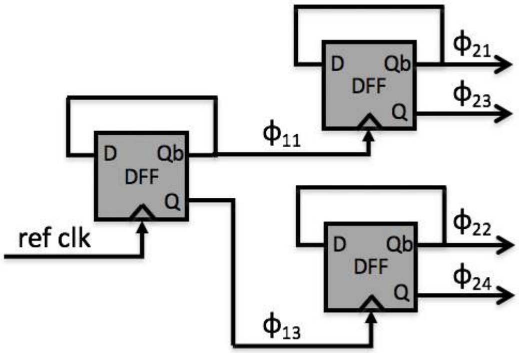


Figure 5.2: Divide-by-two network generating quadrature

5.3 Timing and Initialization

The self clocked nature of this architecture does not come without potential hazards. These issues only manifest themselves in certain “special” cases, which do not surface until extensive simulation is done and are not at all emphasized in the corresponding literature. This point further justifies the need for proper modeling and simulation of the FA synthesizer topology.

5.3.1 Glitching Mitigation and Proper Initialization

In order to ensure no metastability or glitching occurs during the transition from one phase to the other an extra latch is added after the OH decoder, making the actual implementation look like in Fig. 5.3. This latch has synchronous set and reset active low and high respectively, that are also employed to help initialize the circuit. Since this architecture is self-clocking, initialization is of great importance. To properly initialize the synthesizer, we need to ensure that one of the reference phases will be selected at start-up. This reference phase will clock the first couple of cycles until the accumulation operation begins normally, and the free-running, self-clocking mode takes over.

In this design we are using the set and reset of the glitch mitigation latch to achieve that target.

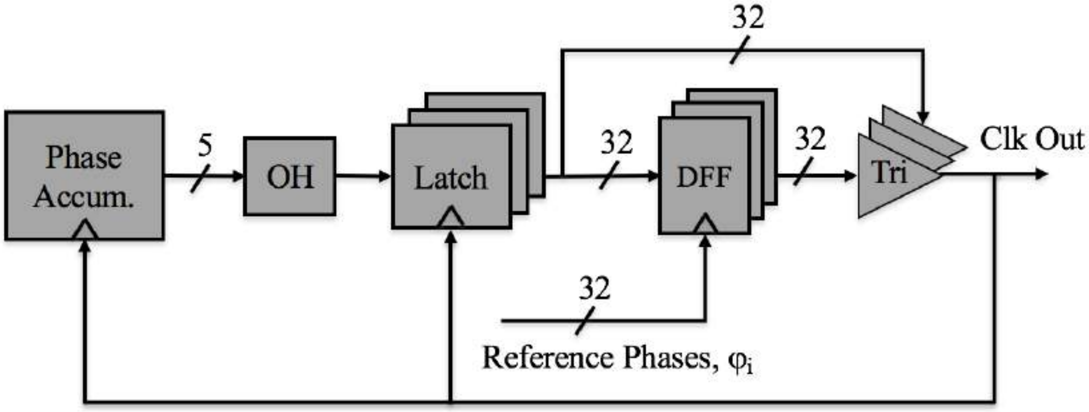


Figure 5.3: Block diagram with glitch mitigating latches in place

Let's say that we select the 2nd reference phase to start clocking the accumulator (this is a random choice since it could be any phase). To do so, we assign the set of the 2nd OH latch to $\overline{set \,\&\, reset}$, where set and reset are the global set and reset signals of the rest of the latches. At start-up, we are resetting the rest of the latches so set and reset are both high. Consequently, the set of the 2nd latch is going to be low, so that latch is set and the 2nd phase is selected, and the circuit is initialized. Note that the latch stops being set after reset is no longer asserted (i.e. during normal operation).

5.3.2 Self-Clocking Intricacies

A very important detail that is not mentioned in the first work implementing phase multiplexing through the use of tri-state buffers [28] is that in the case the same phase gets selected again before it has completed one full cycle (i.e exactly one reference period, T, after it was initially selected) the synthesizer will stop clocking and the circuit as shown in Fig. 5.3 will fail, with the output staying high. To make this more clear imagine that the integer part of the input codeword, w is 16, while the fractional part is relatively small, such that for the first two cycles the carry in from the fractional sub-accumulators is 0. Assume also, that the integer part accumulator is 5 bits and all initial conditions are 0. In that case the sequence of selected phases is $\{\phi_0, \phi_{16}, \phi_0\}$. Let's say for simplicity that at t_0, ϕ_0 is selected and propagated to the output. After some $dt = t_{clk-q} + t_{tri}$, ϕ_0 propagates to clk_{out}, which clocks the accumulator making it calculate the next phase to be selected. Assuming we meet timing, the accumulator output (in this case 16) propagates to the output of the OH decoder before ϕ_{16} goes high. Since all initial conditions are 0 the output of DFF$_{16}$ is also initially 0 so the output, clk_{out} gets pulled low, through the corresponding tristate buffer. When ϕ_{16} goes high, it propagates to the output again after delay $dt = t_{clk-q} + t_{tri}$. This pulse clocks the 3rd accumulator calculation, which gives 0 again. Notice however, that now, assuming we meet timing and since a full cycle has not gone through (i.e. ϕ_0 has not had an opportunity to clock Q$_{\text{DFF0}}$ and set it to 0), the output of DFF$_0$ is still high, hence it does not pull clk_{out} low. Under these conditions, when ϕ_0 goes high, clk_{out} is simply maintained high. Consequently, no clocking pulse will trigger the accumulator to calculate the next phase to be selected, meaning that the accumulation operation will stop and the circuit will fail. These are all depicted in the timing diagram 5.4. Note, that even though diagram assumes there is no latch between the OH decoder and the flops, adding the latching stage does not affect this operation. Even though this seems like a very special case, one cannot argue to simply avoid using such an input code. It should be pointed out that this "clock killing" condition can occur any time the same phase (in the above example ϕ_0 and ϕ_{16}) happens to repeatedly get selected within one reference period window, a case pretty common for inputs that are midcode (i.e. $w_{int} = 14 - 18$ in a 5-bit integer accumulator). Neither can it be easily predicted, given that a pseudorandom accumulator output can lead to this case even after a long number of cycles. Thus, an actual solution that eliminates this problem is in order.

