VCO - based column-ADCs for CMOS image sensors

Tonny Do

Thesis submitted for the degree of
Master in Electrical Engineering, Informatics and Technology
60 credits

Department of Physics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2022

VCO - based column-ADCs for CMOS image sensors

Tonny Do

VCO - based column-ADCs for CMOS image sensors

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

This thesis will present a proof-of-concept VCO-based ADC which uses a highly linear feedforward VCO with a ripple counter. The VCO-based ADC is modeled and simulated, both pre-layout and post-layout. The ADC has also been produced in TSMC180nm, taped out, and measured. The proposed VCO, which is a feedforward VCO, has a slope of 1.77MHz/10mV, a linear input region of 0.45V, an dynamic range of 9.3, a maximum frequency of 89.3MHz, an average INL of 1.33MHz and an average DNL of 0.47MHz.

Aknowledgements

First and foremost, I would like to thank my supervisors, Tongxi Wang, Tore Martinussen and Philip D. Häfliger. Thanks for letting me have the opportunity to take this thesis, and for sharing your knowledge and expertise within Analog Design, scientific writing, lab-related matter and PCB-Design to help me finish the thesis. This wouldn't have been possible without you.

Thank you to the NANO Electronics Group for also providing help with laboratory equipment, fundamental ADC theory and general advice. This also includes the students belonging to said group.

I am eternally grateful to Magne S. Gjennestad, Andreas F. Haavik, Marcus S. Nitschke and William T. Pasnin for 5 memorable years together. University life has been enjoyable and easier to manage, and it wouldn't be the same without you.

My thanks also goes to the people at SEF for being welcoming and friendly, and introducing me to things that I normally wouldn't know of.

Last and definitely not least, I would like to thank my mom and dad for providing support throughout all these years. Without you, I wouldn't even be close to where I am today.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Challenges and Motivation

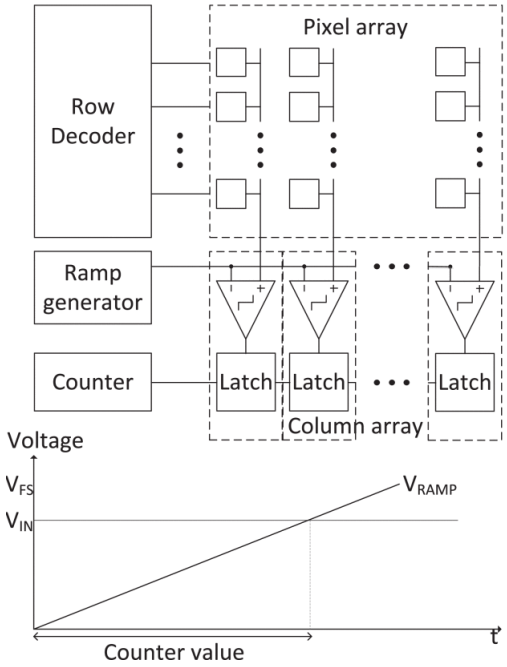


Figure 1: Block diagram of a typical image sensor with SS ADC [12]

CMOS image sensors (CIS) are widely used in many fields, such as consumer electronics, surveillance systems, automotive and so on. Inside of a CIS, a column Analog-to-Digital Converter (ADC) based readout is commonly used for it's small size and low noise readout capability.

The state-of-the-art CIS often use a Single-Slope ADCs (SS ADCs) to do Analog-to-Digital (A/D) convertions [15] [8], where each of the columns consist of comparators, counters and latches. All the column ADCs is connected to a ramp generator on the input which generates a voltage slope that the pixel voltage is compared against. (fig. 1)

The SS ADC is very popular in the CIS field due to its stable A/D conversion processing while still maintaining high fill factor for the photodiodes (PD). It also has very high accuracy due to it being an integrating converter, and also has good linearity (compared to other ADC architectures such as a pipelined ADC or cyclic ADC). It also doesn't need a spesific amount of gain, as opposed to pipeline or cyclic ADC which need exactly 2x gain.

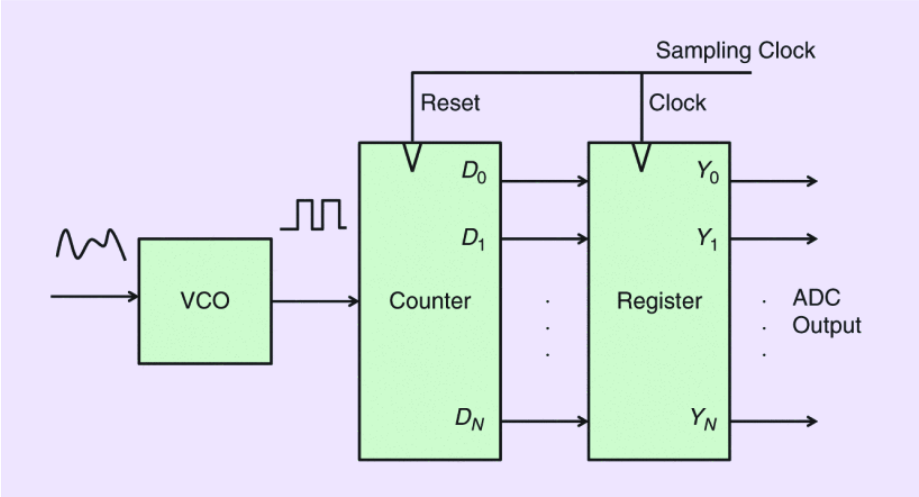


Figure 2: Block diagram of a VCO-Based ADC [3]

However, this structure has many weaknesses. For example, there is noise from the ramp generator (leads to noisy images) and high power consumption on the ramp generator. Another weakness is that the SS ADC is an integrating converter: the time T_1 the ramp generator takes generate the ramp scales linearly with the number of codes out. In other words:

$$T_1 = 2^n T_{clk}$$

where T_{clk} is the clock period of one clock cycle, and n is the number of bits on the ADC.

Other things to consider is the rapid technology scaling of the industry. Less ADC architectures become viable, as the process implemented in IC design become smaller and smaller. Since SS ADC's use comparators for each column (which are all analog circuits), it doesn't transfer well to smaller processes. Therefore, a solution that can easily transfer to smaller processes would be desirable.

A solution to these problems is to replace the column comparators with column Voltage Controlled Oscillators (VCOs), thereby removing the need of a ramp generator and its problems. The VCO used can be as simple as a ring oscillator, which is a digital circuit that can (in theory) be implemented in every process.

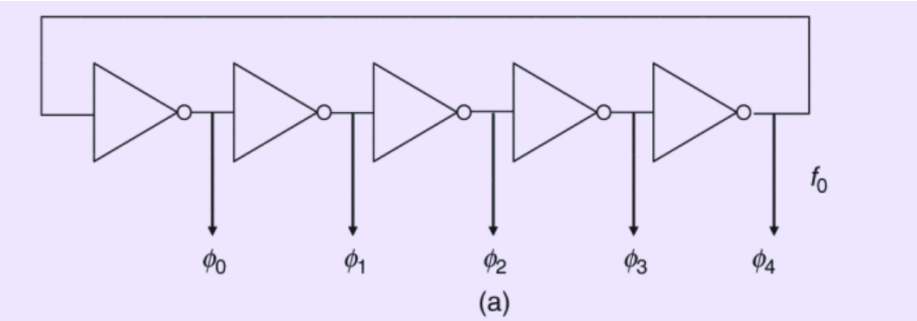


Figure 3: A conventional Ring Oscillator [3]

The VCO-based ADC (VCO ADC) consists of three blocks: The VCO, the counter and the register (see figure 2). It is also possible to think of the counter and register as one block, as they're often designed and implemented together. If it is thought of as one block, then the VCO ADC consist of two blocks instead of three. This doesn't change the composition of the VCO ADC, it's just a matter of preference whether to think of the counter and register as one block or two separate blocks.
In the VCO ADC (figure 2), the input voltage modulates the frequency of the oscillator, and its pulses is counted with a digital counter (for example, a ripple counter). Then after a sampling period of $t_s = 1/f_s$ seconds, the number of pulses in the counter is dumped into a register and the counter is reset. The value in the register is a measurement of the VCO frequency, which is a function of the input voltage (or current) [3].

Although counters and registers are relatively simple to implement in any process as they only consist of digital circuits, the same can't be said for the VCO. The architecture of a conventional Ring VCO (RVCO) is easy to make (in theory): just have $2n + 1$ number of inverters, where n is an arbitary integer connected in a loop as depicted in figure 3 (the loop being reason its called a "Ring" VCO). However, the challenge when implementing the VCO is that the it is notorious for being non-linear and suffering from phase noise and jitter, which affects the reliability of the readout. Combine that with other design constraints, such as maintaining a high readout speed, operating at low power, minimal influence from PVT variations while being small enough to fit inside a pixel column (typically 10x minimum pitch of metal 1), and the task of designing a VCO for a VCO ADC suddenly doesn't sound so easy anymore.

Therefore, this thesis aims to design a linear VCO that has reliable read-

out with minimal influence from PVT variations and noise, while running at low power and being compact in size. This will be done while also designing a counter, a register and other digital circuits and characterizing said designs in TSMC 180nm process with simulations and physical measurement in a lab.

2 Background

2.1 CMOS image sensors

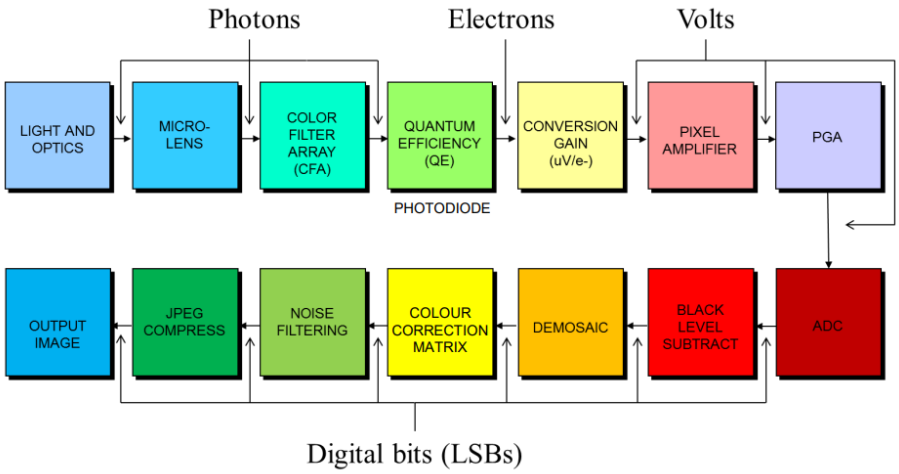


Figure 4: Block diagram of the digital camera signal chain [16]

The ADC made in this thesis is designed to be a part of a CIS. A CIS can be described in the simplest terms as a sensor that produces a digital image from the light that shines on it. Where the ADC fits in the CIS can be seen in the digital camera signal chain shown in figure 4, which shows how the whole process of the CIS. It starts with the light going through the camera optics, then through a microlens that reduces light leakage to neighbouring pixels and light loss, then through a Color Filter Array (CFA) that filters away unwanted colors before finally hitting the photodiode in the pixel. Typically the CFA is an RGB filter, but it can also be other color filters such as an CMYK filter.
The photodiode will then generate a set number of electrons, based on how many photons that hit the photodiode. How many electrons that are generated from a single photon is defined by the photodiodes Quantum Efficiency (QE), which is defined in equation 1. Here, QE is the quantum efficiency, N_{e^-} is the number of electrons and N_{photon} is the number of photons. So a QE of 2 indicates that two electrons are generated per photon (assuming that everything is ideal).

$$QE = \frac{N_{e^-}}{N_{photon}} \tag{1}$$

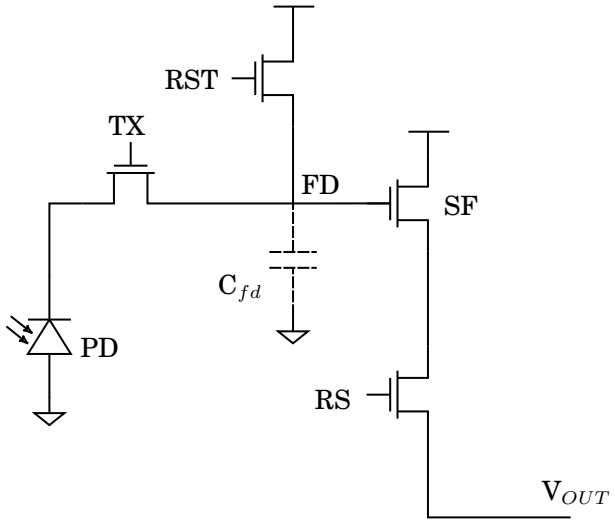


Figure 5: A single 4T pixel, including row select

From there, the electrons get transferred over to the Floating Diffusion (FD) node. Assuming that a 4-transistor (4T) pixel is used on the CIS due to better noise cancellation with Correlated Double Sampling (CDS), one has to turn on the TX transistor in order for the electrons to go to the floating diffusion (see figure 5). As the floating diffusion is floating and has a set number of electrons stored in it, one can imagine that there is a capacitor with a capacitance C_{fd} connected between FD and GND. In reality, the capacitor in figure 5 doesn't exist as a physical component, but exist in the form of parasitic capacitance (typically in the form of gate capacitance and wiring capacitance). This imaginary capacitance helps define the conversion gain (CG) of a pixel, which is how much voltage is produced per electron on FD. The formula for CG is defined in equation 2, where q is the elementary charge (approximately $1.6 \cdot 10^{-19}\ C$) and C_{fd} is the capacitance on the imaginary capacitor. A CG of $0.1\mu V/e^-$ means that when one million electrons are on FD, the voltage on FD is 0.1V.

$$C.G = \frac{q}{C_{FD}} \tag{2}$$

The voltage on FD controls the pixel amplifier, which is always a source follower (SF) which produces an output voltage proportional to the voltage. Often it's the change in FD voltage that's of interest, and the formula for that is defined in equation 3. Here, G_{sf} is the gain of SF (typically between 0.9 and 0.8 [16]) and ΔV_{fd} is the change in voltage of FD.

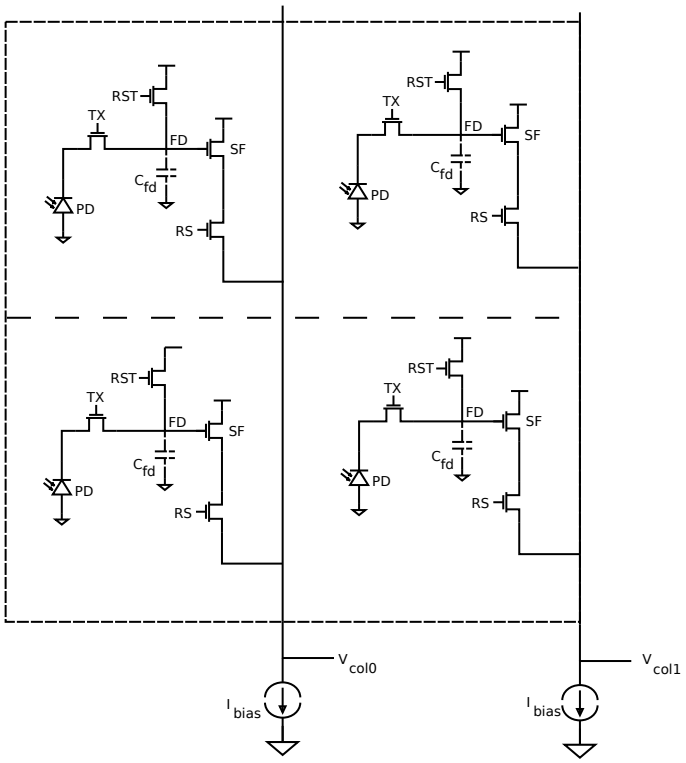


Figure 6: A 2x2 array of 4T pixels

$$\Delta V = G_{sf} \cdot \Delta V_{fd} \tag{3}$$

After the SF, the signal goes to the bitline when RS goes high, which can bee seen in figure (6). Here it's assumed that a column-level architecture is used, meaning that one row is selected at a time and read out. A bias current I_{bias} is at the end of each column, which biases the SF (RS acts as a switch). The voltage from SF is then sent through a programmable gain amplifier (PGA) that helps adjusting the gain, integration time and color response with techniques such as Auto Exposure Control (AGC) and Automatic White Balancing (AWB) before being sent to the ADC.

2.2 VCO

A Voltage Controlled Oscillator (VCO) is an oscillator whose output frequency is controlled by an input voltage. There are many ways to create this, but the one that's implemented in this thesis is a Ring VCO (RVCO). A conceptual model can be seen in figure 7, which is a ring of $2n + 1$ invert-

ers connected in feedback (n is any natural number). As long as there are an odd number of inverters in the ring and the gain of each inverter is high enough, the ring will most likely start to oscillate. More spesifically, the RVCO has a high chance to oscillate if it fulfills the Barkhausen stability criterion, which can be seen in equation 4.

$$\begin{aligned} |H(J\omega_0)| &\geq 1 \\ \angle H(J\omega_0) &= \pi \end{aligned} \tag{4}$$

The criterion is split into two requirements, with the first requirement stating that the transfer function $H(J\omega_0)$ of a single stage in the RVCO must have a loop gain equal or greater than 1, and the second requirement stating that the phase shift of the transfer function $H(J\omega_0)$ of a single stage in the RVCO must be equal to π ($180°$). This criterion must be fulfilled when designing an oscillator, or else it won't oscillate. It's important to note that the criterion is only a necessary condition, but not a sufficient one. As of 2022, no known sufficient criteria for oscillation exists [19]. This means that even if Barkhausen is fulfilled, the VCO in question might not oscillate. So the general consensus is to fulfill Barkhausen, and then do fine tuning after (given that the VCO doesn't oscillate before the tuning).

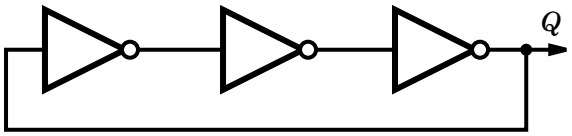


Figure 7: Figure of a conventional ring oscillator, with $n = 3$ [18]

If the RVCO from figure 7 was made with regular inverters, it would most likely oscillate, but there wouldn't be any way to control the RVCO frequency. In order to control the frequency, one has to limit the current flowing through each inverter. Figure 8 shows an inverter that has a current limiting transistor (aka. tail transistor) both on the top and the bottom, with a current mirror to bias the transistors properly. Limiting the current flowing through the inverter (i.e. limiting the crowbar current) also limits the frequency. This can be explained with equation 5 [4], where ΔV is the change in voltage on the output, Δq is the change in charge in a tail transistor, C_{node} being the parasitic capacitance on one stage in a RVCO and Δt being the change in time. If the current I stays constant (tail transistors make sure of that), then $\frac{\Delta V}{\Delta t}$ stays constant which also means the

propagation delay of the inverter stays constant (i.e. constant frequency). If I increases, $\frac{\Delta V}{\Delta t}$ also increases which in turn implies that the propagation delay decreases (i.e. an increase in frequency).

$$\Delta V = \frac{\Delta q}{C_{node}} \Longrightarrow \frac{\Delta V}{\Delta t} = \frac{I}{C_{node}} \tag{5}$$

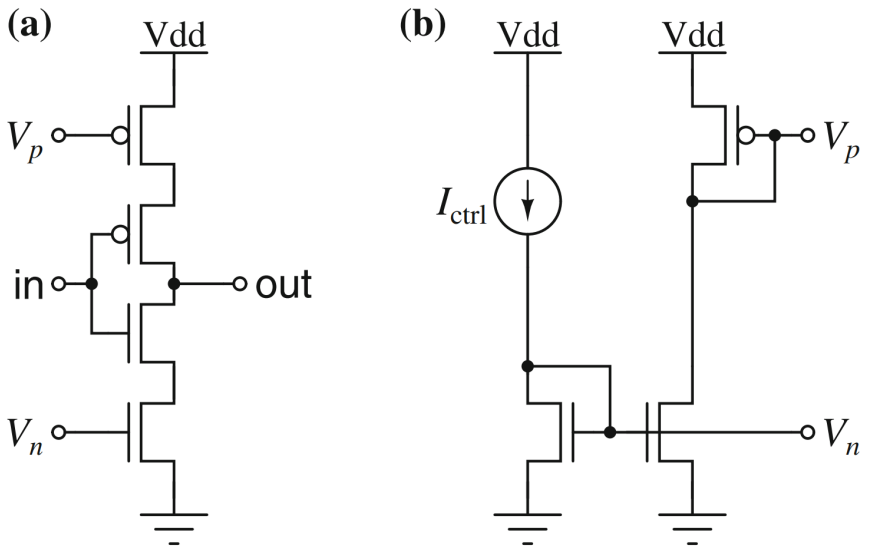


Figure 8: a) Figure of a current limited inverter, with the bottom transistor being a NMOS current limiting transistor and the top transistor being a PMOS current limiting transistor; **b)** Current mirror to generate control voltages V_n and V_p [11]

2.3 ADC

An ADC is in in the simplest terms a circuit that takes in an input voltage V_{in}, compares it to a reference voltage V_{ref} and then send a digital output word B_{out} to the output (see figure 9). To see exactly how an ideal ADC works, see figure 10. Here it's assumed that the ADC is a 2-bit ADC, meaning that the output has a maximum of $2^2 = 4$ stages: 00, 01, 10 and 11. Depending on what the input voltage is, the output will send out one of the four digital words. For example, sending a V_{in} that is $1/4$ of V_{ref} will give a digital output word of 01, and sending $V_{in} = (1/2)\ V_{ref}$ will output 10. If one applies a voltage that's slightly different from $(1/2)\ V_{ref}$ instead, say $(1/1.9)\ V_{ref}$, it will still give 10 as the output. This is because the ADC doesn't have enough *resolution* to differentiate $V_{in} = 1/2\ V_{ref}$ from

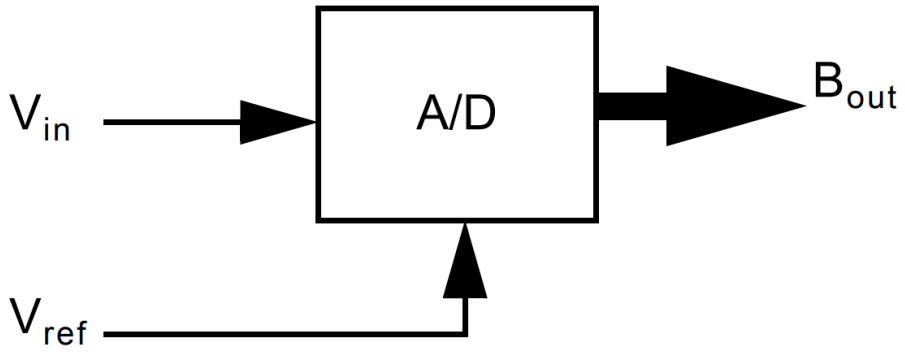


Figure 9: A block diagram representing an ADC [2]

$V_{in} = (1/1.9)\ V_{ref}$ (Resolution will be explained in chapter 4). If the number of bits on the ADC gets increased, the number of output stages increases and if the increase in resolution is enough, then one can see the difference between $V_{in} = (1/2)\ V_{ref}$ from $V_{in} = (1/1.9)\ V_{ref}$ in the digital output.