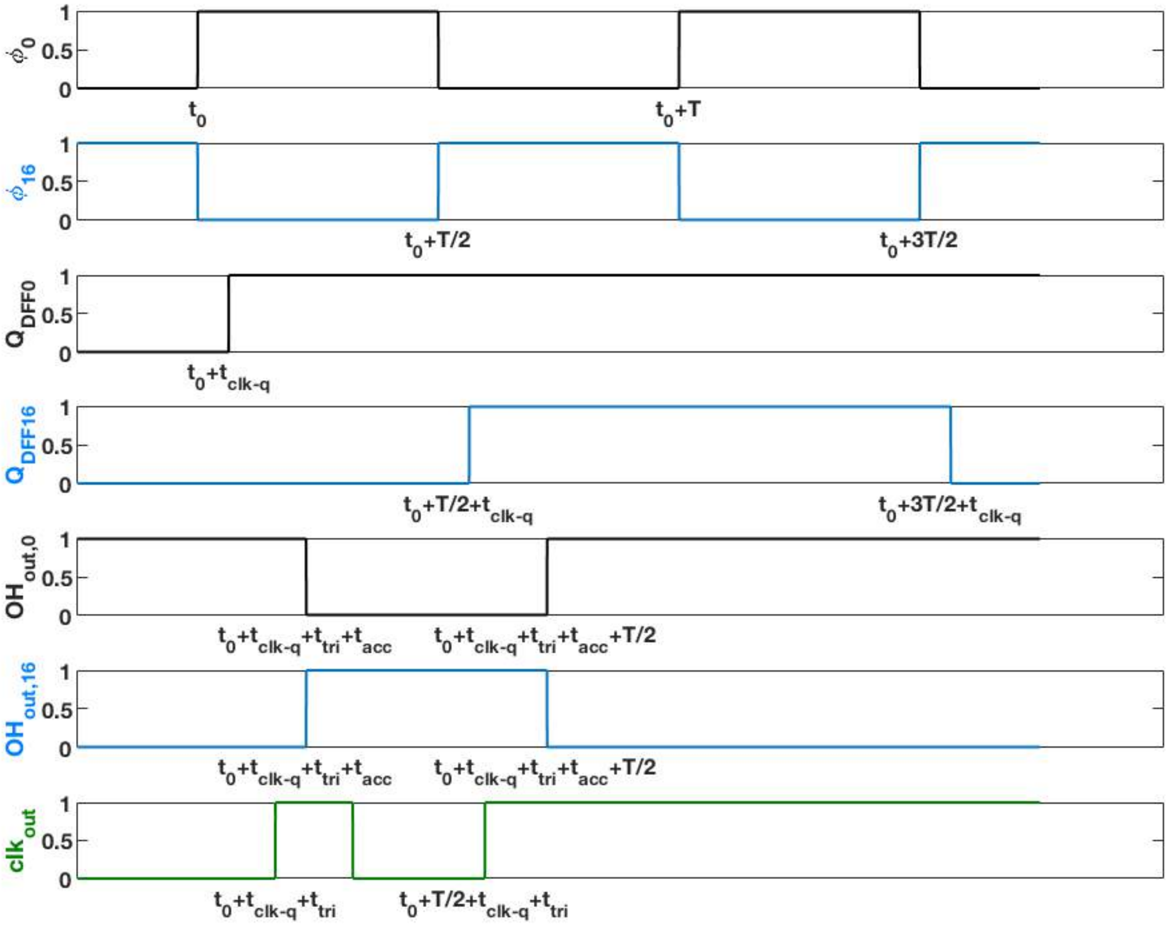


Figure 5.4: Self-clocking failure timing diagram

An elegant way to solve this issue is to add an AND gate prior to the input of each tri-state buffer as depicted in Fig. 5.5. This way, in the case when a reference phase has not had the time to clock its corresponding DFF and pull the synthesized clk output low, we rely on the reference phase itself to bring the synthesized output low during its low phase. The above are illustrated in the timing diagram 5.6, which shows the phase ϕ_0 pulling clk_{out} low through the AND gate. Note that under normal operation the path from the reference phase through the AND gate and the tri-state buffer does not affect the output in any way (i.e. glitching) since the corresponding DFF output will be low.

An alternative solution has been proposed in [28], [29] with the use of differentially clocked DFFs, which ensure that DFF outputs will be pulled low on the falling edge of the phase of interest. Thus, when this phase is selected next it is going to cause a positive pulse that will trigger the accumulator and the phase selection operation will continue normally. However, this point of failure was not as thoroughly explained in [28], [29].