Of course, one can't make an ADC with infinite bits, which implies that there will always be a given number of output bits that an ADC can output. This in turn implies that there will always be a moment where the ADC can't see the difference between two distinctly different values: there will always be a moment where two different analog voltages will output the same digital word. This error where two different analog inputs give the same digital output is known as the *quantization error*. The more bits the ADC has, the more distinct levels are on the output and the smaller the quantization error becomes.

In order for the ADC to differentiate between two analog values, there must be a certain voltage gap between those two values. That voltage gap is referred to as V_{LSB}, the voltage required to change the output with one Least Significant Bit (LSB), which is the smallest bit in a digital word. In the case where the ADC has a V_{ref}, V_{LSB} can be calculated using equation 6, where V_{ref} is the aforementioned reference voltage and N being the number of bits of the ADC. So in the ideal ADC in figure 10, if one changes the input voltage from $(1/4)\ V_{ref}$ to $(1/4)\ V_{ref} + V_{\text{LSB}}$, then the output will change from 01 to 10. Naturally, any change in the input voltage less than V_{LSB} won't be noticed, and the output will not change.

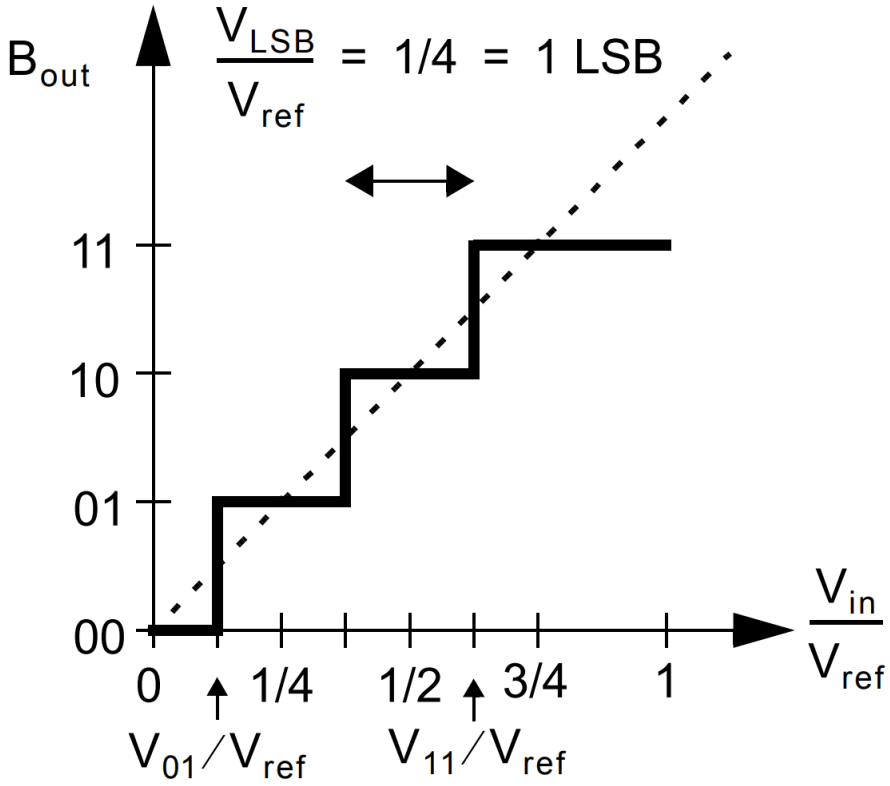


Figure 10: Input–output transfer curve for a 2-bit A/D converter [2]

$$V_{\text{LSB}} = \frac{V_{ref}}{2^N} \tag{6}$$

One thing important to note is that the digital output word from the ADC refers to a specific input voltage. If one uses the example from earlier, that V_{in} is a tiny bit larger than $1/8\ V_{ref}$. From figure 10, one can see that the dashed line is what the ADC should ideally output but instead outputs the values from the solid line. The difference between the ideal output (dashed line) and the solid line is lost in the quantization process, and the loss is often modeled as noise. This noise is known as quantization noise, which will be touched upon in later chapters.

2.4 Pulse Frequency Modulation

The VCO ADC utilises a modulation technique that works almost the same as Pulse Width Modulation (PWM). PWM is a modulation technique that represents an analog signal with a modulated square wave that encodes the information in the signal transitions. For example, say a crane operator wants to rotate a crane 90 degrees. The operator will then input 90° into the crane, which will then convert the number 90 to a pulse with a certain length. The crane will move as long the pulse is high, and the crane will stop moving when the pulse goes low. In other words, the number on the input (which dictates the angle) is translated to a pulse with a given width that will move the crane 90°. A change in the input modulates the pulse on the output, hence the name.

The VCO also uses a technique similar to PWM, called Pulse Frequency Modulation (PFM). PFM also uses the input to modulate the output, but the difference is that the information is stored in the frequency rather than the width of the signal. In order to further understand this, see figure 11. An analog signal $x(t)$ modulates the frequency of the VCO output $w(t)$, and $w(t)$ is sent to a monostable circuit that generates the signal $d(t)$, having a square pulse with fixed width t_s at every instance where $w(t)$ has an edge (rising or falling). According to [5], the signal $d(t)$ is a pulse-frequency-modulated representation of $x(t)$.

To see the difference between PWM and PFM, take a look at figure 12. One can notice that they're both similar to each other, in that they both modulate the output based on the input. However, they do differ in that the information is stored in the width for PWM and in the frequency for PFM.

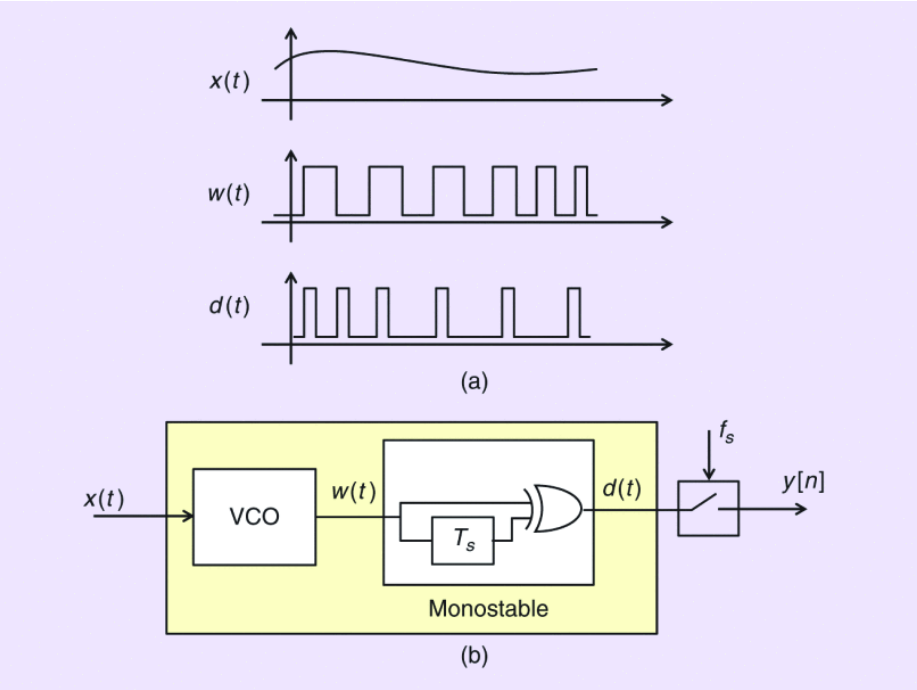


Figure 11: Time-encoding equivalent of a VCO-Based ADC: a) time-domain waveforms of the analog signal x(t), the VCO output w(t), and the PFM modulation signal d(t), and b) conceptual model of a VCO-Based ADC [3]

In the case of implementation and realisation, PFM only need a ring oscillator and digital circuits to implement PFM. This is easy to implement and transfers well to smaller processes and technologies. On the other hand, PWM requires sawtooth generators and analog filters, both which consist of opamps and other analog circuits. These circuits are quite large and consume a lot of space on the die. They also don't scale well into smaller processes.

When it comes to PWM's and PFM's attributes in the frequency domain, PFM and PWM are vastly different. A major difference between the two is that PFM exhibit first order noise shaping when sampled directly, something PWM does not [3]. For reference, see figure 12.

The slow sinusoidal input signal f_0 in 12a) is encoded with either PWM or PFM, and its frequency spectrum are shown in figure 12d) and 12e) respectively. If the PWM signal is sampled with a sampling frequency f_s, the modulation sidebands alias to lower frequencies as quantization noise. PFM avoids this problem by having periodic nulls in the frequency spectrum, as seen in figure 12e). Additionally, if the pulse widths in figure 12c)

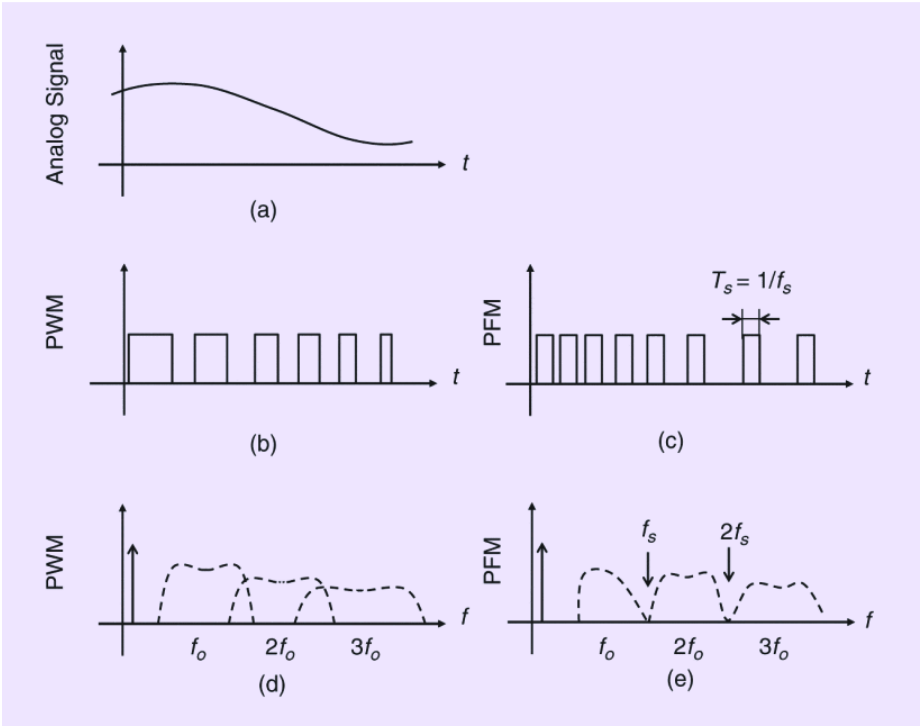


Figure 12: The time-domain waveforms and frequency spectra of a PWM- and a PFM-modulated analog signal: (a) input analog signal; (b) PWM time-domain representation; (c) PFM time-domain representation; (d) frequency spectrum of PWM signal; and (e) frequency spectrum of PFM signal. [3]

are set to be the sampling period, then all first-order nulls in the PFM spectrum in figure 12c) are aliased to DC. It is because of these first-order nulls in PFM that gives the VCO ADC its first-order noise shaping. This is a key attribute of PFM that will come in handy.

2.5 VCO-Based ADC

As described in the introduction, the VCO ADC (aka. Frequency Delta-Sigma Modulator) is an ADC that uses the VCO to encode information from input voltage to output frequency, buffer the signal, send it to a counter to count the number of rising edges (or falling edges), then dump the stored counter value into a register every sampling period t_s (see figure 2). This sampling period is the equivalent of the sampling period in any traditional ADC. Apart from that, the VCO ADC requires a somewhat different approach when implemented. For example, it's not quite obvious how the

VCO ADC gets affected by quantization noise. In traditional, Nyquist-rate ADCs (SAR, Single-Slope, Flash, Pipelined, etc.), the quantization noise gets decided by the sampling frequency and the resolution of the ADC itself (definition of resolution is covered later in chapter 3), and the resolution of the ADC is defined by the number of outputs on the SAR logic, size on shift register, length of the pipeline, etc. In a VCO ADC, which is an oversampling converter, although the resolution is decided by the number of bits in the counter, the counter doesn't directly dictate the size of the quantization noise. Typically, ADCs have a reference voltage V_{ref} which enables the ADC to fully utilize all of its output bits. As the VCO ADC doesn't have a V_{ref}, there is no certain way to use all of the binary outputs. For example, even if the counter and register consists of 100 bits, it wouldn't matter if the VCO only uses 5 of those bits. The only way to use all the bits of the counter is to either have a longer sampling frequency to let the VCO utilize most of the bits, or make the VCO go faster in a given sampling period. This implies that the VCO ADC quantization noise is not actually dependent on the number of bits on the counter, but instead the sampling period (i.e. sampling frequency) and the VCO frequency.

But what happens to the quantization noise exactly if the sampling frequency f_s and the VCO frequency f_{VCO} changes? In order to understand how f_s and f_{VCO} dictates the quantization noise, one can imagine what would happen if there is a change in f_s and f_{VCO}. If one were to increase f_{VCO} while keeping f_s constant, the resolution of the VCO ADC will increase (the more counts, more binary outputs used in the counter). If one were to do the opposite aka. oversample beyond Nyquist (increase f_s while keeping f_{VCO} constant), then the sampling period t_s would be shorter, which results in fewer counts in the register. This would seem to reduce the resolution of the VCO-ADC, as there will be fewer counts in the register. However, it's actually the opposite, and there is quite a few reasons for that:

One reason is that the error introduced by quantizing the frequency of the oscillator into an integer number (the count) is first-order noise-shaped [3], i.e the error introduced by quantizing the frequenccy into a digital value gets first order noise shaped. In order to understand this, see figure 13. The maximum total error possible is $e[n]$, which consists of $e[n]_{final} + e[n]_{initial}$, where $e[n]_{final}$ is the error injected at the end of a counting cycle and $e[n]_{initial}$ is the error injected at the start of a counting cycle. $e[n]_{final}$ and $e[n]_{initial}$ summed together cannot be larger than 1 LSB. To put it in another way, the counter can either start counting too early or ending the counting

cycle too early, and this is in worst case result in an error of one VCO clock pulse. As one VCO clock pulse is equal to one LSB, this directly implies that $e[n]$ cannot be larger than 1 LSB.

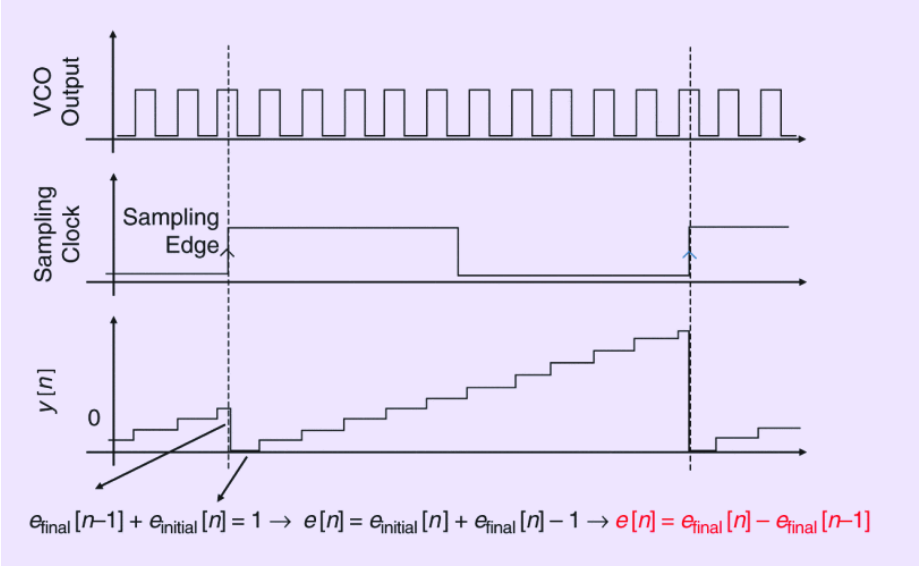


Figure 13: Signal behavior of the VCO ADC in figure 2, with an illustration of the noise shaping. The total error $e[n]$ consist of $e[n]_{final}$ and $e[n]_{initial}$, and the sum of those to can't be larger than 1 LSB [3]

Combine this with the concept of oversampling itself, which is that you can average multiple samples to reduce noise floor (assuming that the noise is white). For example, say that one sample is done in a sampling time t_s seconds. If one instead chooses to sample twice as fast (i.e an oversampling ratio (OSR) of 2), one can instead do two samples in t_s seconds. Then one can choose to average the two samples and use the average to do one conversion. What this effectively does is lowering the (white) noise floor, thereby increasing the SNR and DR. Additionally, noise shaping can also be implemented to further reduce the noise floor and improve the SNR. The oversampling itself improves the SNR with 3dB per octave, as the oversampling results in a SNR improvement of $10\log(OSR)$ [2]. If one chooses to have first order noise shaping in addition to oversampling, the result is an improvement of $30\log(OSR)$ instead [2].

An equation presented in [3] shows the maximum Signal to Quantization Noise Ratio (SQNR) of a VCO ADC with an analog signal bandwidth ABW, sampling frequency f_s, and oscillator rest frequency f_0:

$$\text{SQNR}\,[\text{dB}] \approx$$
$$6\log_2\left(\frac{2f_0}{f_s}\right) - 5.17 + 9\log_2\left(\frac{f_s}{2\text{ABW}}\right). \tag{7}$$

One can see from the equation that the SQNR improves by 6dB every time the VCO frequency is doubled relative to the sampling frequency. If the sampling frequency f_s is doubled together with the VCO frequency f_0, then the SQNR improvement is 9dB. If only the sampling frequency is doubled, the SQNR increases by 3dB, which confirmes the earlier statement.

2.6 Noise

One extremely important aspect of VCOs in general is the noise in the VCO. Although there are many forms of noise in a VCO, the most prevalent of them all is the phase noise. The output of an oscillator can be written as

$$V_{\text{out}} = A(t) \cdot f[\omega_0 t + \phi(t)]$$

where f is a periodic function in 2π, and ϕ and $A(t)$ are fluctuations in phase and amplitude due to internal and external noise sources [4]. Internal noise sources are primarily thermal noise and flicker noise from the transistors themselves, while external noise sources are noise from the supply (aptly named supply noise) and substrate noise [10]. The amplitude noise should not be an issue, as the tail transistors should act as a amplitude limiting mechanism in addition to control the output frequency (to a certain extent). The main problem in VCOs is the phase noise, since any random fluctuation in phase will affect the average frequency of the VCO and cause the counter to have too many or too few counts. To make matters even worse, phase noise persists indefinitely since all later oscillations will also be phase shifted by the same amount. Oversampling and noise shaping isn't neccesarily going to help remove the phase noise either, as the noise must white in order for oversampling and noise shaping to improve the SNDR. Luckily, some of the phase noise do in fact behave as white noise: since the phase noise causes random fluctuations in the phase, it can either cause the phase to be shifted forward or shifted back. Assuming ϕ only causes fluctuations in phase (i.e. an uncertainty in the frequency, not a frequency shift to another frequency entirely), one can see that the VCO output will still be phase shifted, but at a given frequency. For example, say the output of a VCO is 50MHz when it's noise free, but varies

between 49MHz and 51MHz when noise is added. There is an uncertainty in the measured output of the VCO, but the measurement is still within the 50MHz frequency and not at 55MHz. If oversampling and noise shaping are implemented in the system, the white noise characteristics will be less of a problem.

However, the $1/f$ noise characteristics of the VCO won't get removed with oversampling and noise shaping. Since it's low-frequent noise, sampling many times in quick succession won't get rid of it. Compared to white noise in VCOs (which has a lot of written documentation), there's not really a consistent method to remove $1/f$ noise. A common method to reduce $1/f$ noise is to increase the sizes of the transistors. This has been done with the tail transistors, which should be a decent way of compensating for $1/f$ noise. Other than that, one have the samples being further apart from each other and average those values. This doesn't seem like a bad option, given that one can afford the time to do so. Nevertheless, the $1/f$ noise is definetly something that's important to consider, especially at low frequencies [14].

3 VCO Specifications

3.1 Figures of Merit

In table 1 are the specifications that can be optimized in the VCO-ADC.

Specification	Units
Accuracy	ENOB(SNDR)
Voltage range	V
Resolution	Bits
Power Consumption	W
Dynamic Range (DR)	dB
SQNR	dB
ENOB (SNDR)	Bits (dB)
Linearity	INL and DNL
Sampling Rate	Hz
Area	mm^2

Table 1: Specifications for the VCO

All of the specs in table 1 are important to optimalize for, with certain specs being more important to improve than others. However, not all specs can be improved at the same time, as improving one spec will deteriorate another. For example, increasing ENOB might increase the power consumption, which one often wants to keep low. In other words, one must consider the tradeoffs for improving one spec and evaluate cost and benefit. For this reason, it's reasonable to review which of the specs that should be prioritized.

3.1.1 Resolution

Resolution can be bit confusing, as it can refer to the frequency of the VCO or the number of bits that the counter can display. In any case, they are both correlate with each other, since there is no need for a high-frequent VCO with a small-bit counter and vice versa. The resolution is a spec that seems reasonable to give a high priority, as it means a more detailed output. In reality, it's actually the opposite: the resolution only decides the number of binary outputs on the output, and number of bits on doesn't neccesarily have any indication on the accuracy or precision of the ADC. Accuracy is defined as how close a measurement is to the true value, and precision is defined as how close the measurements are to each other. So resolution may define how detailed the output is, but not how precise or true it is. One can have a 12-bit counter, but that doesn't matter when a constant input

doesn't give a constant output. If the output changes while the input is constant, having a 12 bit counter is meaningless as the information it displays is essentially useless. Similarly, having a VCO in the GHz range isn't useful if there is too much noise on the VCO. There is an argument for keeping the resolution high in order to reduce quantization noise, which technically improves the accuracy. However, the quantization noise is usually not the biggest noise source in an ADC, and there are other ways to reduce quantization noise without increasing resolution. Therefore, it's not extremely important to have good resolution unless the accuracy is good to start with.

3.1.2 SQNR

Signal to Qantization Nnoise Ratio (SQNR) indicates the ratio between signal and quantization noise. The quantization noise is not really noise: it's actually an error introduced during the A/D conversion due to the digital domain not having an infinite amount of values to map the analog values. To give an example, one bit in an ADC may represent 0.1V, which means that if 0.06V is sent through the ADC, the output will only represent 0.1V. This is because information from values less than 0.1V gets discarded away, and instead is used to decide whenever the output value should be rounded up or rounded down. In this case, the output got rounded up to 0.1V instead of 0V because the analog input is closer to 0.6V. This leads to an quantization error of 0.04V, which is treated, modeled and analyzed as noise.

SQNR is (in most cases) dependent on the number of bits on the output, which is again dependent on resolution. In that case, the most effective method of increasing SQNR is to increase the VCO frequency, and the number of bits on the counter (in order to display more bits). However, the quantization noise is also dependent on the sampling rate due to the VCO being an oversampling converter, as shown in chapter 2 in equation 7. Therefore, it's possible to improve SQNR without improving resolution.

As a high SQNR indicates little quantization noise per signal, it improves the accuracy of the VCO ADC and should be prioritized. It shouldn't necessarily be a top priority, but it's definitely not on the bottom.