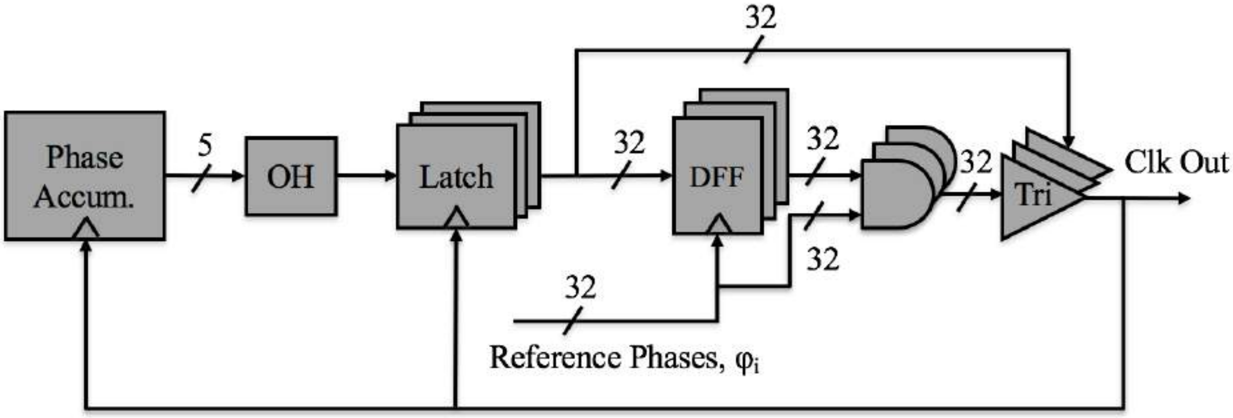


Figure 5.5: Block diagram with AND gates addressing self-clocking failure

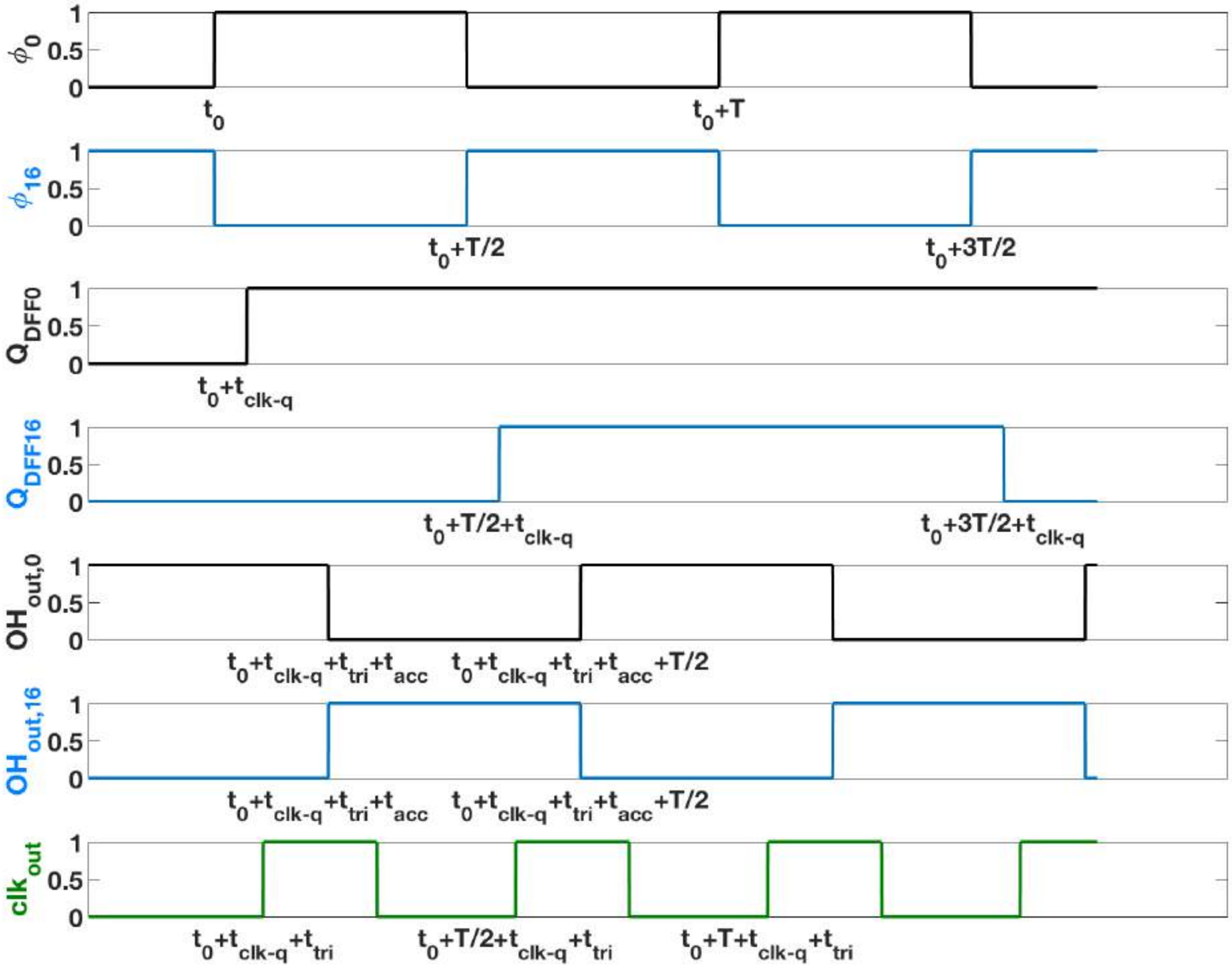


Figure 5.6: Self-clocking timing diagram with AND gates in place

5.4 Chip Layout

A synthesizer layout is shown in Fig. 5.7. This block is only 100 x 140μm^2 justifying the use of this fully digital architecture to generate multiple independent clock domains out of a single reference PLL in a power and area efficient manner. The layout of the entire system outlined in Fig. 5.1 is illustrated here in Fig. 5.8.

VDD:
VSS:
sim_clk
synth_clk_div2
synth_clk

Figure 5.7: Synthesizer layout

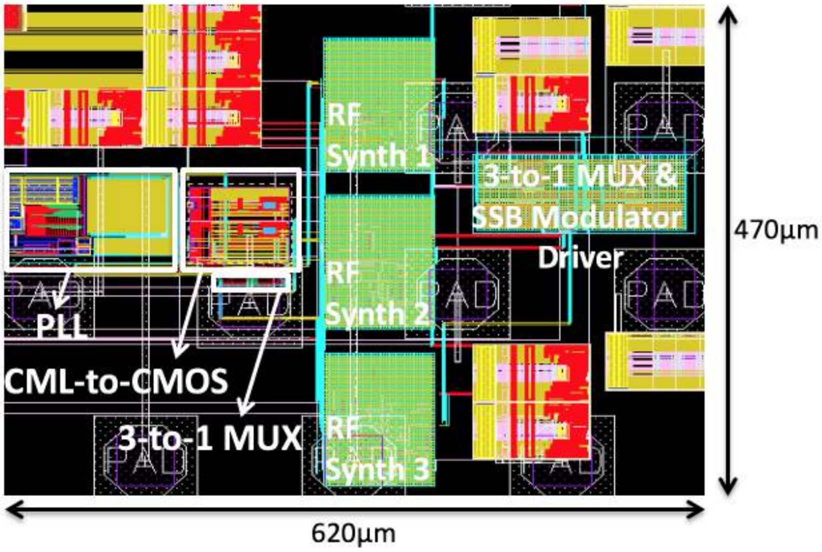
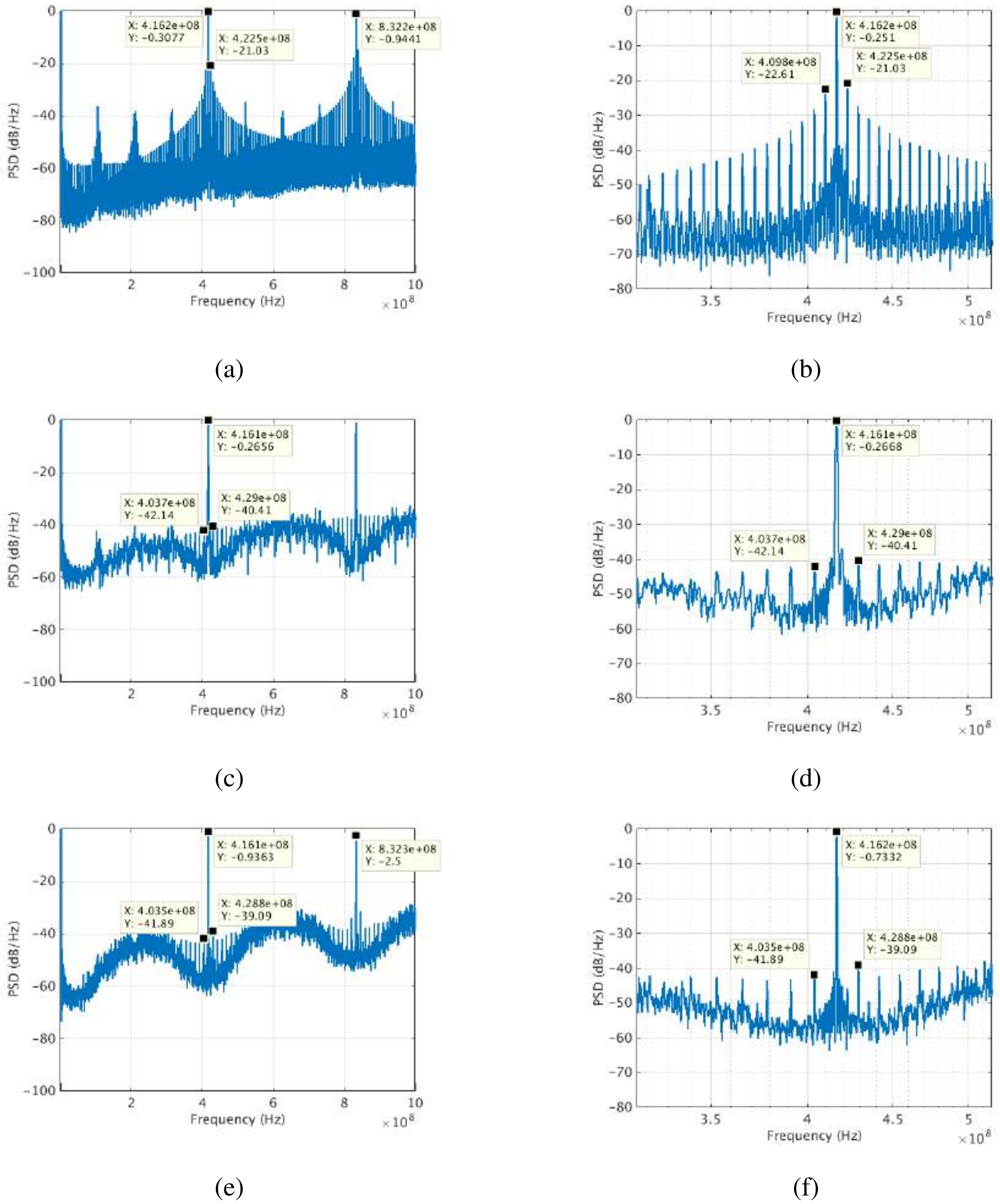


Figure 5.8: Chip layout

5.5 Post Place and Route Simulations

The last verification step to ensure proper functionality of the implemented circuit was to run post place and route (pnr) simulations. Comparing the post pnr sims shown in Fig. 5.9 with the ones run using our matlab models in chapter 4, Fig. 4.4 we see that the spur level is only degraded 1-2dB. At the same time, the noise floor is somewhat increased (5-10dB degradation) which can be mainly attributed to the non-uniform spacing of the phases. This is acceptable since the output signal will be subjected to more filtering through the SSB driver circuit.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.9: Post place and route simulation of different synthesizer variants for $w = 2^{n-1} + 2^{n-2} + \lceil \frac{2^n}{1051} \rceil$. 1st, 2nd, and 3rd order $\Sigma\Delta$ modulators respectively ((a), (c), (e)), along with their correpsonding zoom-ins ((b), (d), (f)).

Chapter 6

Preliminary Experimental Results

In this chapter preliminary measurement results from the chip that was taped out in a 45nm CMOS-SOI Global Foundries process the will be presented. Testing is still ongoing, but results from the 1^{st} and 2^{nd} order modulator variants indicate that the required tuning range and resolution have been accomplished. A photograph of the taped-out chip is shown in the Fig. 6.1. The 3 x 3 mm^2 area was shared with another project, so only the right half of the depicted chip is used to implement this system.

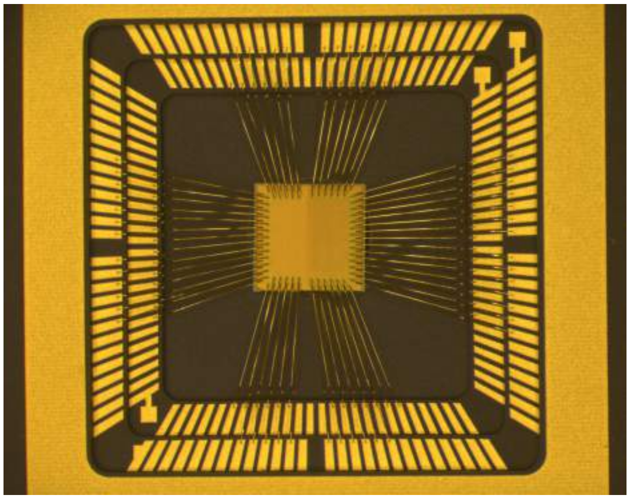


Figure 6.1: Photograph of the packaged and wirebonded chip

So far the PLL has not been brought up and we are directly feed a 1.6GHz clock source to the CML-to-CMOS block. Since we have used 16 and 32 phases for the 1st and 2nd 5.1 the corresponding divider networks will divide down the CML clock input by 8 and 16 respectively. Thus the reference frequency will be 200MHz and 100MHz respectively.

6.1 1st order $\Sigma\Delta$ Modulator

The main use of the 1st order modulator was to ensure that we cover the entirety of the required tuning range and to compare against the higher order modulators. First we wanted to check the functionality of the synthesizer so we attempted to synthesize a random frequency. We set the input codeword to $w = 2^{28} + 2387291$ since 2387291 is a prime number and would be a good worst case test for the synthesizer close-in spur profile. Fig. 6.2 shows the synthesized spectrum. The theoretically expected synthesized frequency is $f_{th} = \frac{f_{ref}*2^{30}}{w} = 792,948,086$Hz, which matches the measurement.