3.1.3 Voltage Range

The linear voltage range is the span of voltages in the VCO where there exist a linear input to output relation. In other words, the linear voltage

range is a range where the input-to-output relation can be described by the function $ax + b$, where a is the gain (in this thesis, the frequency gain) and b is the minimum output frequency. Often, the input-to-output relation doesn't strictly need to fit the function $ax+b$, but can deviate to some extent from the linear function. How much it's allowed to deviate depends on how strict the requirements are. In this thesis, the goal is for the VCO to deviate less than $\pm 10\%$. The larger the linear range, the more accuracy due to an increase in the number of analog steps. Unlike the other specs, increasing the linear voltage range can be a bit tricky. The voltage range is dependent on a number of factors, which means there is no clearly defined way to improve voltage range. Still, there are certain ways to adjust the voltage range to a small degree. For example, it turns out that the linear voltage range is somewhat inversely proportional to the power consumption. Other than that, there's the transistors W/L ratio in each delay cell as well as the VCO architecture.
When it comes to its importance, one migth think that the voltage range implies what voltages the VCO ADC can take in as input. This is partially true: while the VCO ADC can only take a limited range of voltages as input, one can easily map the desired voltage span onto the the linear voltage range. For example, a CIS pixel has a output voltage range from $\approx 0.8V$ down to $\approx 0V$ [7]. That same pixel can then be connected to a buffer that remaps the output voltage range from 0.8-0V down to the linear range of the VCO ADC. But the voltage range cannot be too small, as it then will be too susceptible to noisy input signals.
As a result, the voltage range isn't a top priority, but should be taken into consideration when doing tradeoffs with other specs.

3.1.4 Dynamic Range

The dynamic range (DR) in this case, is defined as the ratio between the highest Y-value and the lowest Y-value in the linear range ($DR = \frac{\text{highest signal}}{\text{lowest signal}}$). To give an example, take a look at figure 14. On the graph, the photodiode can measure somewhere in between 10 000 and 100 000 photons before gets saturated (no matter how bright the light, the photodiode will always output a fixed amount of electrons), and that it can measure low light levels between 10 and 100 photons before read noise gets dominant (the number of output electrons mostly correlate to the read noise instead of the input photons). This is equivalent to a dynamic range of $\frac{10^4}{10} = 1000$, which is often given in dB ($20log(1000) = 60dB$). In the case of the VCO ADC, the

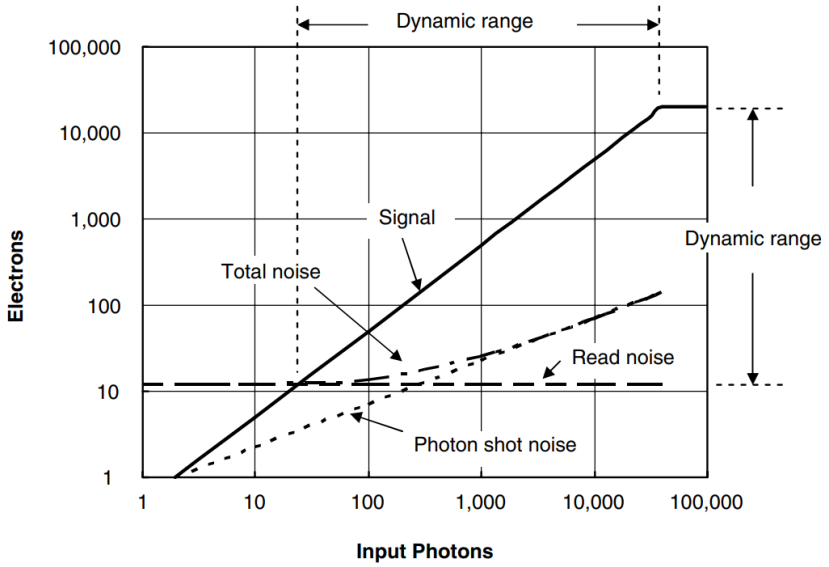


Figure 14: Graph giving a visual explanation of dynamic range (DR) [9]

dynamic range of the output frequency is the main unit of focus instead of output electrons, but the concept still stays the same.

The difference between the highest output value and the lowest output value can help improve accuracy. The higher the DR, the easier it is to contrast between high and low. Going back to the previous example, if the strongest signal measured is 10 000 and weakest is 10, it means that it's possible to see the difference between a high signal and a low signal with relative ease. In contrast, say the highest signal is still 10 000 but the minimum signal measured is 5 000. In this case, it's only possible to see the strong signals but not the weak signals.
Telling the difference between the two is a form of accuracy, but not in a traditional sense. Whereas many specs reduce noise and reduce the difference between measured value and true value, dynamic range defines the region where measurements are possible. In a VCO ADC, a high dynamic range means a big difference between a high frequency and a low frequency, which in turn implies that it's easier to differentiate between a high input voltage and a low input voltage on the VCO output.
However, it should be noted that the DR is only valuable if there is little to no noise on the output. If there is too much noise on the output, it doesn't matter what the DR is: the VCO ADC will have too much variation on the output to extract any meaningful data. Therefore, specs that contribute to-

wards decreasing noise more than DR and specs that increase the signal quality more than DR should be prioritized higher.

3.1.5 ENOB

$$\text{ENOB} = \frac{\text{SNDR} - 1.76\text{dB}}{6.02} \text{ bits} \tag{7}$$

A very important spec that should be highly prioritized is the Effective Number Of Bits (ENOB), which can be calculated with equation shown in figure 7 [2] (SNDR can be calculated with equation shown in figure 8 [2]). As ENOB is purely dependen on Signal to Noise and Distortion Ratio (SNDR, aka. SINAD), one can also list up SNDR instead of ENOB as they're practically the same. In this thesis, ENOB is used as it's easier to explain and understand, given that the thesis revolves around an ADC. What the ENOB tells is how many bits of the ADC output that contain any useful information. As an example, say a person has made a 10-bit ADC with an ENOB of 5 bits. What this means is that while the output is 10 bits, only the 5 Most Significant Bits (MSB) contain any useful information, and that the Least Significant Bits (LSB) don't contain any useful information and thus are irrelevant for the A/D convertion.
As mentioned earlier, ENOB is a very important specs and arguably the most important spec of them all. Since the spec tells how many bits contain reliable information, it also tells the accuracy of the ADC. As a result, ENOB should be among the top specs to prioritize.

$$\text{SNDR} = 10\log\left(\frac{V_f^2}{N_o + V_{h2}^2 + V_{h3}^2 + V_{h4}^2 + ...}\right) \tag{8}$$

As a sidenote, there exist a simpler model of SNDR where the distortion is excluded. This simpler model is the Signal to Noise ratio (SNR), and can be seen in equation 9 [2]. This can be used as a replacement for SNDR in the ENOB equation (equation 7 [2]), although the value of ENOB will be a bit different. Since it will be quite tedious to measure the harmonic of every output of the VCO in order to calculate the SNDR, a simple measurement of the amplitude and compare it with noise in the system seems like a decent measurement of ENOB.

$$\text{SNR} = 10\log\left(\frac{V_{x(rms)}^2}{N_o}\right) \tag{9}$$

Of course, equation 9 only works when looking at noise in the form of voltage amplitude. When looking at phase noise, which is the most dominant noise type in a VCO ADC, the equation for SNR becomes a little different. Paper [6] has derived a formula regarding how SNR due to phase noise looks like, which can be seen in equation 10. Here, K_{VCO} is the frequency gain of the VCO, A is the frequency amplitude of the VCO, L is the phase noise at a frequency offset of $_{offset}$ and f_{in} is the input-signal frequency. This equation for SNR might suit better for the purposes mentioned above.

$$\mathrm{SNR_{vpn}} = 10\log\left(\frac{P_{\Phi_x}}{P_{\Phi_{\mathrm{pn}}}}\right) \simeq 10\log\left(\frac{(K_{\mathrm{vco}}A)^2}{16Lf_{\mathrm{offset}}^2 f_{\mathrm{in}}}\right) \tag{10}$$

3.1.6 Power Consumption

When it comes to power consumption, it depends on the design that is being made. Sometimes, the focus of a design is to make a low-power circuit. In that case, power consumption should be the highest priority. Other times the focus is to make the best design possible regardless of power consumption, in which case the power consumption may be on the bottom of the list of priorities. In other words, it's dependent on the design.
In the case of this thesis, the power consumption isn't a top priority as the focus on the having an ADC which is better in size and performance than the SS ADC. However, increasing the power consumption should be justified with a reasonable decision. For instance, increasing power consumption with 500mW to increase SNDR with 3 dB doesn't sound right, as there is too much power consumption for such a little increase SNDR/ENOB. Also, more power consumption often leads to more heat generation, which is mostly unwanted in any system. As a result, power consumption is neither a top priority or a bottom priority, but falls somewhere in between the two.

3.1.7 Sampling Rate

Sampling rate is how often samples of the input signal is taken, and it's solely dependent on the sampling frequency. As mentioned in the other specs, increasing the sampling frequency increases SQNR, as well as enabling the ADC to sample signals more often. In the case of noise performance, having a high sampling rate is on the middle of the list, as there usually are better methods of improving noise performance. If one simply wants to capture many images in a short period of time, then high sampling rate is essential. But high picture capture rate is dependent on how

fast the digital signal processing (DSP) is, so there's no reason to have a high sampling rate if DSP is the bottleneck. Also, increasing the sampling frequency also increases power consumption, so one may be limited to the power budget as well. In the case of this thesis, it is assumed that the CIS isn't high speed, so sampling rate isn't necessarily that important. It belongs between the middle and bottom of the priority list.

3.1.8 Area

The definition of area in this thesis, is the area needed to implement the VCO ADC. This area is required to be of a quadratic form, as the chip used in the tapeout is of a square shape. Most chips made from multi-project wafers (MPW) are cut out square-shaped, which is why the minimum area required to produce the VCO ADC also must be defined as a quadratic figure.

The VCO ADC made in this thesis is a proof-of-concept ADC that's supposed to fit inside a CIS column. It is ideal that the analog circuitry in the CIS use minimal amount of space as possible, such that more area is dedicated to the photodiodes. This improves light sensitivity, which is the most important performance parameter of a CIS. There should be a really good reason to dedicate area for other things than the photodiode. An example where more area has been dedicated is the implementation of the 4T-pixel in conjuction with the pinned photodiode (PPD). 4T-pixel is bigger than the 3T pixel, but has the advantage of removing kTC-noise with correlated double sampling (CDS). This huge benefit is the reason why the 4T is used more than 3T, despite requiring more area. So if there are circuit solutions that significantly improve performance, then more area usage is acceptable. If not, it's most likely better to retain the photodiode area.

In terms of priority, area should be highly prioritized to enable more area for the photodiode. Additionally, the VCO ADC should fit inside a CIS column, which is another reason the area must be kept low. As long these requirements are fulfilled, one can use some space for the VCO ADC within reasonable limits. The definition of reasonable limits in this thesis is that there is a good cost-to-benefit ratio (both in terms of performance and price), and that the solution implemented must be area efficient.

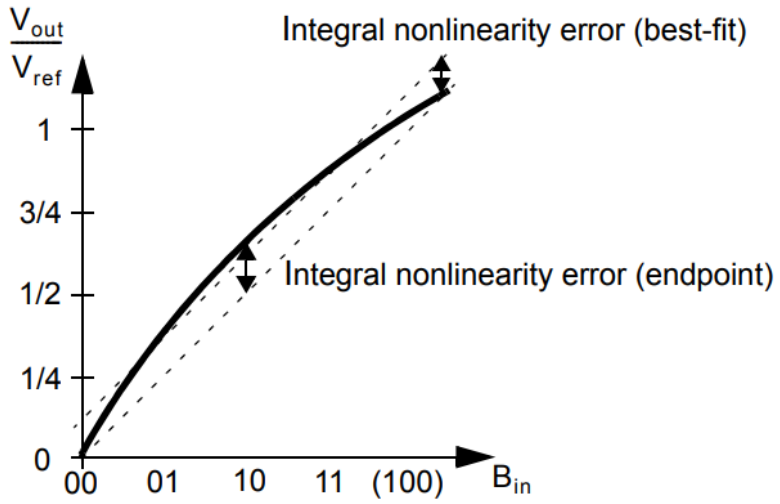


Figure 15: Graph giving a visual explanation of INL [2]

3.1.9 Linearity

Somewhat connected with the voltage range, the linearity is defined as how closely the input-output relationship between input voltage and output frequency can follow the linear function $ax + b$. Often, an ADC do have an increasing input-output relationship, and ideally the plot of an ADC should follow $ax + b$. In reality, this is often not the case as the output frequencies are a bit higher or lower than what they should've been ideally. In order to measure how good the linearity of an ADC is, one can measure the Integral Non-Linearity (INL) and Differential Non-Linearity (DNL). INL is defined as the deviation from a straigth line, which can be seen in figure 15. When measuring INL, one can either choose to have the ideal line based on the start and endpoints, or to have a best-fit line that minimizes the average INL error. DNL is defined, as the variation in analog steps away from 1 LSB [2]. For example, a DNL of 0.2 LSB has it's steps varying between 0.8 LSB and 1.2 LSB. Usually, DNL isn't evenly distributed between both sides, meaning one can have a variation between 0.7 LSB and 1.1 LSB. For this reason, some papers also specify the variation on both sides (for example, one can have a DNL of -0.8 LSB/+1.3 LSB). For this thesis, only the maximum INL and DNL will be reported. This way, one can see the maximum deviation from the ideal response of an ADC and the maximum deviation from 1 LSB.

It should be mentioned that INL and DNL doesn't neccesarily have to measure in the unit of LSB, but can also measure in units ofV_{LSB}, frequency or other units that seem fitting to linearity. This means that it's possible to measure INL and DNL of the VCO frequency, which is useful to do.

3.2 Specifications Priority list

Table 2 shows the specifications ordered list, with highest priority starting at the top:

Specification	Units
ENOB (SNDR)	Bits (dB)
Dynamic Range (DR)	dB
Size	mm^2
Linearity	INL and DNL
Voltage range	V
SQNR	dB
Sampling rate	Hz
Power Consumption	W
Resolution	Bits

Table 2: Priority list of the VCO ADC, with the top priority starting at the top of the list

ENOB is put on the top of the priority list as it defines the accuracy of the VCO ADC. DR and voltage range are very close in terms of priority, but DR is put above voltage range as differentiating high and low on the output is more important than differentiating high and low on the input. Size is also important as the VCO ADC is supposed to fit in a CIS, and the linearity is also important (Although some might call it overrated). SQNR and Power consumption are also close, but SQNR is put higher since noise performance is considered more important than power consumption. Since SQNR is prioritized higher, one has to either prioritize sampling rate or resolution over power consumption. Here, sampling rate is prioritized as it enables a higher SQNR and a faster system. Resolution simply means a fast VCO and a large counter, which isn't the most important thing to optimalize.

From the block diagram in figure 2, one can notice that all the specs are mostly dependent on the VCO. Although the counter and register do also matter for certain specs, such as power consumption and resolution, they're all digital circuits which have very little room for improvement compared to the analog blocks. As long the digital blocks in figure 2 don't bottleneck the ADC in terms of speed, most of the specs are heavily reliant on the VCO.

Therefore, most of the focus will be shifted towards the VCO (Its specs are highlighted in chapter 4).

4 Implementation

4.1 VCO

4.1.1 Design considerations

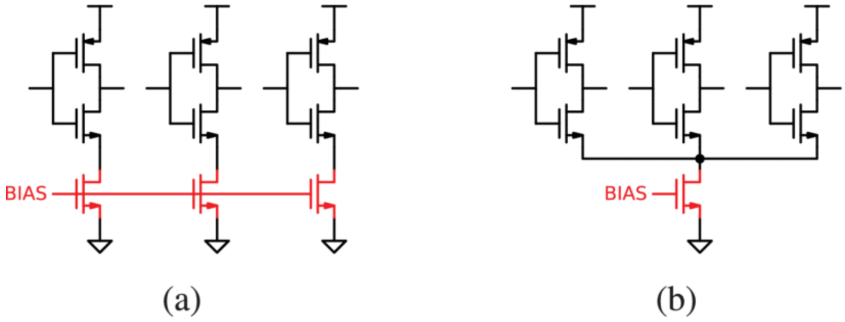


Figure 16: Figure showing the difference between **a)** individual tail transistor and **b)** shared tail transistor [1]

As with all analog circuits, one must consider the tradeoffs that comes with design choices. For instance, in a ring oscillator (3), one can use a single tail transistor as a current source on all inverters rather than one tail transistor for each inverter (see figure 16). One major advantage with that is that mismatch between transistors are avoided, as there is only one transistor instead of dozens of them. This means that there aren't any imbalances between the VCO stages due to mismatch, which would affect the frequency and phases of each stage.
A downside of the shared tail transistor is that it introduces a low-frequency pole at the drain of the tail transistor, due to the parasitic capacitance. Individual tail transistors have this problem too, but a shared tail transistor has more parasitic capacitance than one individual tail transistor, which means that the pole occurs at a lower frequency. This will limit how quickly the VCO will respond to a change in the bias voltage [1].

As mentioned in figure 8 from chapter 2, one can choose to implement a tail transistor by only using a NMOS, by only using a PMOS or by using both. Usually, the choice is between either using a NMOS alone, or using NMOS and PMOS together. Implementing tail transistors by only using PMOS is usually not considered a good option, as the NMOS takes less die

area and has better conductivity. Even if PMOS has the same conductivity as a NMOS, it still takes additional die area as a N-well is required for the PMOS. It also is more sensitive to noise on the power supply. Considering all the downsides of only using PMOS, it's safe to assume that implementing tail transistors by only using PMOS is not a valid option.
This results in using two options: either use only the NMOS, or use NMOS and PMOS together. If one were to use only NMOS, an advantage of that is that it takes the least amount of die area and is less susceptible to supply noise. The downsides is that the rise time on the output will not be equal to the fall time.
If one were to use NMOS with PMOS, rise time being unequal to the fall time is not a problem. By using biasing techniques like the one shown in figure 8, the threshold voltage effect shouldn't become an issue and will automatically give equal rise times and fall times. The downside is that implementing a NMOS with PMOS takes more die area (even without considering the implementation of the current mirror), and more susceptibility to supply noise.

Another design choice to make are the sizes on the transistors inside the inverters. Bigger width on the transistors means better performance, less mismatch and less flicker noise. Downsides of bigger transistors (bigger width and/or lenght) are more power consumption, more parasitic and routing capacitance and more die space.

Other choices to consider is how many stages to implement in a VCO. A formula often associated with RVCO is the formula of the output frequency, which is written as

$$f_{OSC} = 1/(2Nt_d)$$

where N is the number of stages and t_d is propagation delay of the unit stage [13]. One can see that fewer stages results in a higher frequency, which does makes sense as each stage adds an additional amount of paracitic capacitance. The propagation delay is mostly dictated by the dimensions of the transistors in the inverter, so the number of stages should not have an effect on the propagation delay. Based on the formula, more stages seems like a detriment in performance, but more stages decreases the gain requirement per inverter. In other words, a VCO can be driven using a lower VDD. For example a 0.6V VDD instead of 1.8V VDD, if sizing, biasing, etc. is done correctly. In short, more stages result in a lower frequency and lower power needed to optimally run the VCO.

4.1.2 Implemented design choices

In this thesis, each inverter share a NMOS and a PMOS tail transistor. This is to avoid inbalance due to mismatch, which can impact the oscillation phase of the VCO. This is done on the low side and high side to keep the balance between rise-time and fall-time, and to improve the tuning range. It's also important to mention that the tail transistor mostly operates in saturation region, which isolates the VCO from supply noise. The low-frequency pole will limit the maximum output frequency, but as mentioned in chapter 3, a fast VCO isn't a high priority. Besides, having mismatch is far worse than having a slower VCO.

The transistors in the inverters are made as small as possible to have less power consumption, more die area available and less parasitic capacitance. With proper layout techniques, one can acheive even less mismatch. Flicker noise will still be prevalent, but it's magnitude isn't big and should be tolerable to a certain extent. Besides, the upsides of having smaller transistors far outweight the small flicker noise introduced to the ADC. Smaller transistors results in less current needed to drive the inverters, less parasitic capacitance which increases the maximum frequency and less die area used, which results in more area for photodiodes, which is important for light sensitivity. All in all, a tradeoff where the ADC gets more flicker noise with all the aforementioned upsides is a tradeoff worth taking.

As the conventional VCO and the FF VCO have different number of stages, the reasoning of the number of stages will be presented in their respepective subsections.

4.2 Overview of system

A system overview of how the produced chip is supposed to look like can be seen in figure 17. The switches are realized with transmission gate, that consist of unit-size NMOS and PMOS. The smallest inverters consist of unit-size CMOS devices (more on unit-size CMOS devices/transistors in chapter 4.3.1), the second largest inverters consist of CMOS with three times the W/L ratio than the smallest inverters, and the largest inverters are three times the W/L ratio than the second largest inverters (i.e. nine times W/L the ratio than the smallest inverters).

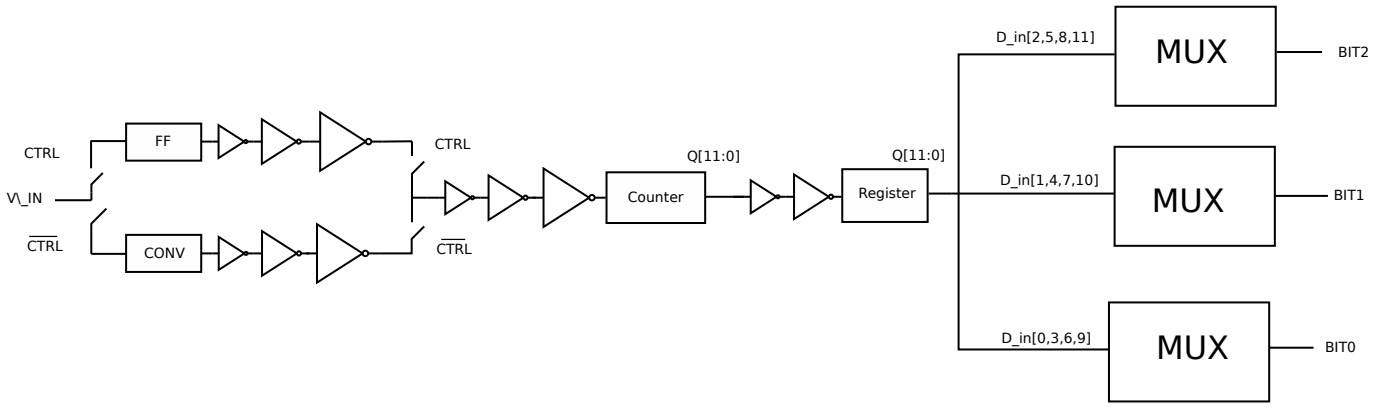


Figure 17: Top level overview of the entire chip

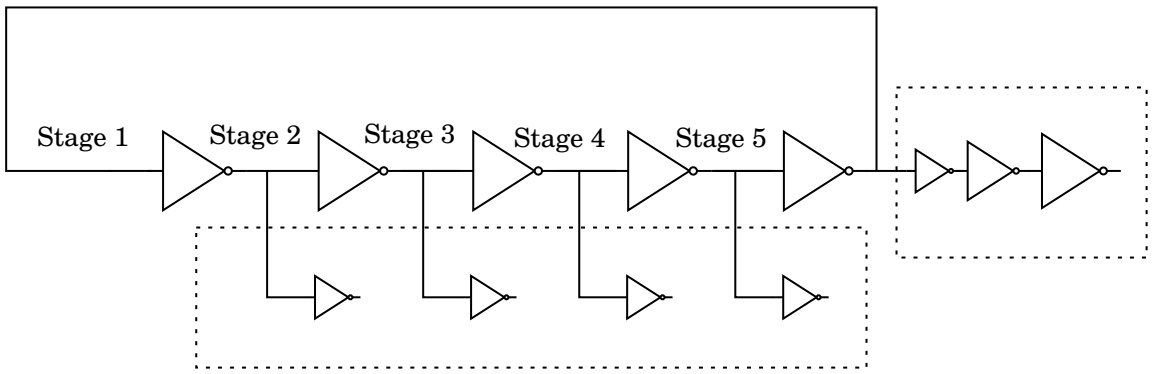


Figure 18: Top-level schematic of conventional 5-stage VCO. Circuits within dashed rectangles are buffers. Also see figure 19 for a more detailed schematic

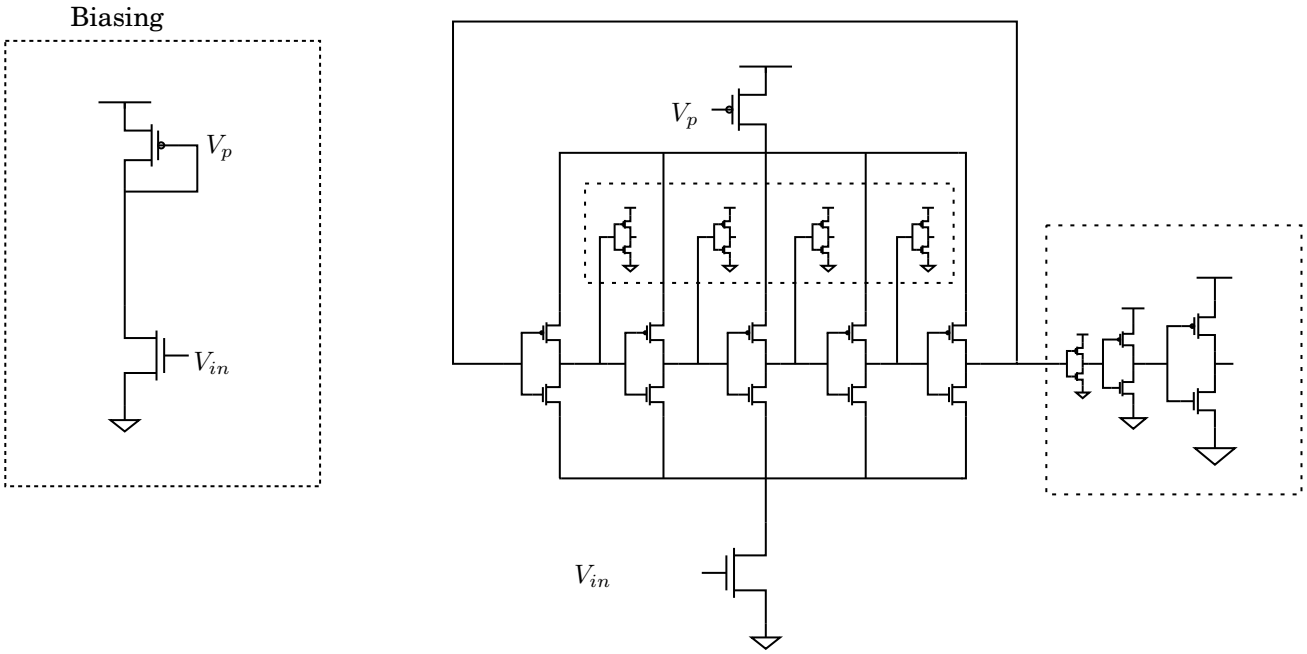


Figure 19: Detailed schematic of conventional 5-stage VCO. See figure 18 for a top-level schematic view

4.3 Conventional VCO

The VCO that will be used as a comparison for the FF VCO is the conventional VCO. This VCO is a 5-stage ring oscillator, with inverter-based delay cells. The design of the conventional VCO can be seen in figure 18 and figure 19. It should be mentioned that the inverters inside the dashed rectangles (both of them) are buffers and are not a part of the VCO itself. For now, know that the three buffers on stage 1 buffers the output to the counter, while the others are capacitive loads that makes sure each stage sees the same capacitance.

4.3.1 Sizing

All NMOS transistors are unit-sized, which in this thesis means a width $W = 0.22\mu$m and length $L = 0.18\mu$m. The exception to this is the tail transistor, which has $W = 0.88\mu$m and $L = 1.08\mu$m. All PMOS transistors have the same length as their NMOS counterpart, but have double the width since they have half the output conductance of the NMOS. A table of their sizes can be seen in table 4.

The unit-size of the NMOS is the smallest W and L that an NMOS can

Name	Ratio (W/L)
PMOS	440 nm/180 nm
NMOS	220 nm/180 nm
NMOS tail transistor	0.88 $\mu m/1.08\mu m$
PMOS tail transistor	1.76 $\mu m/1.08\mu m$

Table 3: Transistor sizes used of all PMOS and NMOS in the 5-staged VCO, and the NMOS and PMOS current-limiting transistors in the Conventional VCO

have in TSMC 180nm process. All the transistors (except for the the tail transistors) are made as small as possible in order to have a high transition frequency f_t, which increases the VCO oscillation frequency. It also reduces the total size of the VCO itself, which is a benefit for reduced area. The PMOS also has the same reasoning behind their sizing, except double the width for the reasons mentioned earlier.

4.3.2 Characteristics

A property of the conventional VCO, which can be thought of as an advantage, is that no start-up circuit is needed. This is because the conventional VCO is inherently unstable, i.e. it naturally fulfills Barkhausen's criterion. When the conventional VCO is turned on, it will begin to oscillate when enough time has passed.

VDD used on the conventional VCO is 1.8V. Smaller VDD has been used and simulated for, but for the sake of consistency and optimal performance, the VDD used here shall be 1.8V. The reason for this being that the transistors are all nominal 1.8V transistors.

The number of stages chosen are based on Park, Han and Choi's paper [13], where they use a 5-stage ring oscillator. As the conventional VCO is used as a comparison with the FF VCO, it's either 3 stages or 5 stages. Although 3 stages has higher frequency and fewer components, 5 stages can be driven at lower VDD and also has less noise issues compared to the 3 stage VCO (the number of stages itself doesn't matter, but the minimum phase noise is inversely proportional to the power dissipation and grows quadraticaly with oscillation frequency [4]). For those reasons, the number of stages

chosen on the conventional VCO is 5.

4.3.3 Performance

A good start to characterize the conventional VCO is with its performance. As there is a primary focus on how the conventional VCO will operate inside an ADC in this thesis, it's a good idea to take a look at the relations between the input voltage and output frequency. Figure 20 shows the input voltage vs. output frequency of the conventional VCO, before Parasitic Extraction (pre-PEX) and after Parasitic Extraction (Post-PEX). The frequencies for each input voltage are found by running a transient analysis with a time of 1μs for each voltage in the range of 0 to VDD (1.8V) and meausring the average frequency. Total points in each input voltage vs. output frequency plot are 200, each of the points linearly spaced from each other.
One can see how the parasitic capacitance affect the frequency of the VCO. Notably, the slope of the VCO has decreased, that the minimum voltage has increased, and that the maximum voltage has decreased. The linear region may have changed by some mV, but it's more or less the same. One can choose to look into the small changes, but there isn't really much need to as there are other factors that causes more variation and non-ideal effects than the parasitics from PEX (see chapter 5).

The minimum input voltage is approximately 0.55V pre-PEX and 0.6V post-PEX. The voltage range spans from the minimum input voltage (0.55V for pre-PEX and 0.6V for post-PEX) to approximately 1.1V, which is a voltage range of 0.4V. The minimum output frequency is around 11MHz in both pre-PEX and post-PEX. The maximum output frequency in the linear range is around 200MHz pre-PEX and 112MHz post-PEX.

However, the plot in figure 20 doesn't paint the entire picture of the conventional VCO performance. For example, it doesn't tell anything about the duty cycle. Figure 21 shows the buffered output signal of the minimum output frequency of the conventional VCO. One can see that the pulse widths are very small and that the time the VCO stays low is much longer than when it's high. The reason for this is that the conventional VCO struggles to go all the way up to 1.8V during its oscillation. This can be seen in figure 22, where the conventional VCO only oscillates between 0V and 0.95V. In counters that use both the rising edge and falling edge, this duty cycle distortion will cause errors in the counter. This may lead to large INL and DNL errors, which can reduce the linear range of the conventional VCO.

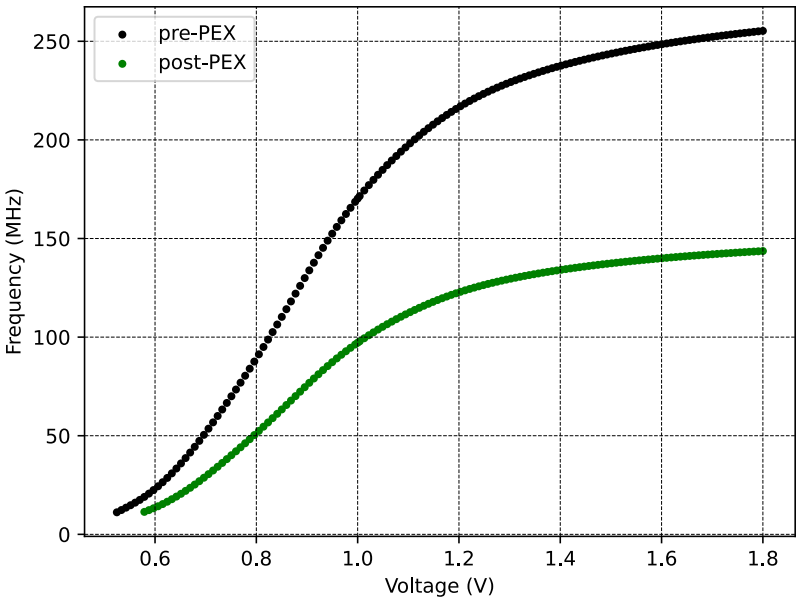


Figure 20: Schematic simulations of Conventional 5-stage VCO before PEX (Pre-PEX) and post-PEX. Plot shows Input voltage vs. Output frequency

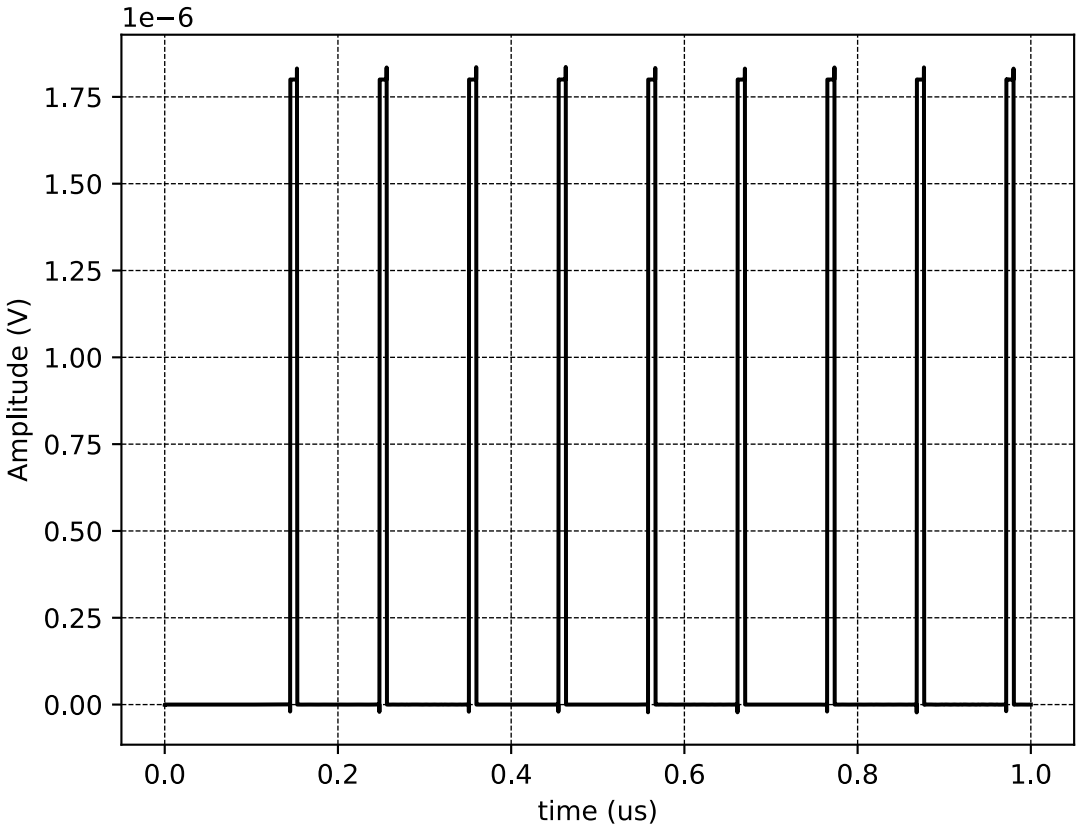


Figure 21: Plot of the minimum output frequency of the conventional VCO pre-PEX

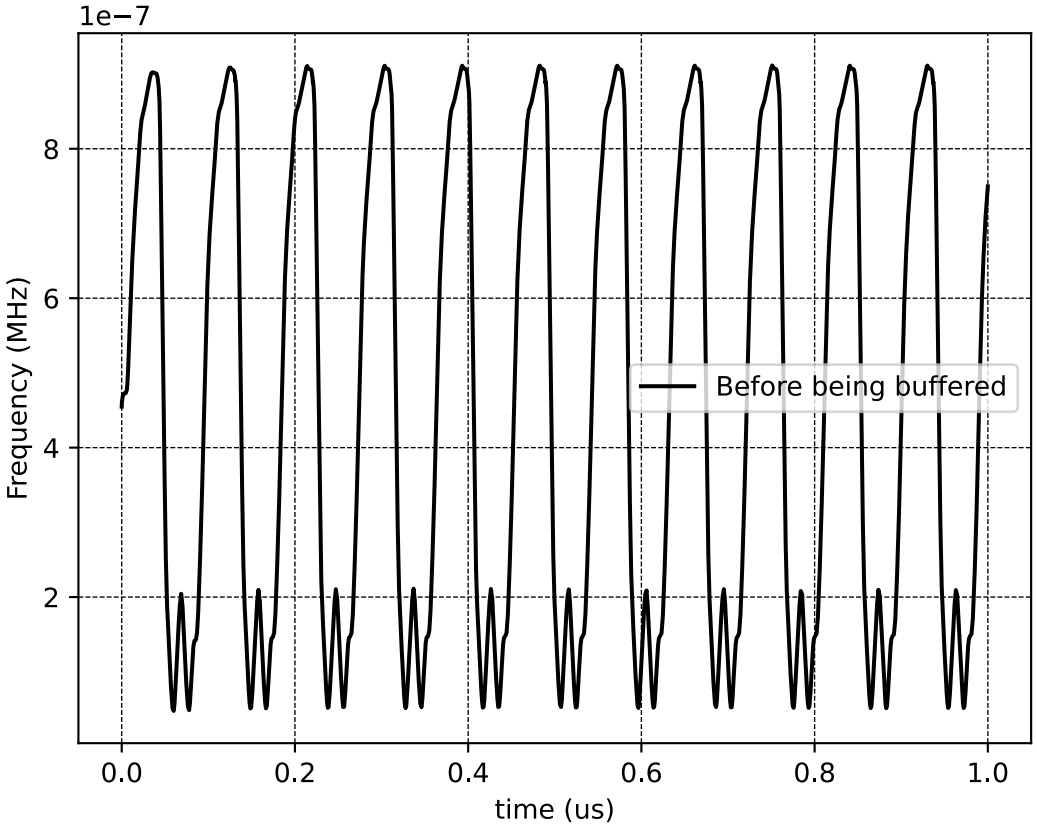


Figure 22: Plot of the minimum output frequency of the conventional VCO pre-PEX before being buffered

Getting rid of duty cycle distortion shouldn't be hard: One can send the output of the buffered signal to through a JKFF on toggle mode, which effectively removes the duty cycle distortion at the cost of halving the frequency. Removing the duty cycle distortion makes will cause less INL and DNL errors, but it will increase the quantization noise as the VCO frequency is halved (more spesifically, the SQNR will decrease with 6 dB, see equation 7).

However, removing the duty cycle distortion may not be necessary. If one implements a counter that only cares about the falling edge, then duty cycle distortion doesn't matter. Going back to PFM (figure 11), this is exactly the case. The output $d(t)$ only cares whethever $w(t)$ has an edge, either rising or falling. In this thesis, a counter where the falling edges are counted has

Name	Ratio (W/L)
PMOS	440 nm/180 nm
NMOS	220 nm/180 nm
NMOS tail transistor	0.88 $\mu m/1.08\mu m$
PMOS tail transistor	1.76 $\mu m/1.08\mu m$

Table 4: Transistor sizes used of all PMOS and NMOS in the 5-staged VCO, and the NMOS and PMOS current-limiting transistors in the Conventional VCO

	Min frequency (MHz)	Max frequency (MHz)	linear region
Pre-PEX	~ 11	~ 255.3	$\sim 0.55V - \sim 1.1V$
Post-PEX	~ 11	~ 112	$\sim 0.6V - \sim 1.1V$

Table 5: Performance characteristics of conventional 5-stage VCO with 1.8 VDD

been implemented. Regardless, it doesn't really matter whether there is duty cycle distortion in the VCO or not.

What it does imply, is that the minimum input voltage may not be the minimum input voltage after PEX. Since parasitic resistance and capacitance increases the voltage requirements of the VCO, the input voltage may not be sufficient after PEX or tapeout.

To summarize, a table containing the dimensions are listed in table 4, and performance parameters in 5.

4.4 Feedforward VCO

The main focus on this thesis is the feedforward VCO (FF VCO), which is an unique VCO, as it consists of four stages. In a conventional VCO, having four stages wouldn't be possible since it would violate Barkhausen's criterion (Four stages would result in $\angle H(J\omega_0) = 0$). However, it's a bit different for the FF VCO, and the reason for this because the FF VCO has an additional four inverters that are connected to cause oscillations (see figure 23 and 24). These inverters have a slightly bigger W/L ratio, which gives them a higher output conductance and less propagation delay. What these inverters are used for, is to take an output from one stage and feed it forward two stages. So one such inverter with a signal from stage 1 as input will

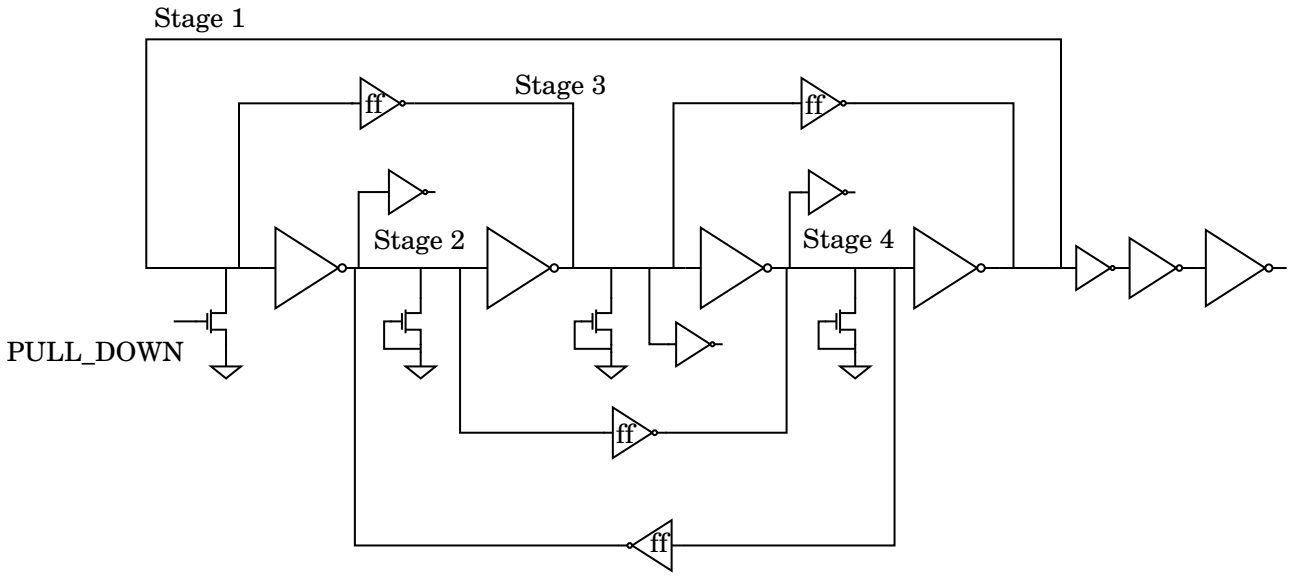


Figure 23: Top level view of FeedForward VCO schematic. Inverters inside the dashed boxes are used as buffers. See figure 24 for a detail view

feed it forward to stage 3. For this reason, these inverters are aptly named feedforward inverters, and the signal paths they form in the FF VCO are called feedforward paths (FF paths). So when a signal is produced in one stage (say stage 1), it travels through the FF path and the "direct path" (DP path) via the "direct path inverters" (DP inverters).

The main idea behind the FF VCO is that it's more linear than it's traditional counterpart and has higher oscillation frequency (in theory). Of course, this comes at the expense of more power consumption, as the FF VCO has more inverters than the conventional VCO.