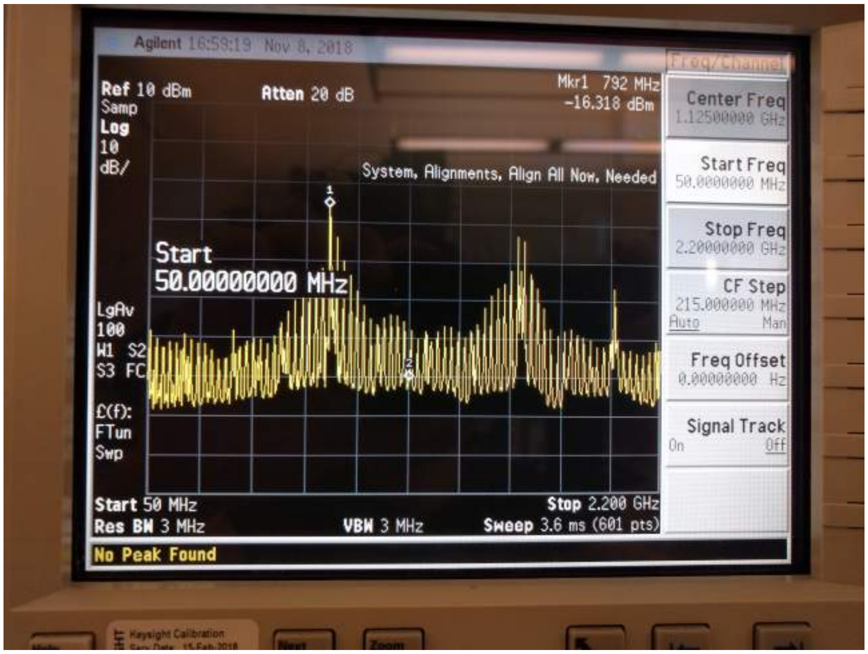
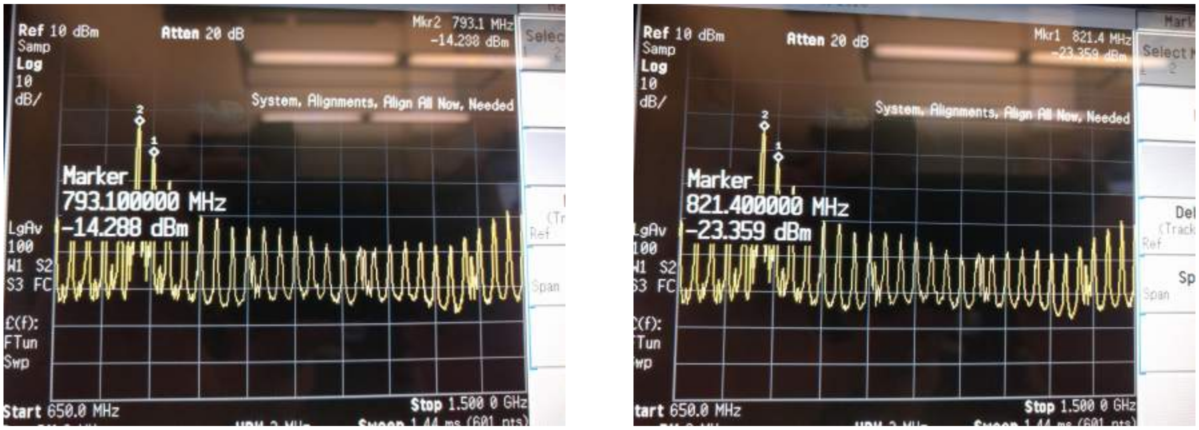


Figure 6.2: 1st order modulator with input $w = 2^{28} + 2387291$, $f_{th} = 792,948,086$Hz



(a) (b)

Figure 6.3: Zoomed-in spectra of Fig. 6.2 showing the carrier (a) and closest spur power (b)

A zoom-in of this measurement in Fig. 6.3 shows a close-in spur at 29MHz and -9dB. This result validates the developed theory in chapter 3 regarding the location of close-in spurs, but also shows the SFDR is worst than expected. Hence, we need to rely on higher order modulator variants for higher spectral purity.

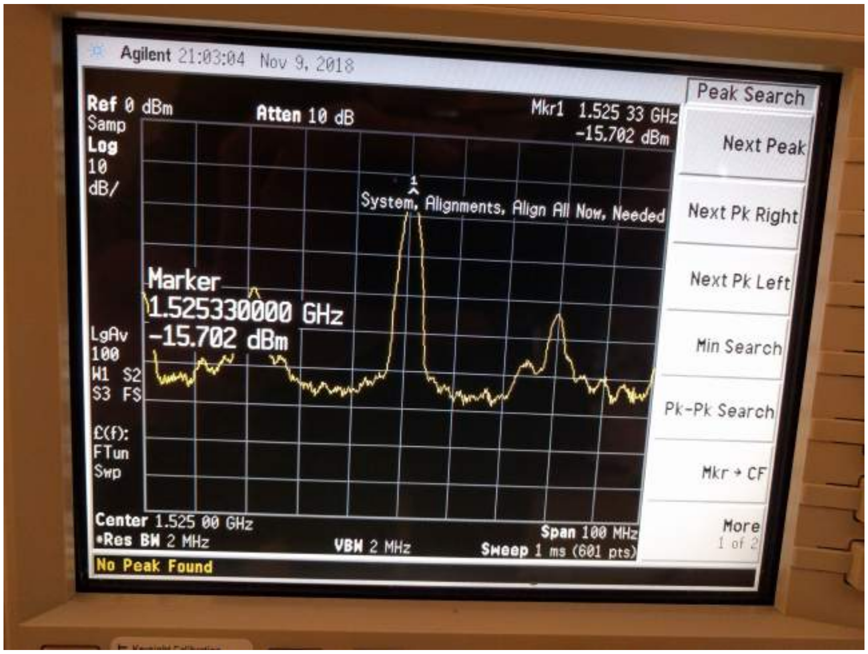
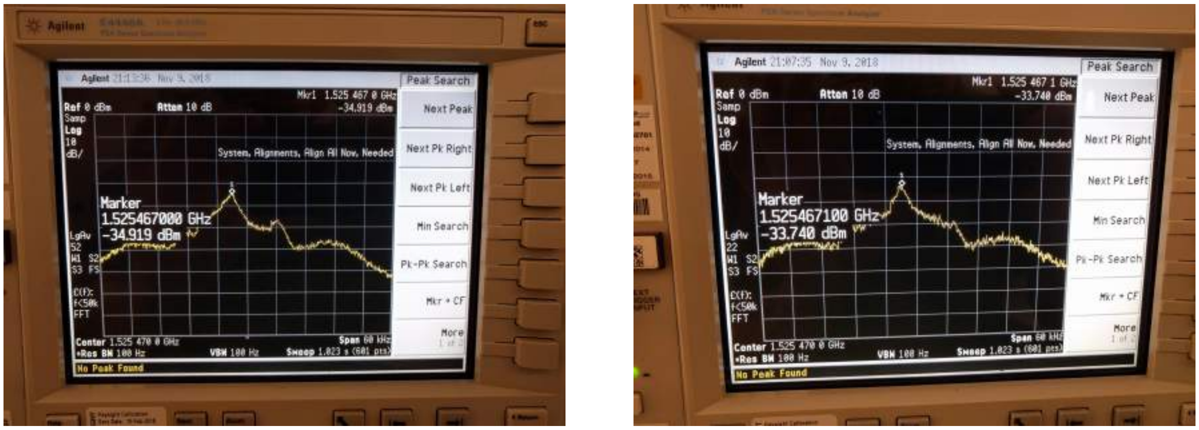