4.4.1 Working Principle

Since the FF VCO has a different architecture than a conventional VCO, it might be hard to understand how the FF VCO works: how does feeding a logic 0 two stages forward cause oscillations? The answer for that consists of two factors: timing (propagation delay) and strength (output conductance). The timing factor is relatively simple to understand: due to the difference in propagation delay, the FF inverters and DP inverters will pull the voltage on one stage at different times. This timing difference alone isn't going to do anything, but things change drastically when the inverter strengths are taken into consideration. It's already mentioned that the FF inverter has more output conductance than the DP inverter, which means

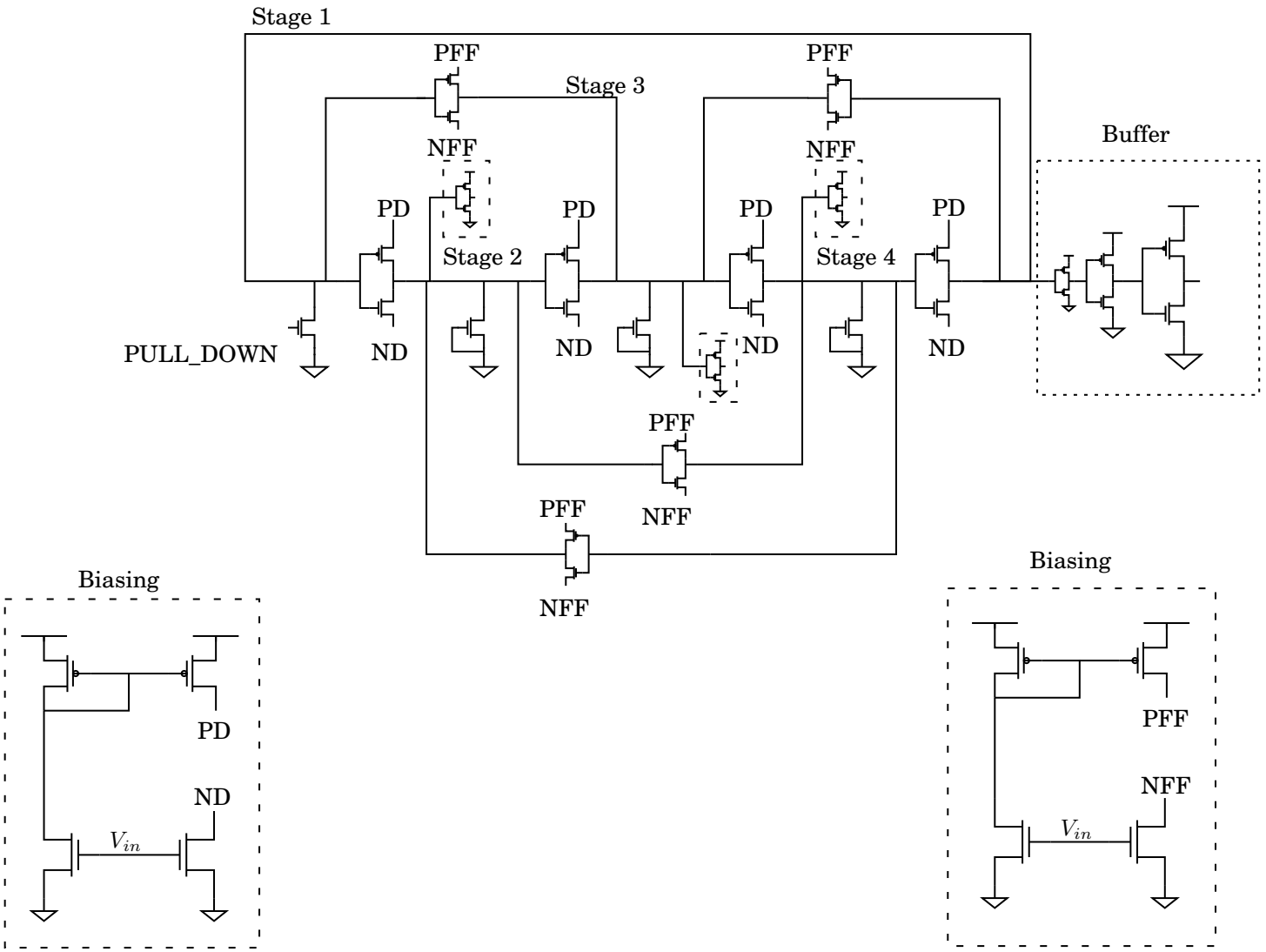


Figure 24: Detailed view of FeedForward VCO schematic. Inverters inside the dashed boxes are used as buffers, except the boxes labeled "Biasing". See figure 23 for top level view

that the FF inverter is the dominating inverter of the stage. This means that in the case where the FF inverter and DP inverter pulls in the opposite direction, the FF inverter will be the inverter that decides the state of the stage. However, if the FF inverter is *barely* stronger than the DP inverter, then the DP inverter has just enough strength to cause a ripple in the stage. The ripple will then propagate throughout the FF VCO to different stages at different times, and this will eventually cause the FF VCO to oscillate continuously.

An analogy which can be of use when trying to understand the FF VCO is the analogy of two people in a tug-of-war. They are for the most part evenly matched, but there is one who is stronger than the other. This means that there will be no stalemate, as the stronger one will be the one dominating. But due to different timings on when they pull, the weaker person will also be able to pull the rope away when the stronger one isn't pulling. And thus it will go back and forth.

4.4.2 Sizing

The sizing of the FF VCO isn't much different from the conventional VCO, as the same methodology applies: make the transistors as small as possible to have the highest oscillation frequency and consume the smallest area possible. Where things get different is the sizing of the transistors in the FF inverters. The FF inverters need to have more output conductance than the DP-inverters, i.e. a higher width/length (W/L) ratio (but not too strong!). There are two ways to do this, which is to either let the transistors in the FF inverters have bigger W, or to let the transistors in DP inverters have longer L. In this thesis, a longer L for the transistors in the DP inverters were chosen, as this would result in less current and therefore less power consumption. Of course, increasing L does mean a decrease in output conductance and therefore a decrease in transition frequency f_t, and it becomes even lower as there will be more parasitic capacitance added. This may make one think if increasing W would be a better answer, and while there would be an increase in performance, it wouldn't necessarily be worth it. Increasing W would increase the speed of the VCO at the cost of increasing the power consumption, but the increase W also increases the parasitic capacitance (the parasitic capacitance is proportional to the area of the transistor). This means that there is little performance improvement to gain from bigger W, as the increase in f_t also comes with capacitance that limits the f_t improvement. This doesn't seem like a good trade-off,

Name	Ratio (W/L)
Direct Path PMOS	440 nm/360 nm
Direct Path NMOS	220 nm/360 nm
FF Path PMOS	440 nm/180 nm
FF Path NMOS	220 nm/180 nm
Direct path PMOS current-limiter	1.76 $\mu m/1.08\mu m$
Direct path NMOS current-limiter	0.88 $\mu m/1.08\mu m$
FF path PMOS current-limiter	1.76 $\mu m/0.9\mu m$
FF path NMOS current-limiter	0.88 $\mu m/0.9\mu m$

Table 6: Transistor sizes used in all PMOS and NMOS in the 5-staged VCO and FF VCO, and the NMOS and PMOS current-limiting transistors in the Conventional VCO and FF VCO

which is the reason why the decision of increasing the length was chosen: the trade-off with increased W seems worse than the trade-off for increased L.

The sizes of the transistors can be seen in figure 6. As with the conventional VCO, the PMOS has twice the width as the NMOS to compensate for the lack of output conductance.

4.4.3 Characteristics

Unlike the conventional VCO which don't require a start-up circuit, the FF VCO should have some sort of start-up mechanism to start the oscillations. This is to avoid being conditionally stable, which can happen due to the FF VCO having an even number of stages. In order to make sure the FF VCO does start up, a switch that connects stage 1 to GND has been implemented. The swicth is realised with a NMOS switch to GND, which is chosen due to its small size. But implementing a NMOS on stage 1 results in parasitic capacitance on the output stage, which causes inbalance in the different stages of the FF VCO. To compensate, NMOS switches to GND that are permanently off are implemented on all stages. However, this will introduce additional parasitic capacitance on all the stages, which will reduce

FF VCO frequency. One can choose to not implement the start-up circuit, and instead start up the FF VCO by switching the VDD from 0V to 1.8V (i.e. turning on the FF VCO). However, this method is a poor design decision, as there is technically nothing that says the FF VCO will start to oscillate when modulating the VDD. Therefore, it's absolutely recommended to have a start-up circuit despite decreasing the frequency and not being mandatory.

The number of stages chosen for the FF VCO is 4, which is chosen due to higher frequency and smaller size. It's also possible to make a 3-stage FF VCO, which goes even faster and theoretically uses less space. In practice. they require the same space anyways, since the same die area is required to realise both circuits. Also, 3-stage FF VCO suffers from noise gradients, which 4-stage FF VCO doesn't. Having an even number of stages cancels out any linear noise gradients when symmetrical layout is implemented.

VDD used on the FF VCO is 1.8V. Smaller VDD has been used and simulated for, and at one point the FF VCO was intended to have a VDD of 1V. However, a VDD of 1.8V was used in the end to avoid any problematic issues with the NMOS and PMOS. As TSMC describes their models as "1.8V nominal_VT transistors", it is safe to assume they're intended to be implemented in a circuit where VDD is 1.8V.

4.4.4 Performance

Similarly to the conventional VCO, the FF VCO performance will first be measured by relations between the input voltage and output frequency. This relation can be seen in figure 25, both pre-PEX and post-PEX. Also similarly to the conventional VCO, the frequencies are found by running a transient analysis in 1μs for each voltage in the range of 0 to VDD (1.8V) and meausring the average frequency. Total points in each plot are also 200, and linearly spaced from each other.

As with the conventional VCO, the slope and the maximum frequency has decreased after PEX, which is no surprise. What is surprising is that the minimum voltage has decreased, meaning that the FF VCO can oscillate at a lower input voltage. Nevertheless, the linear voltage range is still pretty much the same despite the changes in input voltage. Small variations aren't worth to look deep into, as other factors will affect the FF VCO more (see chapter 5).

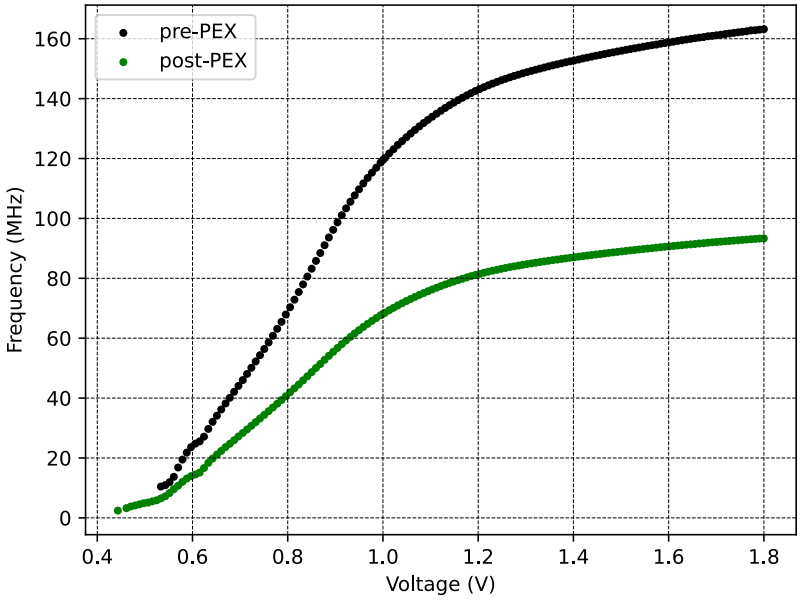


Figure 25: Schematic simulations of Feedforward 5-stage VCO before PEX (Pre-PEX) and post-PEX. Plot shows Input voltage vs. Output frequency

An interesting discovery made was that many of the outputs of lower input voltages measured in the FF VCO seemed to oscillate, but used too much time to settle. If the transient analysis was longer than 1μs, then maybe oscillations could've been detected. Nevertheless, they're most likely outside the linear range, which makes those frequencies irrelevant. However, it's an interesting thought and brings forth the question of how low the FF VCO actually can go.
Unlike the conventional VCO, the FF VCO has no problems oscillating at lower frequencies. In figure 26, one can see the smallest frequency almost oscillates all the way to VDD. Notably, figure 26 shows the lowest frequency from post-PEX, which rarely performs better than its pre-PEX counterpart. In other words, even after PEX, the FF VCO still manages to oscillate all the way to VDD.

The minimum input voltage is approximately 0.498V pre-PEX and 0.44V post-PEX. The voltage range spans from the minimum input voltage (0.498V for pre-PEX and 0.47V for post-PEX) to approximately 1.1V, which is a voltage range of $0.4 \sim 0.5$V. The minimum output frequency is around 6.6MHz

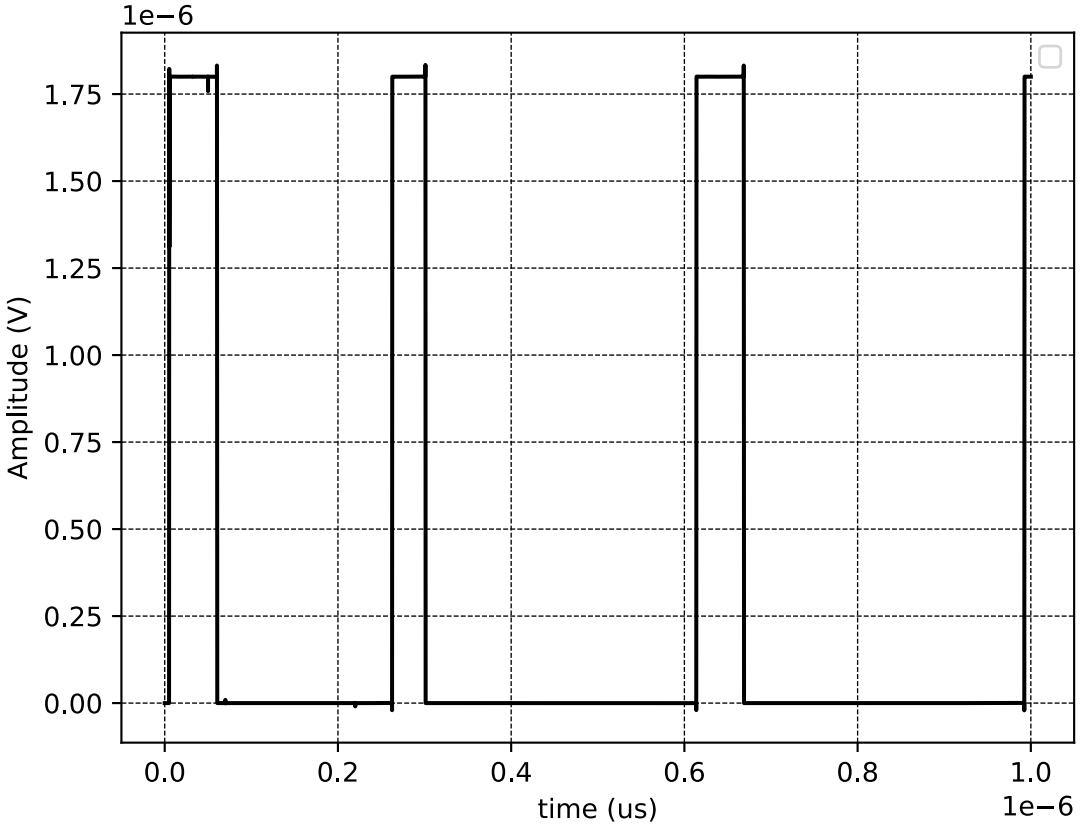


Figure 26: Plot of the minimum output frequency of the FF VCO post-PEX

	Min frequency (MHz)	Max frequency (MHz)	linear region
Pre-PEX	~ 6.6	~ 134	$\sim 0.498V - \sim 1.1V$
Post-PEX	~ 3.8	~ 76.3	$\sim 0.47V - \sim 1.1V$

Table 7: Performance characteristics of conventional 5-stage VCO with 1.8 VDD

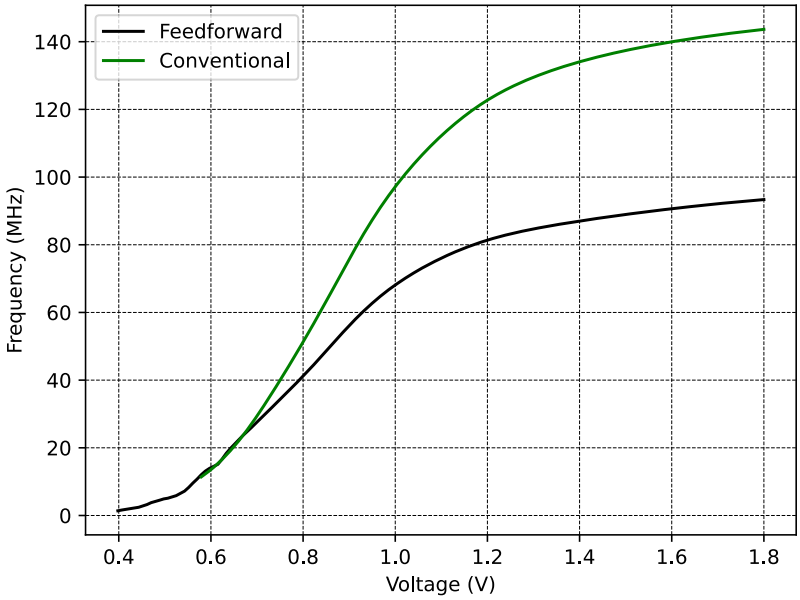


Figure 27: Post PEX simulations of Conventional 5-Stage VCO and FF VCO

in both pre-PEX and 3.8MHz post-PEX. The maximum output frequency in the linear range is around 134MHz pre-PEX and 76.3MHz post-PEX.

A table containing the dimensions are listed in table 6, and performance parameters in 7.

4.5 Comparison

Figure 27 shows both the conventional VCO and FF VCO in the same plot. Discussions regarding the difference of the two will be taken in chapter 6

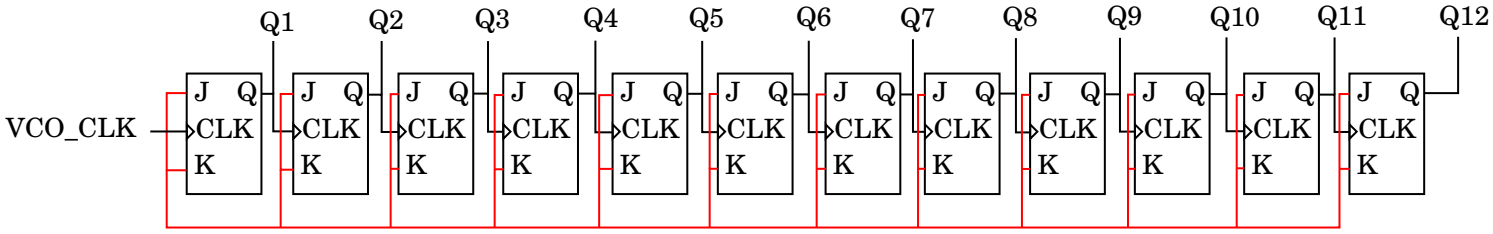


Figure 28: Schematic of ripple counter

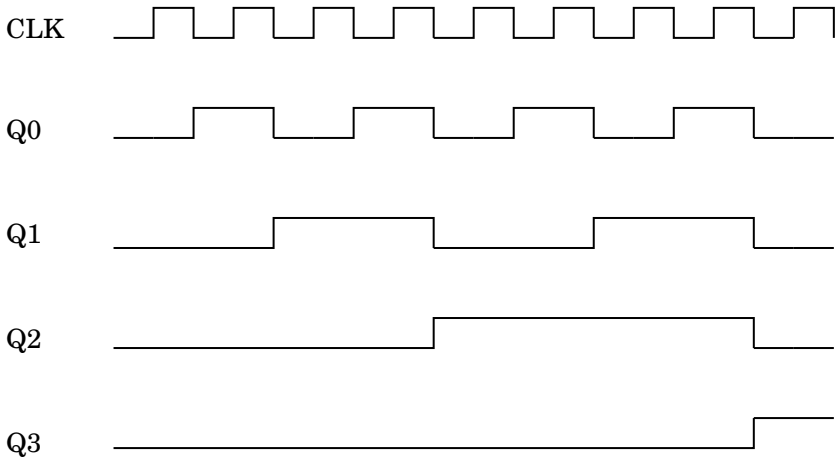


Figure 29: Conceptual timing diagram for how the ripple counter operates

4.6 Counter and register (including a MUX)

4.6.1 Architecture

The counter made in this thesis is a 12-bit ripple counter, which can be seen in figure 28. Twelve JK flip-flops are used to implement the ripple counter. The ripple counter itself has an asynchronous reset to reset the counter values. NMOS transistors are kept to minimum size while the PMOS has twice the width to match the NMOS output conductance. The JKFF are set to "toggle", meaning that the first JKFF alternate between logic 1 and logic 0 for each falling edge from the VCO, the second JKFF alternates between 1 and 0 for each falling edge from the first JKFF, and so on. This effectively means that if the first JKFF ripples at half the speed of the VCO, the second JKFF ripples at one quarter of the speed of the VCO, and so on (see figure 29).

The register used in the VCO ADC consists of 12 D flip-flops (DFF): one for each bit from the counter. The outputs of the DFF are then buffered and sent to one of three MUXes. In this thesis, the 1st, 4th, 7th and 10th bit are

Digital input	Bits on MUX output
00	1st, 2nd and 3rd
01	4th, 5th and 6th
10	7th, 8th and 9th
11	10th, 11th and 12th

Table 8: I/O logic table between MUX input and output

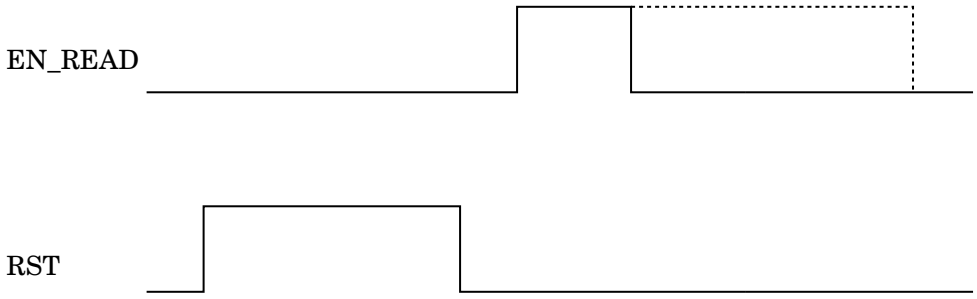


Figure 30: Conceptual timing diagram of the control logic for the ripple counter

connected to the 1st MUX, the 2nd, 5th, 8th and 11th bit is are connected to the 2nd MUX, and the 3rd, 6th, 9th and 12th bit are connected to the 3rd MUX . All the MUXes are controlled with a 2-bit input that controls which bit is sent to the output (see table 8).

The MUXes are exclusive to this thesis, as there were a limited amount of analog and digital pads to connect to. Since a 12-bit counter theoretically requires 1 pin for each bit, 12 pins are required in order to measure each bit. As the chip only has 16 pins available to connect the entire VCO ADC, having 12 bits use 12 of 16 pins is unacceptable. Therefore, a MUX is implemented in order to reduce the number of pins required from 12 to 5: two digital input pins and three digital output pins. Instead of looking at all 12 bits simultaneously, the MUX displays 3 bits at a given time, which can be changed with the two-bit digital input. See figure 8 for MUX I/O logic table.

4.6.2 Working Principle

The way to start and stop the counting of the ripple counter is by having two control signals: one to enable the counting (EN_READ) and one to reset the counter (RST). The control signals are provided with VHDL code that's implemented on the PYNQ Z1 FPGA (see figure 37). The VHDL code can be seen in the Appendix (chapter 8). Figure 30 depicts the timing of the two control logic signals. One can notice that EN_READ has a dotted line

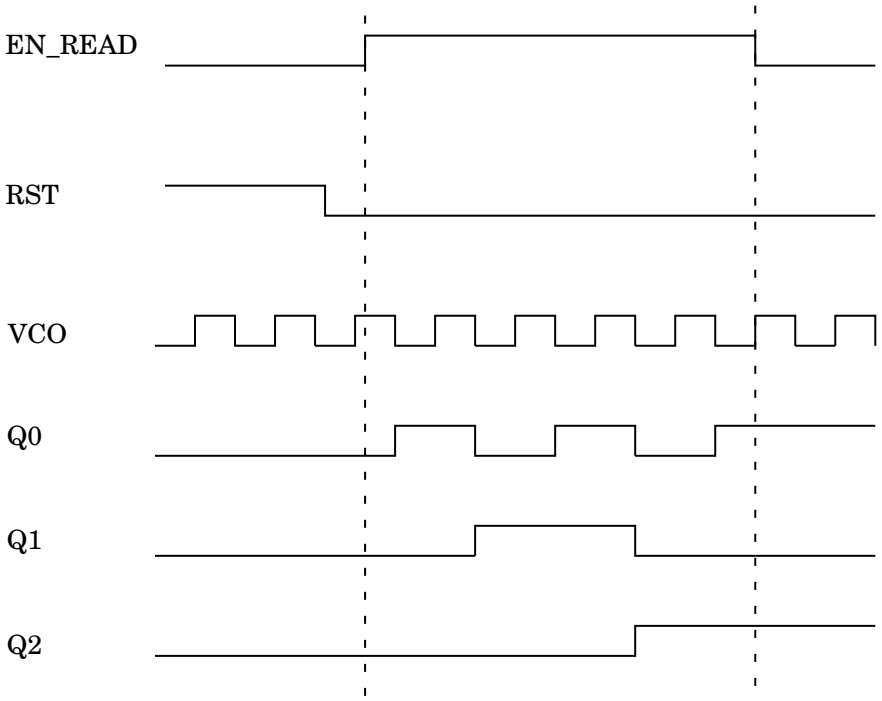


Figure 31: Timing diagram of how the VCO works together with ripple counter and control logic

on its timing diagram, reason being that its length depends on the desired sampling frequency used in the system. An EN_READ signal with a period of 1μs means that information is gathered from the VCO for 1μs, which is the same as sampling the VCO for 1μs. In other words, an EN_READ signal with a period of 1μs is the same as having a sampling frequency of $\frac{1}{1\mu s} = 1$MHz. Thus, EN_READ is where the sampling clock is connected to, and its frequency contributes to the resolution to the whole VCO. How the EN_READ and RST controls the counter can be seen in figure 31.
The sampling frequency used in the counter is no different from the one explained in the Nyquist-Shannon theorem, which is written as

$$f_{Nyqu} \leq 2f_{in,max}$$

where f_{Nyqu} is the sampling frequency and $f_{in,max}$ the highest frequency of the sampled signal. So a 1MHz sampling clock will only correctly sample frequencies strictly lower than $0.5M$Hz.

After every sampling period t_s, the counter sends it's counted value to the register. The register stores the most recent digital count inside it, and is

refreshed after every t_s. This enables the counter to keep counting while the output is unchanged, allowing the register output (which is also the ADC output) to be digitally processed in parallel with the counting. The register output is often buffered to increase the output conductance.

4.6.3 Performance

As the counter is a digital circuit, the key performance factor is speed. More precisely, how high frequency can the input of the counter handle before the counter becomes too slow to count. From simulation tests ran with Spectre, the counter is able to handle an input frequency of at least 1.5GHz. Since the maximum output frequency of both the conventional VCO and the FF VCO doesn't go higher than 200MHz, it's safe to say that the counter isn't the bottleneck of the system.

When it comes to the registers and MUXes, minimal sizes on all NMOS is sufficient (PMOS has same L, double the W). As long they can hold and send the digital numbers and don't bottleneck the ADC, it shouldn't be a problem.

4.7 Layout

4.7.1 Conventional VCO

In the layout of the conventional VCO, there has been an attemt at symmetrical design to have all stages see the same capacitance and be balanced. The transmission gates are placed opposite of each other in attemt of common centroid layout. It should in theory cancel out linear gradients to a small extent, but this effect may be minimal as the the top left section of the layout have PMOS current sources and the bottom left section have NOMS current sources (see figure 32). There has also been an attempt at using common centroid layout technique in the VCO itself, but to a lesser degree of success. As the number of stages are odd numbered, there is no feasible way do a perfectly symmetrical design. The next best thing to do is to put the inverters as close to each other as possible to avoid mismatch and long routing.

The conventional VCO itself is 10.61μm in height and 7.14μm in width. If the transmission gate and current mirror is included, then the total height is 18.76μm and total width is 20.81μm.

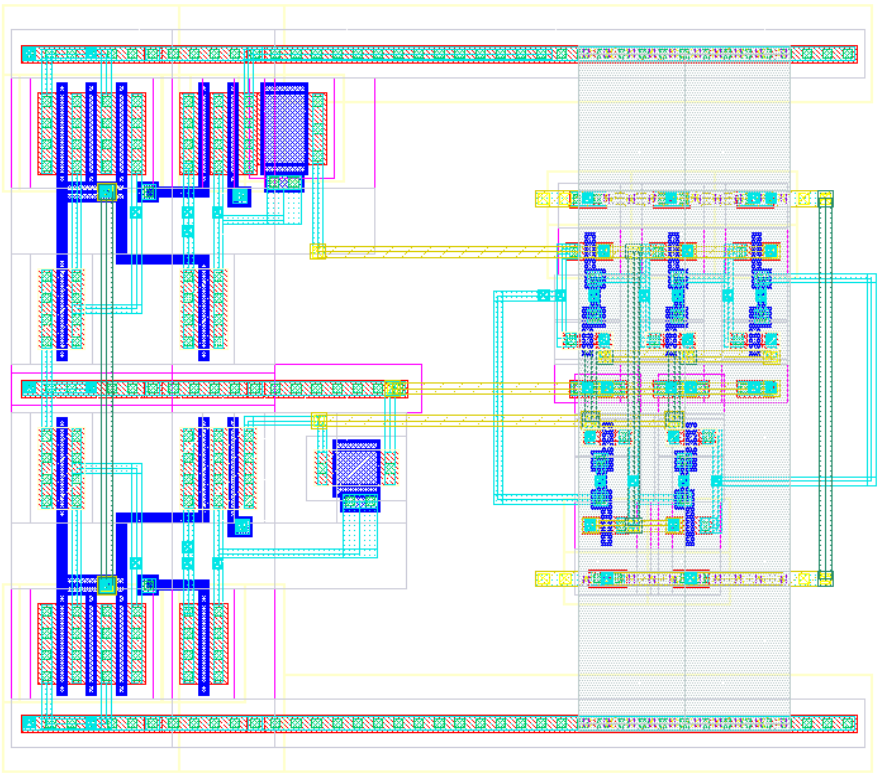


Figure 32: Layout of Conventional 5-stage VCO, including current mirror and transmission gates

4.7.2 Feedforward VCO

In the layout of the FF VCO, symmetrical design have been a major consideration, so that each stage shall see the same capacitance and thus have as little imbalance in phases as possible. The FF inverters and direct path inverters have been placed as close to each other as possible to avoid mismatch. A thing worth mentioning is that the FF inverters and direct path inverters could've shared terminals, thereby using even less space than neccesary. However, the version of Cadence used to do the layout wouldn't allow an increase in the diffusion layer of the NMOS in the inverters. The other option would be to increase the width of the NMOS or decrease the width of the PMOS in schematics. Whichever option is chosen, it comes at the cost of VCO performance. More spesifically, the fall time of the VCO will be slightly less than the rise time due to difference in drive strength. In other words, there is a tradeoff between less size and better performance. VDD lines are made long and wide to avoid parasitic resistance.

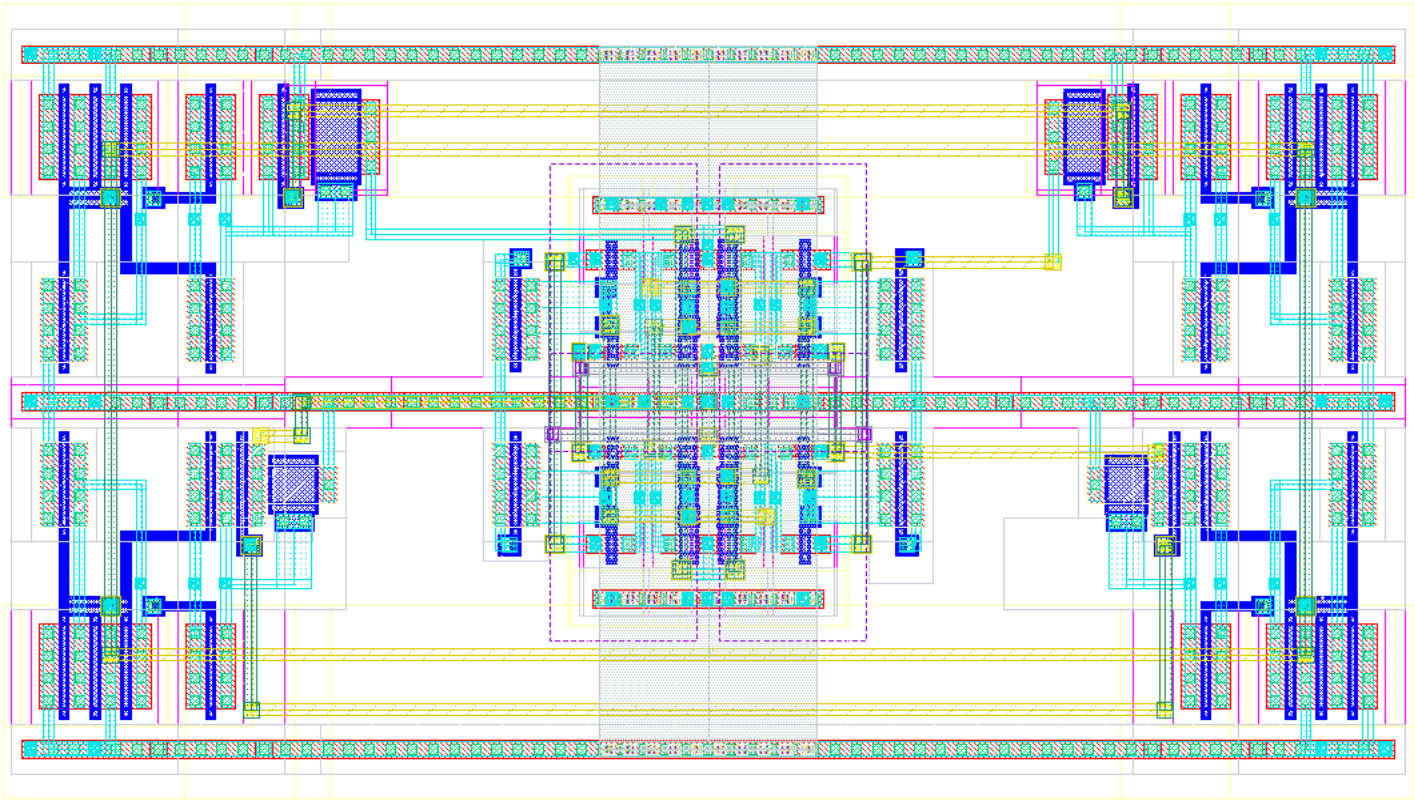
The FF VCO itself is 11μm in height and 10.42μm in width. If the transmission gate and current mirror is included, then the total height is 18.76μm and total width is 32.64μm.

4.7.3 Both VCOs

A figure showing both VCOs, implemented with transmission gates, current mirrors and buffers can be seen in figure 34.

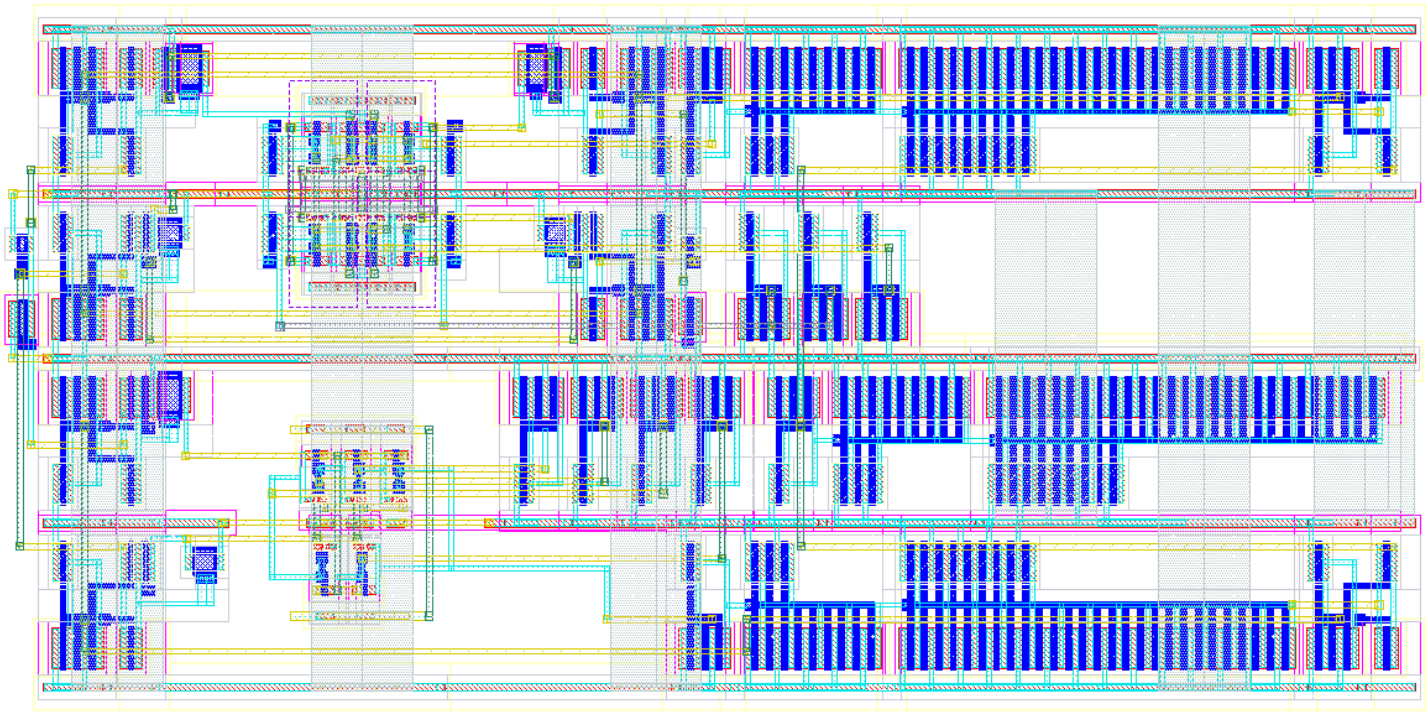
4.7.4 Counter, Register and MUX

The Counter, Register and MUX, i.e. the digital circuit, are all implemented using unit-sized NMOS and PMOS (see figure 35). For this reason, they can be stacked up in a nice and compact way. This is to save as much space as possible, and to have minimal routing paths.



/

Figure 33: Layout of FeedForward VCO, including current mirror and transmission gates



/

Figure 34: Layout of both VCOs and the transmission gates, current mirrors and buffers

Figure 35: Layout of ripple counter with DFF register and MUX

4.7.5 Total system

A figure showing the entire system is shown in figure 36. Power lines are wide to provide minimal resistance, with VDD (both analog and digital) and GND explicitly consisting of metal 5. It's also possible to see how the system is connected to the padframe.

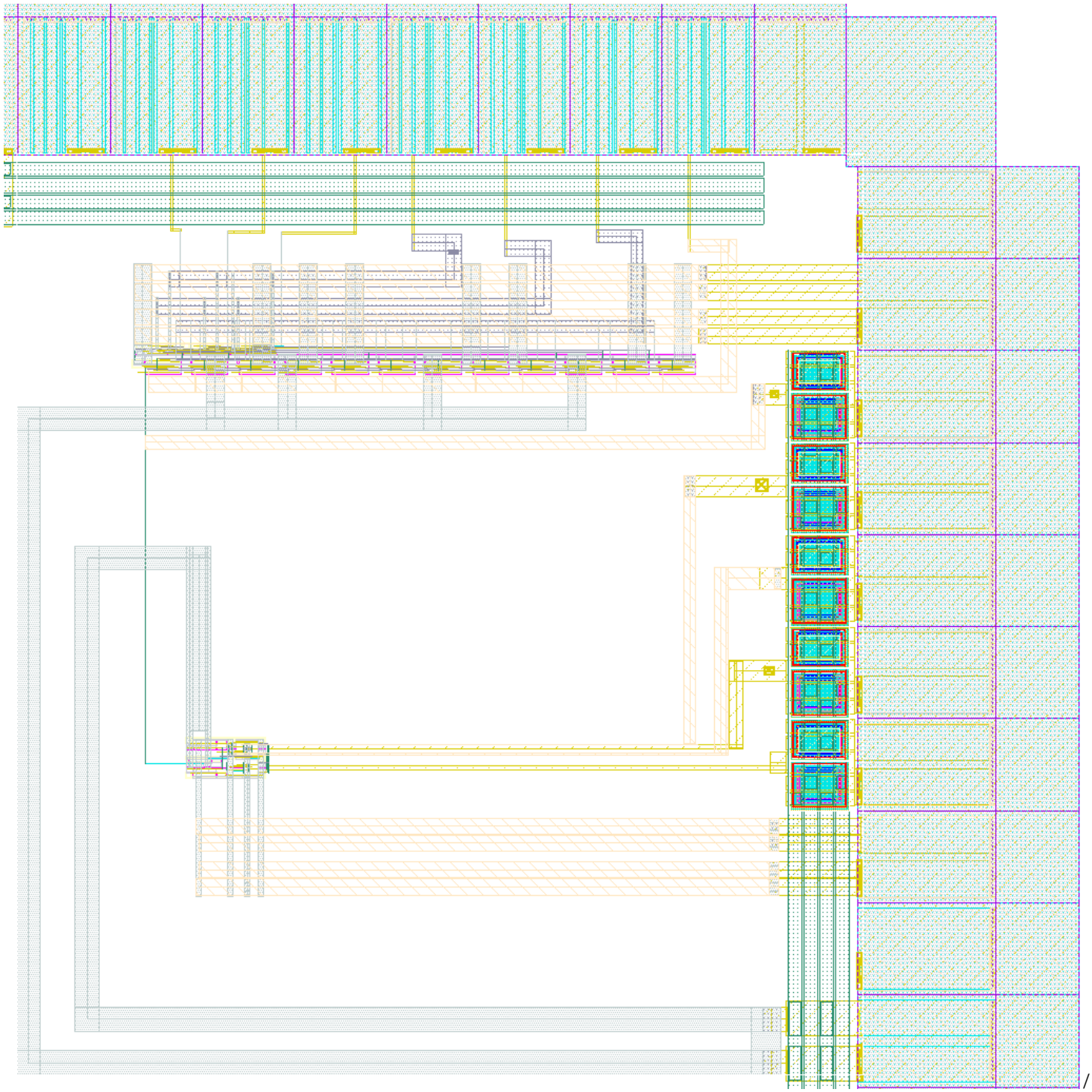


Figure 36: Layout of both the full system

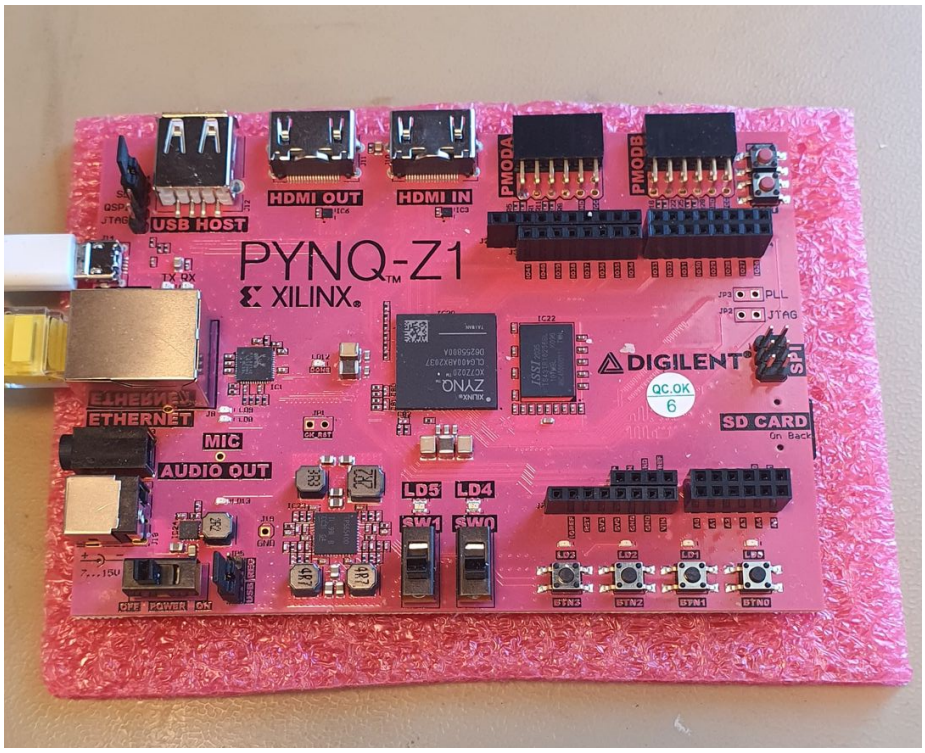


Figure 37: Picture of the PYNQ-Z1 development board

5 Measurements

5.1 Test Setup

To test the chip, a custom PCB was made that the chip is placed on and a PYNQ Z1 is used to control digital I/O and timing.

5.1.1 PYNQ Z1

The FPGA used to control digital I/O is the PYNQ Z1. It has a clock frequency of 100MHz, which implies that the minimum pulse width of a signal sent from the PYNQ Z1 is at minimum 10ns. The PMOD interface is used to send and recieve signals to the chip, with all ports of the PMOD. Level shifters are used to shift the digital voltage for readout. As the clock frequency is 10MHz, the step size between each timing is 10ns. The logic is coded in VDHL, and the synthesis and bitstream is made with Vivado ML. The PYNQ is controlled using jupyter notebook.

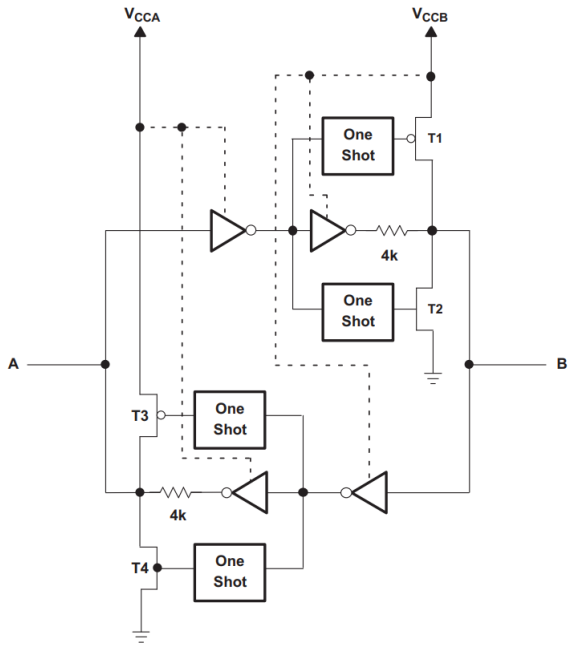


Figure 38: Architecture of TXB0104 cell [17]

5.1.2 BOB-11771

The BOB-11771 is an evaluation PCB that uses the TXB0104, which is a 4-bit bidirectional voltage-level shifter with automatic direction sensing. This 4-bit noninverting translator uses two separate configurable power-supply rails. The A port is designed to track VCCA. VCCA accepts any supply voltage from 1.2V to 3.6V. The B port is designed to track VCCB. VCCB accepts any supply voltage from 1.65V to 5.5V. This allows for universal low-voltage bidirectional translation between any of the 1.2-V, 1.5-V, 1.8-V, 2.5-V, 3.3-V, and 5-V voltage nodes. VCCA should not exceed VCCB.

The architecture of a TXB0104 cell is shown in figure 38, which consist of latches, one-shots, pull-up and pull-down tramsistors. The way a cell works is that a signal sent to A gets level-shifted and sent out to B. This happens due to the signal being sent through a double-inverter to buffer the digital signal, with the output of the first inverter connected to two one-shots. These one-shots detect rising and falling edges, which momentarily turns on the PMOS or the NMOS depending on whenever the signal has a falling edge or a rising edge. The one-shots are implemented mostly to speed up the high-to-low or low-to-high transistion.