Figure 6.4: 1^{st} order modulator with input $w_{high} = 140799305$, $f_{th} = 1.525$GHz



(a)

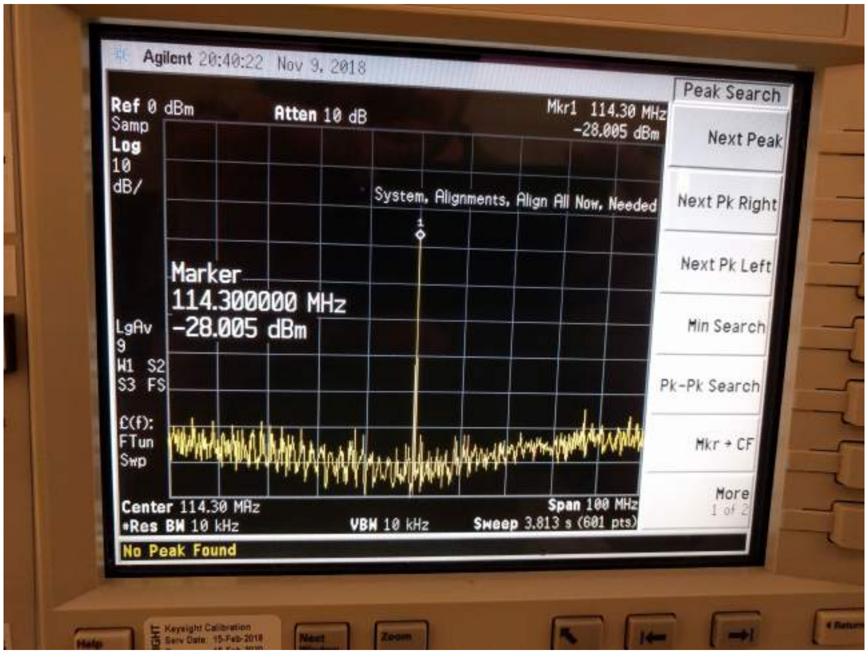
(b)

Figure 6.5: Zoomed-in spectra used to estimate the synthesizer resolution. (a) $w = w_{high} + 32$, and (b) $w = w_{high}$

The highest synthesizable frequency (up to which timing in the accumulator was met) was found to be ≈ 1.525GHz, as shown in Fig. 6.4. The corresponding codeword was $w_{high} = 140799305$. The worst case resolution was indirectly measured at that frequency (highest end of the tuning range) by changing the input codeword by 32 and observing the change in the carrier frequency. The difference observed was 100Hz as illustrated in Fig. 6.5, while the spectrum analyzer span for this measurement was set to 60kHz. This indirect measurement yields an expected worst case resolution of $\delta f_{out} \approx \frac{100}{32} = 3.125$Hz.

6.2 2$^{\text{nd}}$ order $\Sigma\Delta$ Modulator

Moving on to the 2^{nd} order synthesizer variant we started by testing a fractional input code, $w = 1879050240$ with $f_{th} = 114.286$MHz to verify the operation and check the performance in terms of SFDR. The measured spectrum shown in Fig. 6.6 has an SFDR > 50dB and validated the proper operation of the synthesizer. The small offset that is observed between the theoretical and measured carrier frequencies in both 1$^{\text{st}}$ and 2$^{\text{nd}}$ order variants can be attributed either to lack of calibration in the clock source, skew between the reference phases that are generated by the divide-by-two



Agilent 20:40:22 Nov 9, 2018

Ref 0 dBm Atten 10 dB Mkr1 114.30 MHz -28.005 dBm

Samp
Log
10
dB/

System, Alignments, Align All Now, Needed

Marker
114.300000 MHz
-28.005 dBm

LgAv
9
W1 S2
S3 FS
£(f):
FTun
Swp

Center 114.30 MHz Span 100 MHz
#Res BW 10 kHz VBW 10 kHz Sweep 3.813 s (601 pts)