According to the datasheet, it's important that PCB traces (and thereby any wiring) are kept to a minimum lenght. This is due to two reasons: the first reason being to avoid excessive capacitive loading and the second reasons being that the round-trip delay of any reflection must be shorter than the duration of the one-shots. Excessive capacitive loading can cause the one-shot to time-out before the signal is driven fully to the positive rail, which can make the level-shifted signal have slow high-to-low or low-to-high transitions. This can cause uncertainty in when the RST and EN_READ are considered to be high/low, which then causes an uncertainty in when the counter resets and reads. Reflections can also become a problem if its strength is large enough, since the circuit in figure 38 is technically an unclocked latch with extra features. If both sides of the TXB0104 cell has strong reflections, then the outputs will begin to oscillate instead of shifting the levels of a signal.

5.1.3 PCB

A PCB was made to place the chip and test it. As the VCO ADC will be packaged with a JLCC-84 package, an appropriate socket will be implemented on the PCB. The schematics and layout of the PCB can be seen in figure 39 and 40 respectively. The PCB has LM317 voltage regulators to regulate the analog VDD and digital VDD. Decoupling capacitors are added on the power supply lines and the analog input signals. Additional capacitance is going to add some delay to the input signals, but they aren't signals that's supposed to change dynamically. Therefore, it's okay to put some decoupling on the analog inputs.

On figure 39, One can see the analog signals and supplies are connected on the left side of the chip socket. AVDD and DVDD are the supplies for the analog circuits and digital circuits respectively. IN1 is the input voltage port for the conventional VCO and FF VCO. CTRL is the signal that controls which VCO that should appear on the output. CTRL = 1 connects the FF VCO on the output, while CTRL = 0 puts the conventional VCO on the output. IN2 is the input for the FF path of the FF VCO. PULL_DOWN is the input of the startup circuit of the FF VCO. V_OUT is the output of the buffered VCO output before entering the counter.

The digital part of the VCO ADC is beneath the JLCC-84 socket. RST is the reset signal for the counter, READ is the enable read for the counter, BIT0 and BIT1 are the MUX control signals and PIN2, PIN1 and PIN0 are

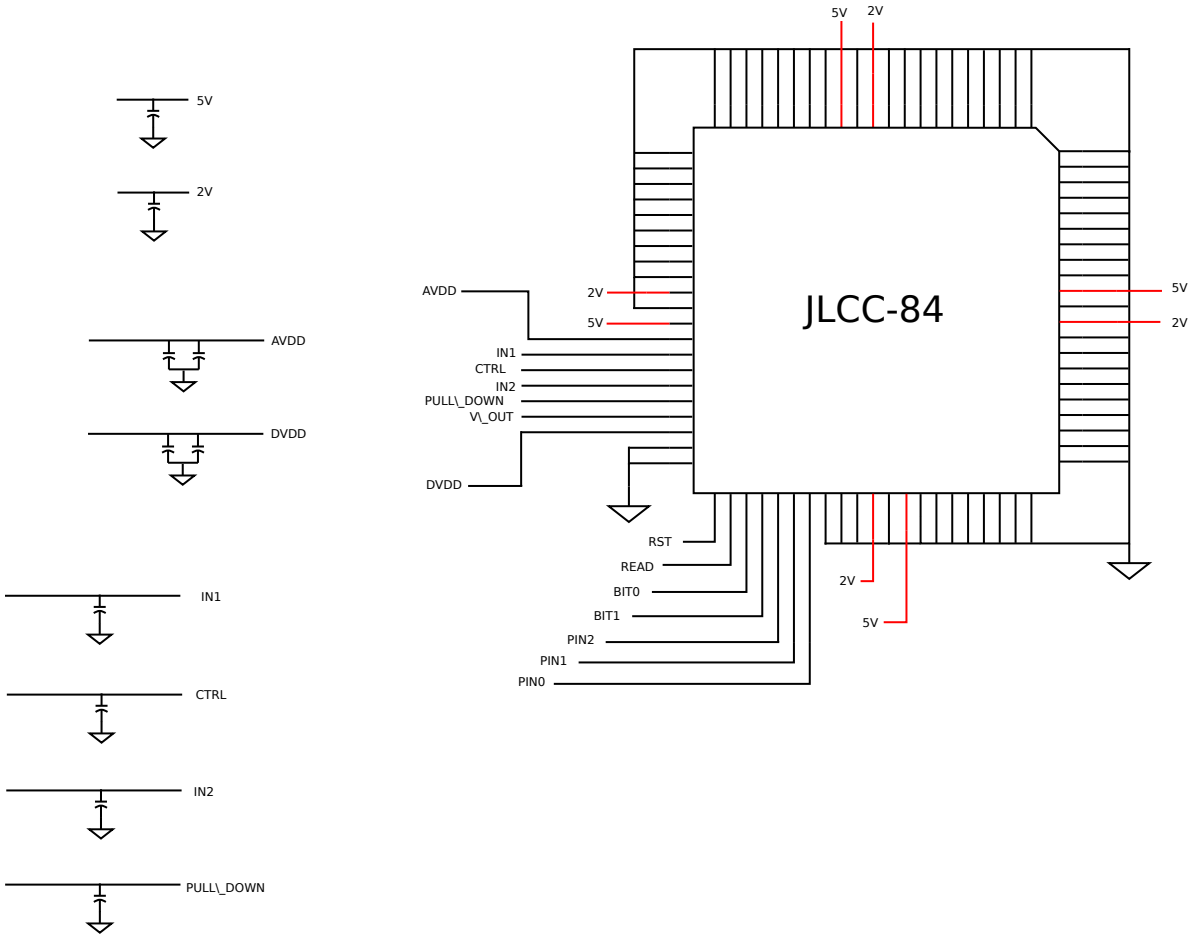


Figure 39: Architecture of the connections on the PCB used for testing

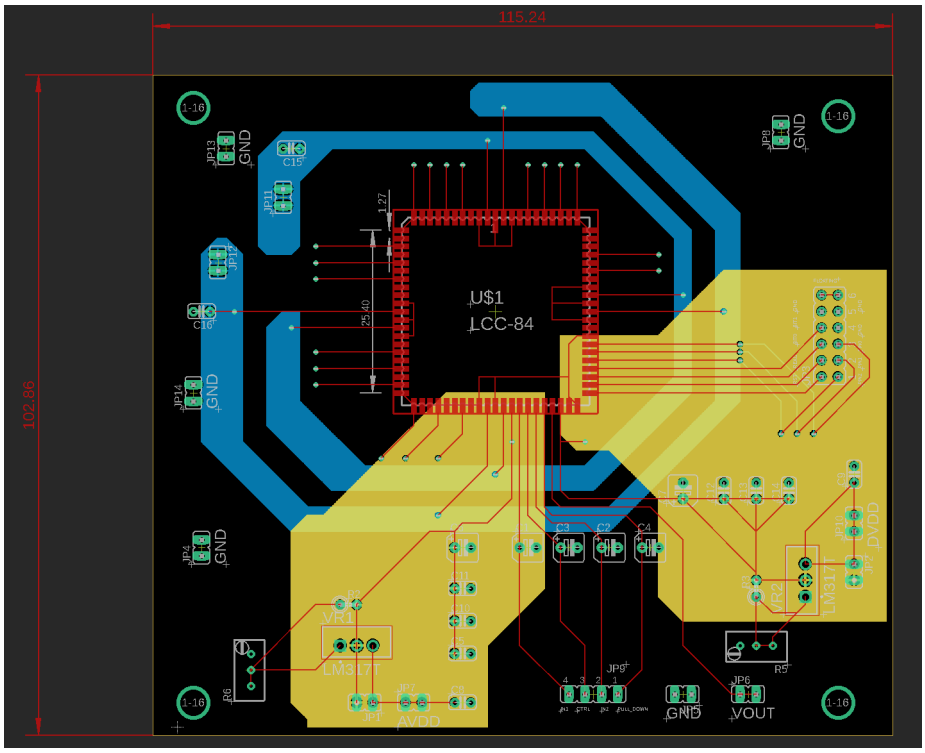


Figure 40: Layout of the testing PCB

the output pins of the three MUXes.

The red wires that are labeled 2V and 5V are the inputs for the chip's ESD protection ring.

The layout of the PCB is straight forward: Make supply lines as wide as possible, while the routing between the pins and the JLCC-84 socket pads as thin and short as possible. In the layout implemented in figure 40, the entire third metal layer is connected to GND (It's not shown in the layout, as the whole view would be orange), the second layer is dedicated to power supplies (right yellow plate is digital VDD, the other is analog VDD), and the blue rails are supplies for the ESD guard rings (innermost ring is 5V, while the outermost is 2V).

5.2 Lab Equipment

The labratory equipment used in the measurements can be seen in figure 42. Power supplies used in the measurement are the E3631A power supply from HP (Agilent), which is a stable power supply with built-in power

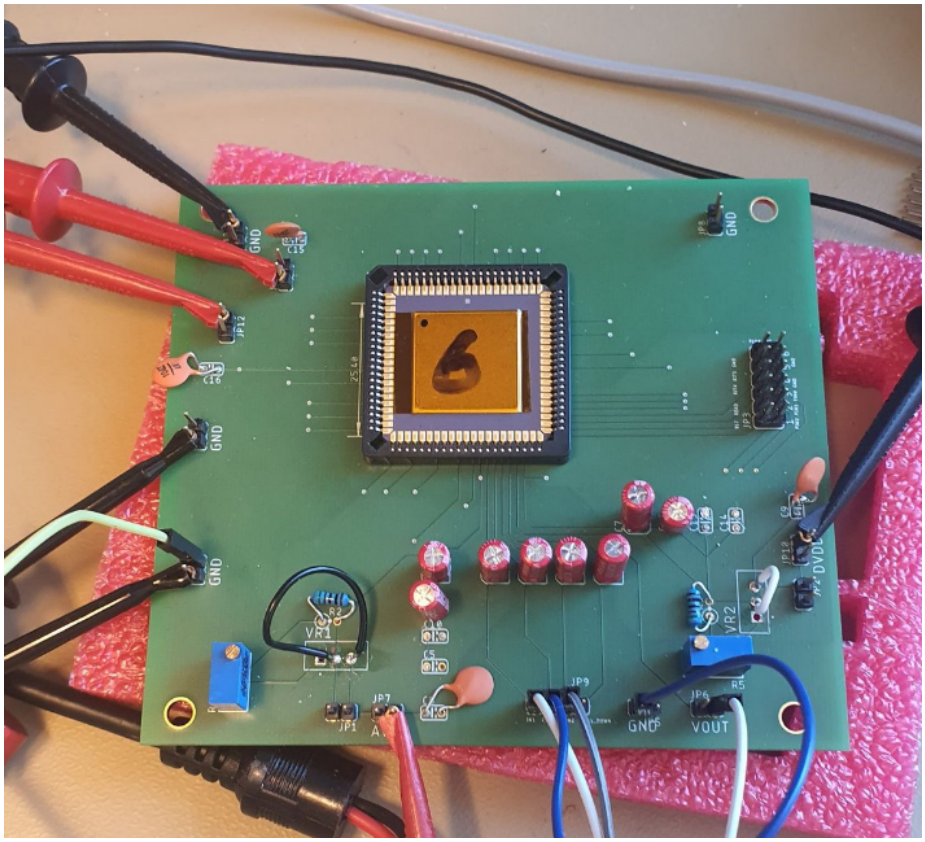


Figure 41: Picture of the testing PCB

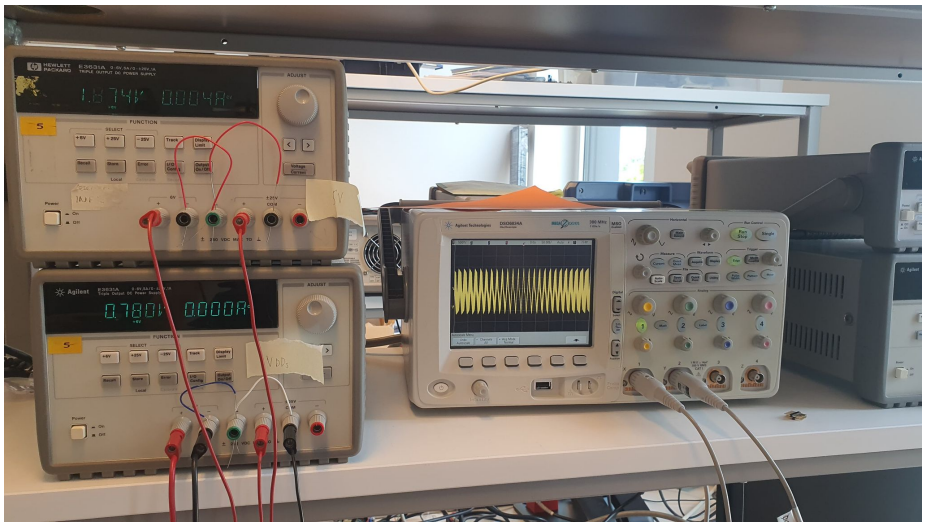


Figure 42: Picture of the labratory equipment used for measurements. Top left: HP (Agilent) E3631A power supply for ESD guard ring. Bottom left: Agilent (HP) E3631A power supply for Analog VDD and VCO input voltages. Right: Agilent DSO 6034A oscilloscope.

regulation (is able to limit voltage and current). The oscilloscope used for the measurements are the DSO6034A from HP (Agilent) and has a sampling rate of $2G$s/s. Considering that the VCOs don't go above 200MHz, it's should be safe to assume that there's no alias in the oscilloscope measurements.

5.3 Measurement Results

The measurements was done using the PCB pictured in figure 41, with the chip inserted into the socket. The inside of the chip can be seen in figure 43, with the VCO ADC placed in the uppermost right corner.

5.3.1 Conventional VCO

The measurements of the conventional VCO can be seen in figure 44, where the same VCO from two different chips have been measured. Each point on the graph is 10mV apart, and each point is an average of 10 measurements for each input voltage. In other words, each measurements has an oversampling ratio (OSR) of 10. Here, one can see how process variations affect the conventional VCO. More spesifically, one can see how process variations affects the gain in frequency. On the graph, it's easy to see that chip 4 has less frequency gain than chip 5.

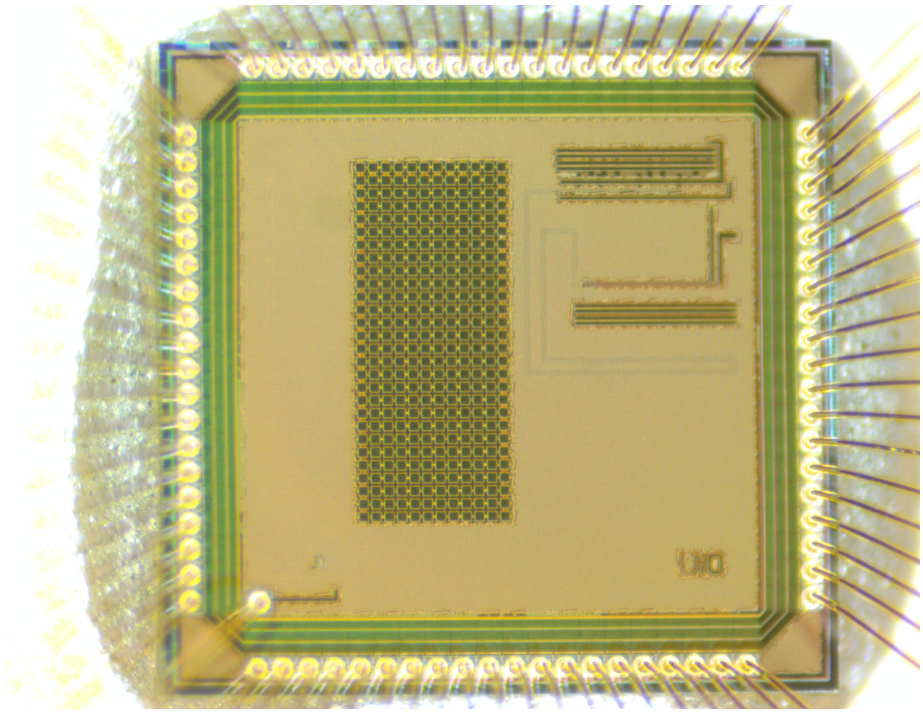


Figure 43: Picture of the chip produced, with some of the VCO structure visible in the top right corner

It's important to note that the input voltages don't go all the up to VDD, as opposed to the simulations. The reason is that the output frequencies above 1.15V have such a small peak-to-peak voltage (V_{pp}) that there is no reason to measure higher voltages. Above 1.15V, the V_{pp} is around $100m$V, and the output frequencies swing around $700m$V. Considering that the output frequency doesn't reach VDD/2, it's no point in measuring any further.

In order to measure the linearity of the conventional VCO, a regression line was plotted alongside the VCO curve (see figure 45). The regression line is drawn so that it would be a best-fit for the conventional ADC, such that the maximum difference would be minimized. The VCO curve plotted alongside the regression line is the conventional VCO from chip 4.
In the graph, it's possible to see that the conventional VCO is more nonlinear than expected, as the frequency points appear both over and under the regression line. The regression line has a slope of 2.53MHz/10mV, which was found using `scipy.stats.lingress` function from scipy. The INL has been calculated by taking the values of the measured points and sub-

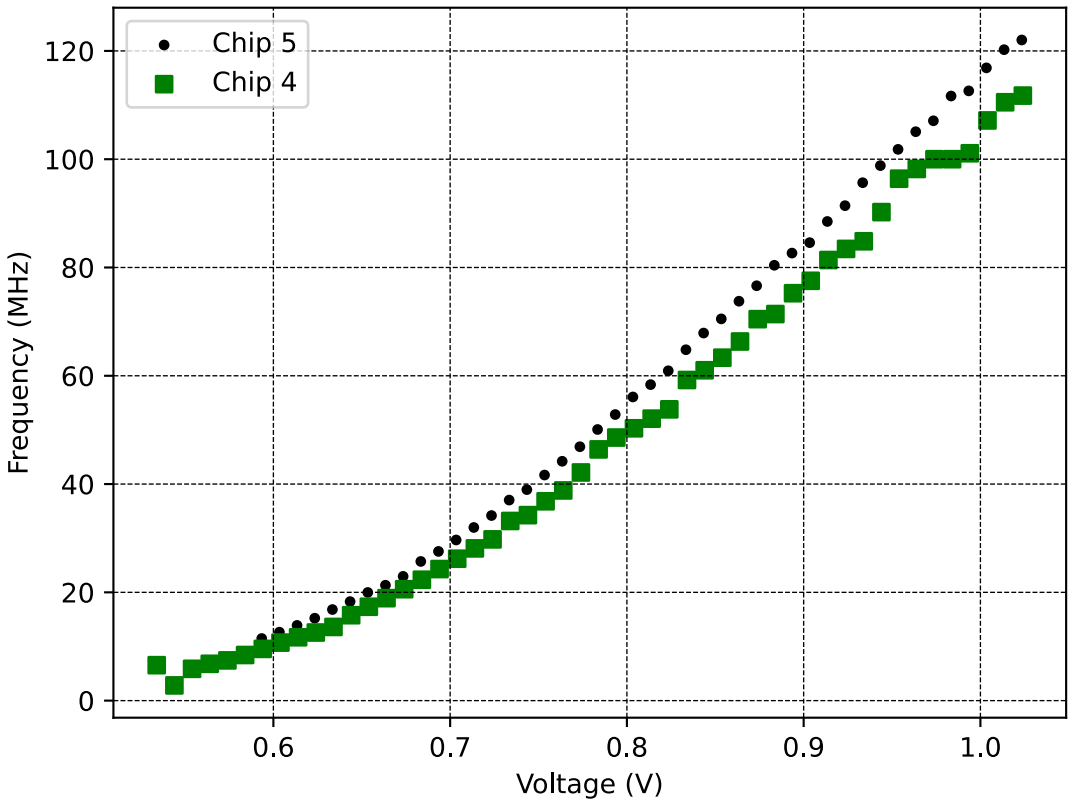


Figure 44: Plot of measurements of two conventional VCOs on separate chips. Plot shows Input Voltage vs. Output Frequency

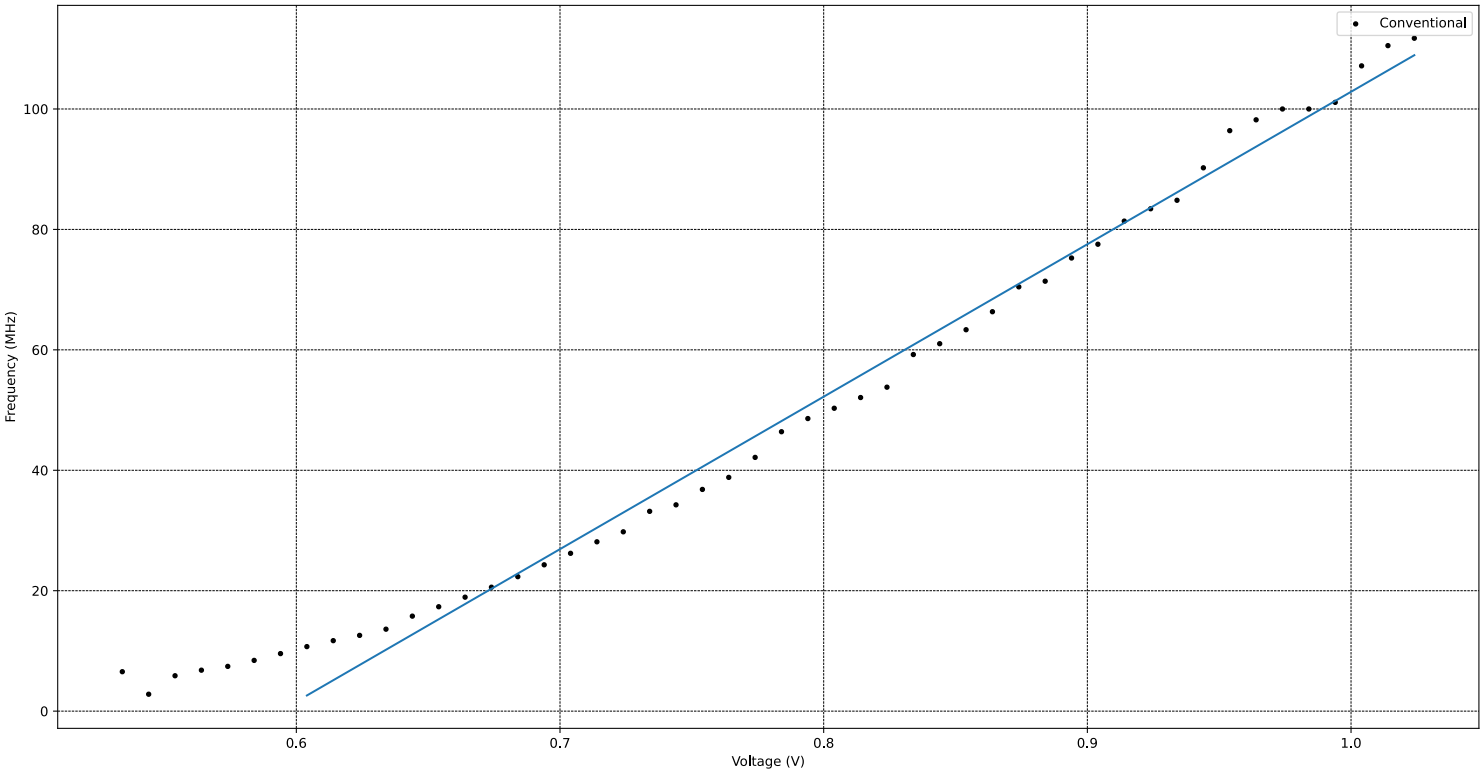


Figure 45: Plot of measurements from the conventional VCOs from chip 4, with a linear regression line that's based on the points in the VCO linear region. Plot shows Input Voltage vs. Output Frequency