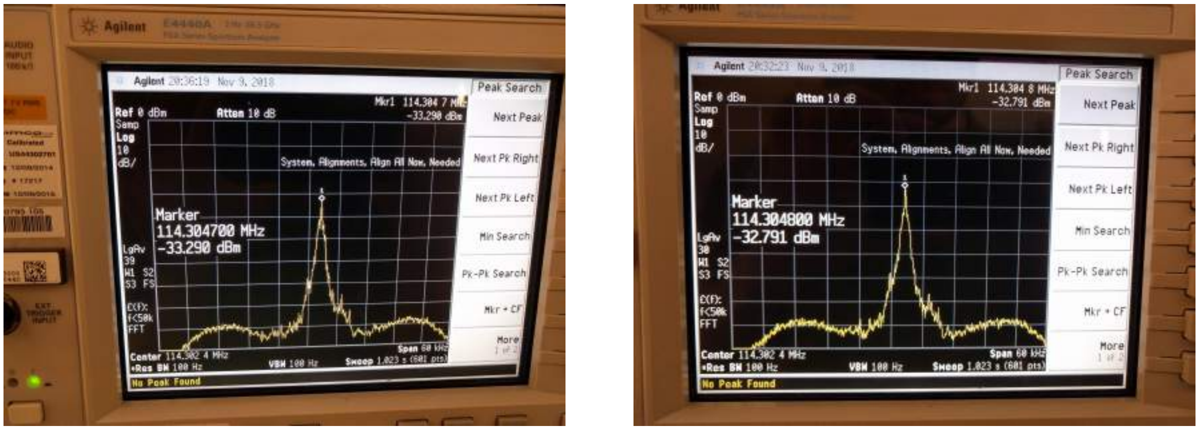
No Peak Found

Figure 6.6: 2^{nd} order modulator with input $w = 1879050240$, $f_{th} = 114.286$MHz

network, or inaccuracy in the measurement. On the up-side, this offset appears to be constant; thus we should be able to calibrate it out without any effect on the resolution or stability of the synthesized clock.

The same method as in the 1^{st} order synth was used to evaluate the resolution of this variant as well. However, in this case in order to observe a change of 100Hz in the synthesized output frequency we had to vary the input code by 1024, yielding an expected frequency resolution of $\delta f = \frac{100}{1024} = 0.097$Hz! This measurement is illustrated in Fig. 6.7. Admittedly this resolution is only valid for the lower range of the tuning range.

We found that the highest f_{out} we can get out of this variant $\approx$ 1GHz, as shown in Fig. 6.8, for $w = 107,395,661$. The observed reduction compared to the 1^{st} order counterpart is due to the increased accumulator bit-count. Running the same experiment we found that the worst case resolution was $\delta f = 100/128 = 0.78$Hz (an input code change of 128 incurred a frequency change of 100Hz), still well within the 1Hz spec. As we can see the noise floor of $\approx$ 50dB was maintained but some unwanted spurs came up at $\approx$ 3MHz offset from the carrier and -22dBc.



(a)

(b)

Figure 6.7: 2nd order synth resolution measurement. (a) $w = 1879050240 + 1024$, and (b) $w = 1879050240$

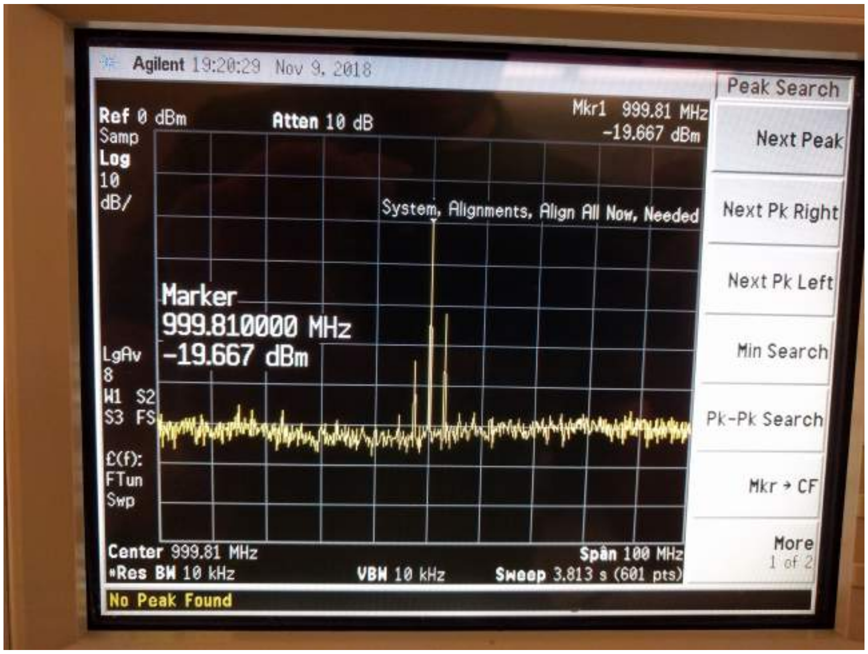


Figure 6.8: 2nd order synth with input $w = 1879050240$, $f_{out,max} = 999.8$MHz

6.3 Future Measurements

Thorough characterization of the SFDR and resolution over the entire tuning range is the next step in this ongoing testing effort. On top of that, we have to bring up the PLL and measure the total power dissipation which is expected to be lower than traditional fractional-N PLL approaches. Finally, we need to further evaluate stability by measuring jitter, allan deviation and phase noise of our synthesized output both for typical and pathological input codes for all synthesizer variants. Essentially, the goal is to validate through measurements that this architecture is a solid option for anyone who wishes to trade-off spectral purity for area and power efficiency as well as design agility.

Chapter 7

Conclusion

In summary, a fully digital RF frequency synthesizer has been designed. The resolution, tuning range and stability requirements of this circuit have been derived, with the consideration that it will be used to enable locking of a CW laser, thus implementing a direct on-chip digital optical synthesizer. In the process, a useful translation from the long-term time domain stability metric of Allan Deviation to the frequency domain metric of phase noise and the short-term time domain metric of jitter has been presented. Subsequently, the selected architecture of the Flying Adder frequency synthesizer has been analyzed and simulated. Useful guidelines and intuition behind the expected spurious content of the synthesized output have been developed, attempting at the same time to refer back to the Allan Deviation requirement. The design procedure along with some intricacies of this self-clocked fully-digital topology have been outlined. Finally, some preliminary measurement results have been presented indicating that the designed synthesizer can meet the required tuning range and frequency resolution, while testing is still ongoing.

Apart from enabling fine tuning of the optical frequency synthesizer, to a 1Hz resolution over a 4THz tuning range, this chip constitutes the first fully digital implementation of a Flying Adder frequency synthesizer. This is particularly interesting since this topology can also be used to provide multiple clock domains through a single reference with great power and area efficiency in large SoC’s as well as in cognitive radio type of applications.

This report not only provides a proof of concept through implementation, but also attempts to concisely present mathematical analysis backed by a simulation framework. At the same time, it bridges fundamental stability metrics that may be relevant to frequency synthesis depending on the type of application. The method and analysis presented here provide valuable insights to designers interested in utilizing the Flying Adder frequency synthesis topology.

Bibliography

[1] P. Del'Haye, et al. "Optical frequency comb generation from a monolithic microresonator," in *Nature 450, 1214-1217 (2007).*

[2] W. Loh, P. Del'Haye, S. B. Papp, and S. A. Diddams, "Phase and coherence of optical microresonator frequency combs," in *Physical Review A*, vol. 89, p. 053810, 2014.

[3] V. Brasch, E. Lucas, J. D. Jost, M. Geiselmann, T. J. Kippenberg, "Self-referenced photonic chip soliton Kerr frequency comb," in *Light Sci. Appl.* 6, e16202 (2017).

[4] I. Coddington, W. C. Swann, N. R. Newbury, "Coherent multiheterodyne spectroscopy using stabilized optical frequency combs," in *Phys. Rev. Lett.* 100, 013902 (2007).

[5] J. Millo, R. Boudot, M. Lours, P. Y. Bourgeois, Y. Kersalé, and G. Santarelli, "Ultra-low-noise microwave extraction from fiber-based optical frequency comb," in *Opt. Lett.* [online]. vol. 34, no. 23, pp. 3707-3709, 2009. available: http://ol.osa.org/abstract.cfm?UrI=ol-34-23-3707.

[6] S. Papp, et al. "Microresonator frequency comb optical clock," in *Optica 1*, 10-14 (2014).

[7] P. J. Delfyett, S. Gee, M.-T. Choi, H. Izadpanah, W. Lee, S. Ozharar, F. Quinlan, and T. Yilmaz, "Optical frequency combs from semiconductor lasers and applications in ultrawideband signal processing and communications," in *J. Lightwave Technol.24*, 2701-2719 (2006).

[8] N. R. Newbury, "Searching for applications with a fine-tooth comb," in *Nature Photon.5*, 186-188 (2011).

[9] [Online]. http://www.menlosystems.com/assets/documents-2/FC1500-ProductBrochure.pdf

[10] D. T. Spencer, et al. “An optical-frequency synthesizer using integrated photonics,” in *Nature* 557, 81-85 (2018).

[11] M. Xin, N. Li, N. Singh, A. Ruocco, Z. Su, E. S. Magden, J. Notaros, D. Vermeulen, E. P. Ippen, M. R. Watts, and F. X. Kärtner, “An optical frequency synthesizer using an integrated erbium tunable laser,” in *Conference on Lasers and Electro-Optics*, OSA Technical Digest (Optical Society of America, 2019), paper SW4G.6.

[12] W. Allan, “Statistics of atomic frequency standards,” in *Proc. IEEE*, vol.64, pp. 221-230, 1996.

[13] J.A. Barnes, A.R. Chi, L.S. Cutler, D.J. Healey, D.B. Leeson,T.E. McGunigal, J.A. Mullen, Jr., W.L. Smith, R.L. Sydnor,R.F.C. Vessot, and G.M.R. Winkler, “Characterization of Frequency Stability,” in *IEEE Trans. Instrum. Meas.* 20, 105, 1971.

[14] D.W. Allan, “Conversion of Frequency Stability Measures from the Time-domain to the Frequency-domain, vice-versa and Power-law Spectral Densities,” [Online]. Available: http://www.allanstime.com/Publications/DWA/Conversion_from_Allan_variance_to_Spectral_Densities.pdf

[15] D. W. Allan and J. A. Barnes, “A modified “Allan variance” with increased oscillator characterization ability,” in *Proc. 35th Ann. Frequency Control Symp.*, 1981, pp. 470-475.

[16] G. Trudgen, “Variance as Applied to Crystal Oscillators,” Rakon Tutorial, [Online].

[17] “Variance as Applied to Crystal Oscillators,” Statek Technical Note 35 Tutorial, [Online]. Available: http://statek.com/wp-content/uploads/2018/03/tn35-Rev-B.pdf.

[18] W. Kester, “Converting oscillator phase noise to time jitter,” Analog Devices, Inc., Tutorial MT-008.

[19] H. Mair and L. Xiu, “An architecture of high-performance frequency andphase synthesis,” in *IEEE J. Solid-State Circuits*, vol. 35, no. 6, pp. 835-846, June 2000.

[20] D. E. Calbaza and Y. Savaria, “A direct digital periodic synthesis circuit,” in *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1039-1045, 2002.

[21] T. Rapinoja, et al., “A Digital Frequency Synthesizer for Cognitive Radio Spectrum Sensing Applications,” in *IEEE Trans. Microwave Theory and Tech.*, vol. 58, no. 5, pp 1339-1348, May 2010.

[22] P. P. Sotiriadis, “Theory of flying-adder frequency synthesizers-PartI: Modeling, signals? periods and output average frequency,” in *IEEE Trans. Circuits Syst. I*, Reg. Papers, vol. 57, no. 8, pp. 1935-1948, Aug. 2010.

[23] P. P. Sotiriadis, “Theory of flying-adder frequency synthesizers-Part II:Time and frequency domain properties of the output signal,” in *IEEE Trans. Circuits Syst. I*, Reg. Papers, vol. 57, no. 8, pp. 1949-1963, Aug. 2010.

[24] P. P. Sotiriadis, “Exact spectrum and time-domain output of flying-adder frequency synthesizers,” in *IEEE Trans. Ultrasonics, Ferroelectrics, Freq. Control*, vol. 57, no. 9, pp. 1926-1935, Sept. 2010.

[25] K. Hosseini and M.P. Kennedy, “Minimizing Spurious Tones in Digital Delta-Sigma Modulators,” Springer, 2011.

[26] M. J. Borkowski, T. A. D. Riley, J. Hakkinen, and J. Kostamovaara, “A practical delta sigma modulator design method based on periodical behavior analysis,” in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, pp. 626?630, Oct. 2005.

[27] S. Pamarti, L. Jansson, and I. Galton, “A wideband 2.4-GHz fractional-N PLL with 1 Mb/s in-loop modulation,” in *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 49?62, Jan. 2004.

[28] T. Rapinoja, L. Xu, K. Stadius, and J. Ryynänen, “Implementation of all-digital wideband RF frequency synthesizers in 65-nm CMOS technology,” in *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, May 2011, pp. 1948-1951.

[29] T. Rapinoja, Y. Antonov, K. Stadius, and J. Ryynänen, “Fractional-N open-loop digital frequency synthesizer with a post-modulator for jitter reduction,” in *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)*, May 2016, pp. 130-133.