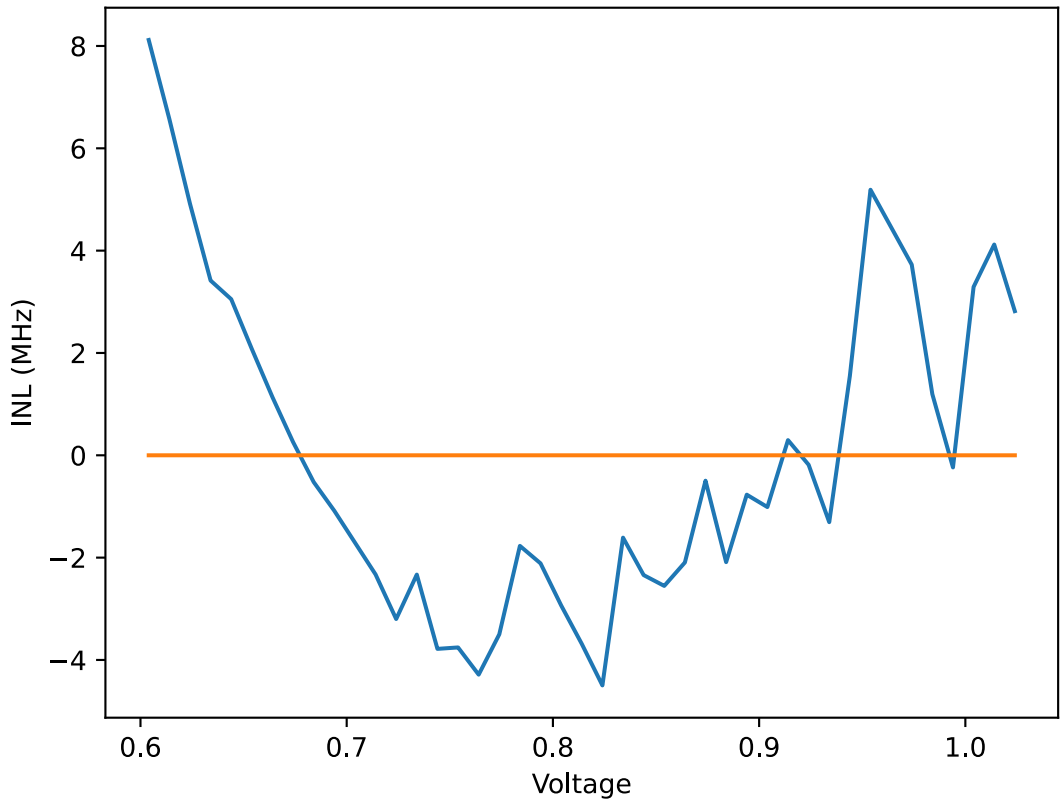


Figure 46: Plot of the INL error in the conventional VCO, given in MHz. Plot shows Input Voltage vs. INL (MHz)

tract them from the ideal points located on the regression line. The result of this can be seen in figure 46, which shows the INL error for each input voltage. On the graph, one can see that the largest INL error is actually at the lower voltages of the VCO. This is an indicator that the linear region is smaller than anticipated, and should be reduced. If the start of the linear region is shifted towards ≈ 0.64V, then the max INL would instead be the highest peak near 1V. The INL was measured on the top, and it's around 5.20MHz. When looking at the difference between each point (i.e. `INL[n+1] - INL[n]`, where n is a natural number), the DNL is calculated. The result of this can be seen in figure 47, which shows the DNL as a function of input voltage. Here, one can see that the large DNL errors are on the right side of the plot, meaning that the largest DNL errors are at the higher voltages (and thereby frequencies) of the VCO. In this case, even if the linear region was reduced, the DNL would still remain the same. The highest DNL error measured in figure 47 is approximately 3.6MHz. But one may wonder what the average INL and DNL may be? A script was written in python, and an average INL of ≈ 2.4MHz was found and an average DNL error of ≈ 1.17MHz.

Looking at the other performance parameters, the conventional VCO has reduced the linear region to $\approx 0.65V- \approx 1.01V$. The minimum frequency is now 17.3MHz, and the max frequency is 112MHz. The output frequencies generated from the VCO have been carefully observed, and there are little signs of any duty-cycle distortions, and the amplitude do reach up to VDD. Something interesting that was observed is that the buffered output frequencies are not square-waved, but sine-waved (see figure 42). This may indicate that there are a lot of parasitic capacitance on the measurement, which increases the rise-time and fall-time. There is also jitter in the output frequencies, which are strongest at lower frequencies and weaker at higher frequencies. The increased fall-time combined with jitter can cause huge uncertainties in the counting of the counter. Oversampling should make the uncertainties less of an issue, but it's still something to be aware of.

The resolution of the conventional VCO is defined by the max frequency minus the lowest possible freqency. which is $112-2.3 \approx 110$MHz. The SQNR is defined by the equation given in equation 7 and is 38.3. The dynamic range is $\frac{112}{17.3} = 6.5$, ENOB is INSERT LATER, power consumption is measured to vary between $1.8\text{V} \cdot 4\text{mA} = 0.007\text{mW}$ and $1.8\text{V} \cdot 5\text{mA} = 9\text{mW}$, sampling rate is 1μs and the area occupied is $10.61\mu\text{m} \cdot 7.14\mu\text{m} = 75.75\mu\text{m}^2$ (mentioned in

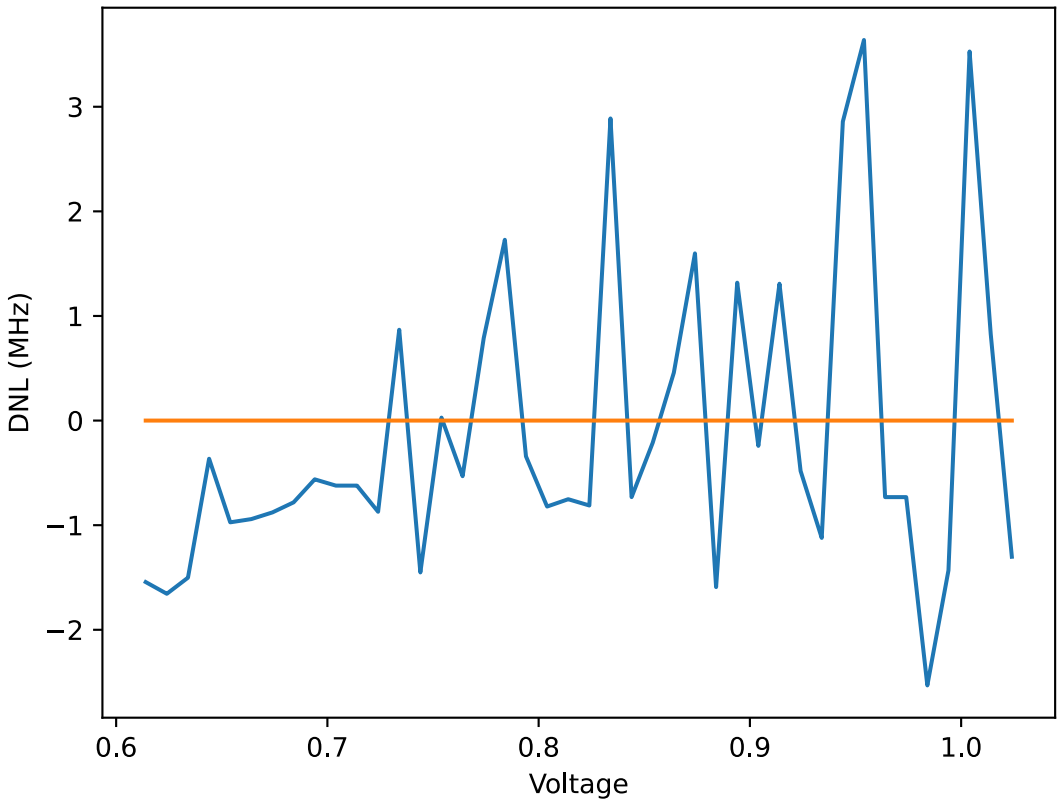


Figure 47: Plot of the DNL error in the conventional VCO, given in MHz. Plot shows Input Voltage vs. DNL (MHz)

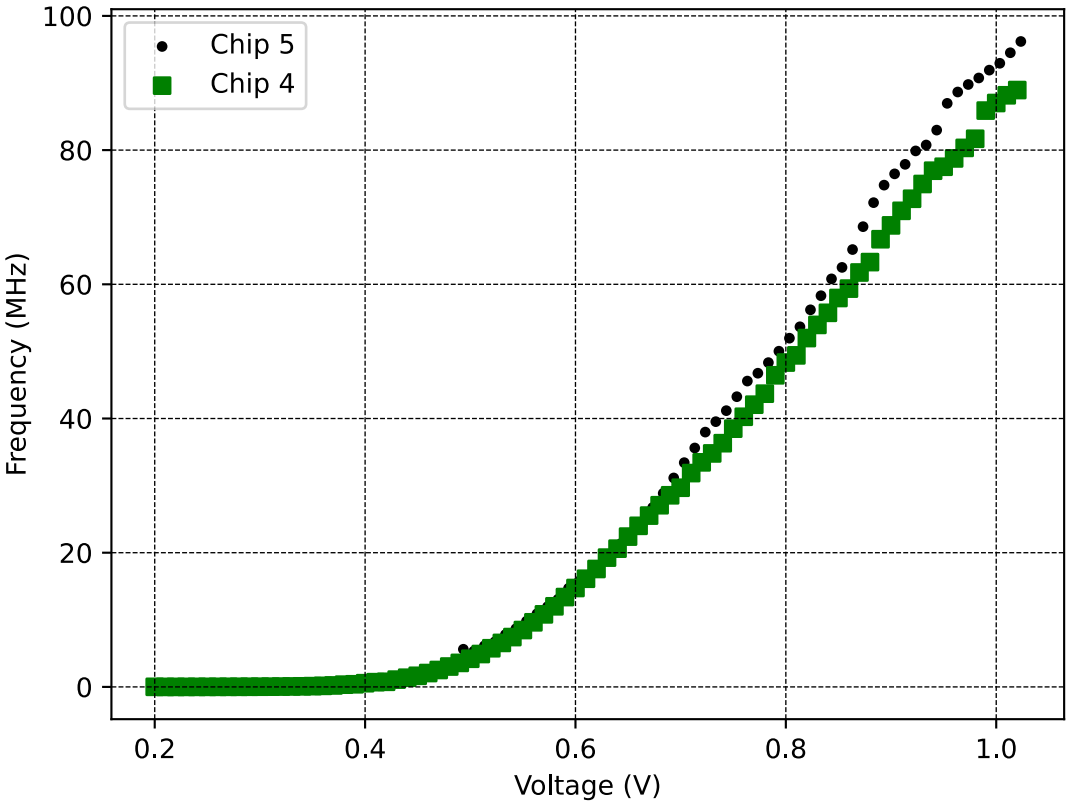


Figure 48: Plot of measurements of two FF VCOs on separate chips. Plot shows Input Voltage vs. Output Frequency

chapter 4).

A table summarizing these values are found in table 9, 10 and 11.

5.3.2 Feedforward VCO

The measurements of the FF VCO can be seen in figure 48, where two FF VCOs from different chips have been measured. Same as the conventional VCO, each point is an average of 10 measurements of each input voltage. In the plot, one can see that there is some difference in frequencies, due to process variations.
What's inveresting about the FF VCO is that it managed to oscillate at a lot lower frequency that what has been simulated. In terms of frequency, the FF VCO in chip 4 manages to go all the way down to a couple of hundred

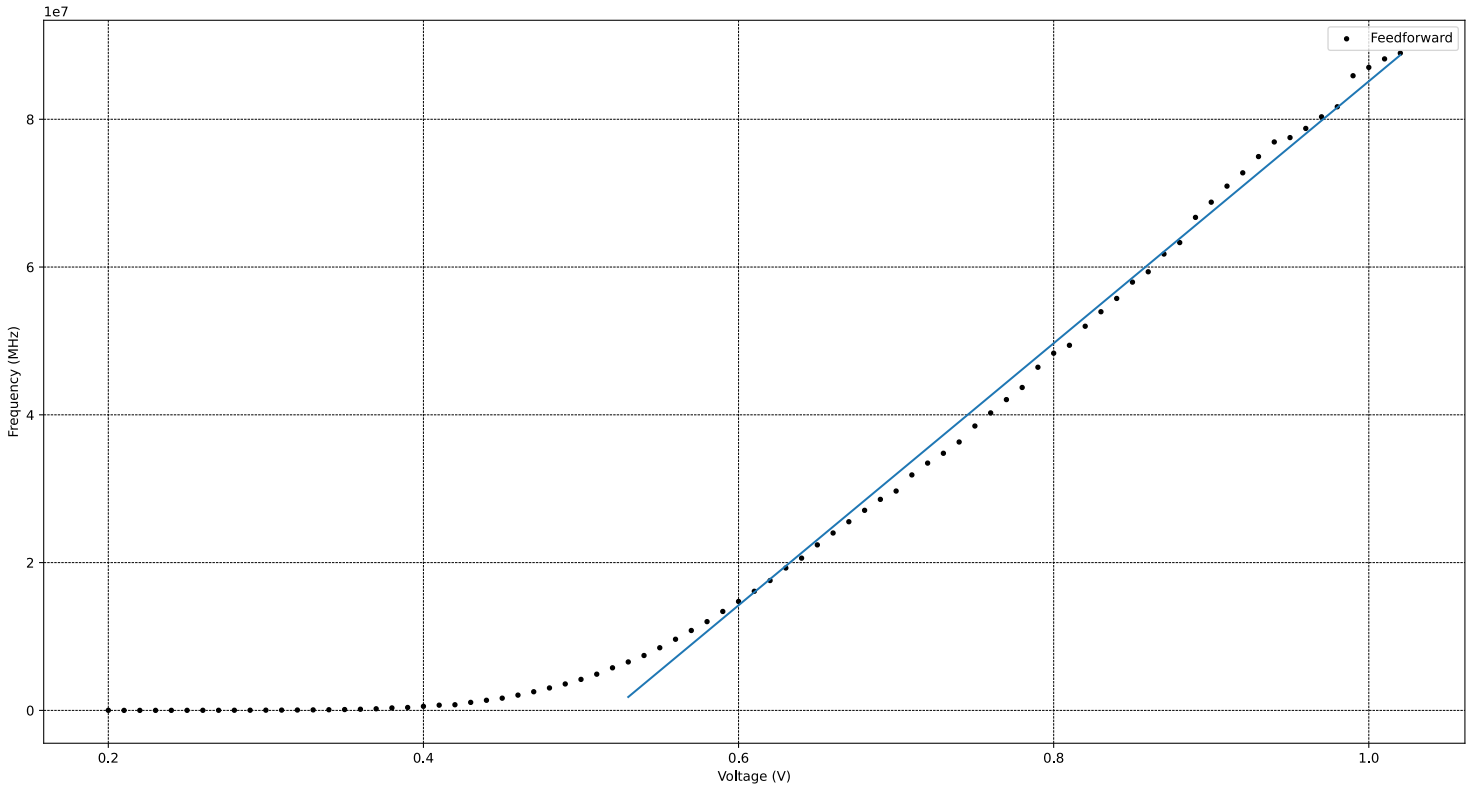


Figure 49: Plot of measurements from the conventional VCOs from chip 4, with a linear regression line that's based on the points in the VCO linear region. Plot shows Input Voltage vs. Output Frequency

kHz. Nevertheless, this is only seen in chip 4, as chip 5 can only go down to 0.5V before it stops oscillating. Also, the ability to have an input voltage as low as 0.2V is only trivial at best, as it doesn't improve the linear range nor dynamic range.

As with the conventional VCO, the output frequencies above 1.15V have such a small peak-to-peak voltage (V_{pp}) that there is no reason to measure higher voltages. The V_{pp} is $200m$V in this case, and the frequencies swing around $700m$V in the FF VCO as well.

In order to measure the linearity of the FF VCO, a regression line was plotted alongside the VCO curve (see figure 49). The regression line is drawn so that it would be a best-fit for the conventional ADC, such that the maximum difference would be minimized. The results plotted alongside the

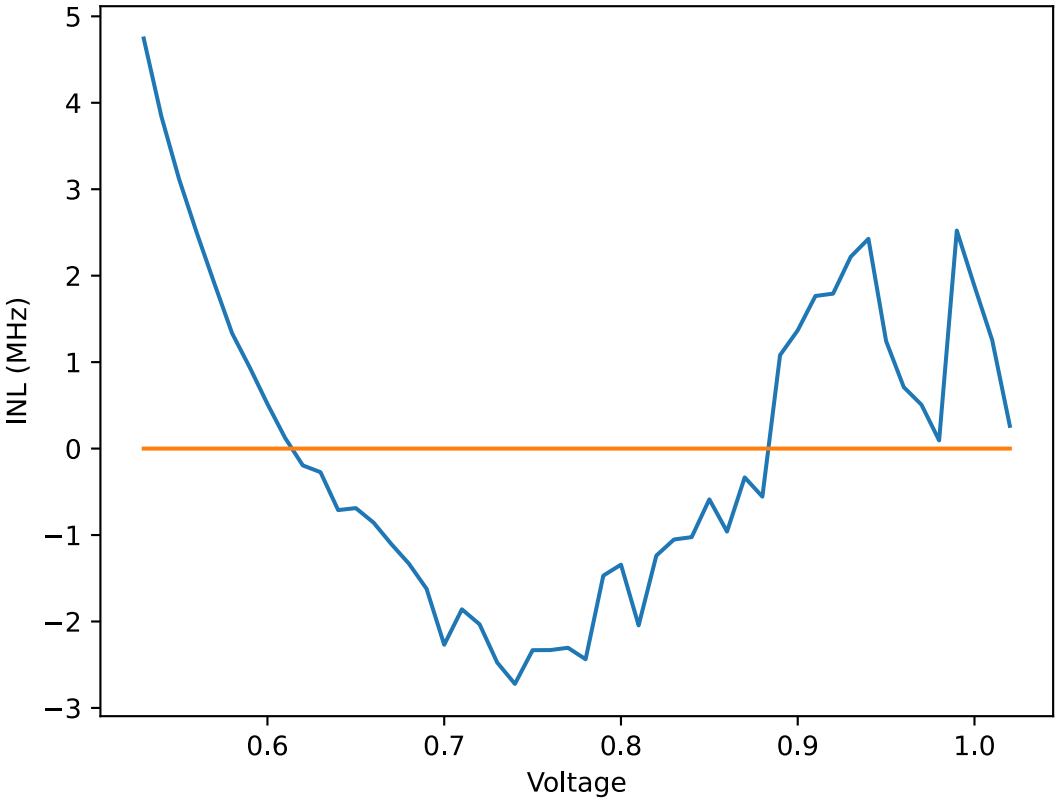


Figure 50: Plot of the INL error in the FF VCO, given in MHz. Plot shows Input Voltage vs. INL (MHz)

regression line is the FF VCO from chip 4.

As with the conventional VCO, the nonlinearity of the FF VCO becomes more apparent when plotted alongside a regression line. The slope of the FF VCO is 1.77MHz/10mV, which was also found by using `scipy.stats.lingress` function from scipy. INL of the FF VCO has been calculated the same way as the conventional VCO (take the values of the measured points and subtract them from the ideal points), and has the maximum value of 2.5MHz when reducing the linear range of FF VCO to ≈ 0.56V$- \approx 1.01$V. The plot of input voltage vs. INL can be seen in figure 50. The DNL is also calculated the same way as the conventional VCO

(i.e. `INL[n+1] - INL[n] for n in range(0, len(INL)-1)`)

and has the maximum value of 2.42MHz. It's plot can be seen in figure 51. Same as the conventional VCO, the largest DNL error is also found at the

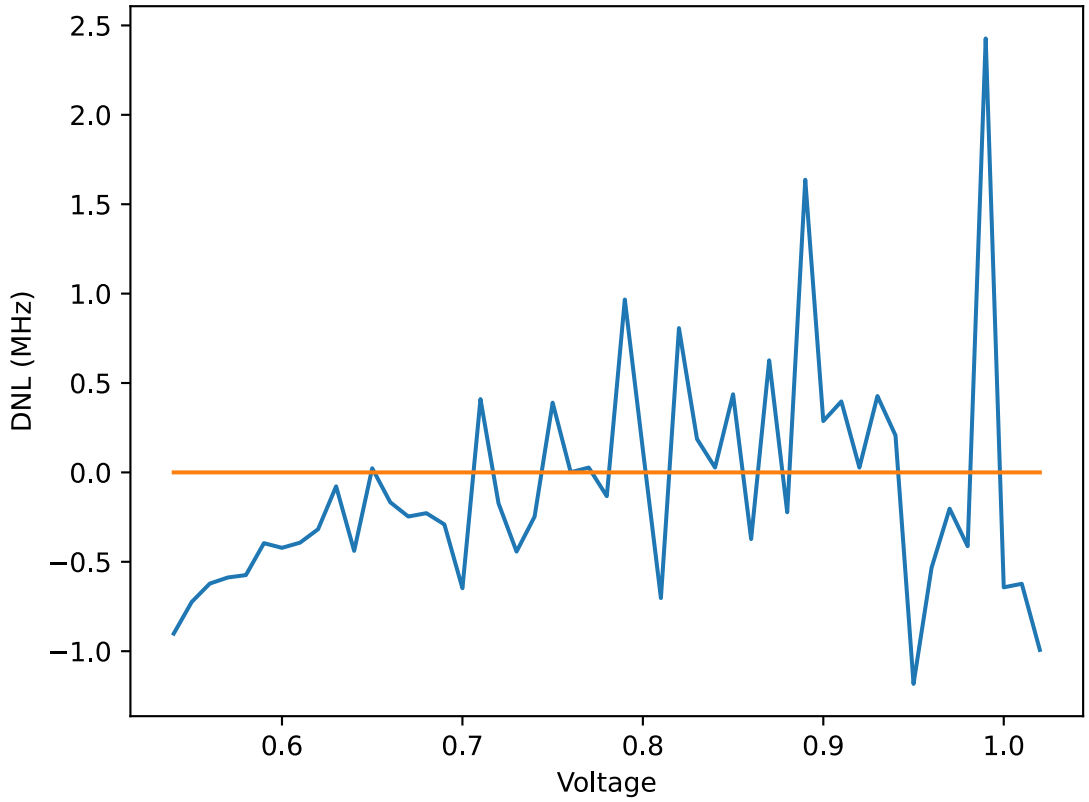


Figure 51: Plot of the DNL error in the FF VCO, given in MHz. Plot shows Input Voltage vs. DNL (MHz)

higher input voltages. The average INL of the FF VCO was found to be ≈ 1.33MHz and an average DNL error of ≈ 0.47MHz.

As with the conventional ADC, there are little signs of any duty-cycle distortions and amplitude reach up to VDD. The buffered outputs are also sine-waved, as opposed to being square-waved in the simulations. This also results in the uncertainties mentioned with the conventional VCO, which is something to think about, but not a huge issue.

The resolution of the FF VCO is defined by the max frequency minus the lowest possible freqency. which is $89.3 - 0.011 \approx 89$MHz. The SQNR is defined by the equation given in equation 7 and is 36.3. The dynamic range is $\frac{89.3}{9.6} = 9.3$, ENOB is INSERT LATER, power consumption is measured to

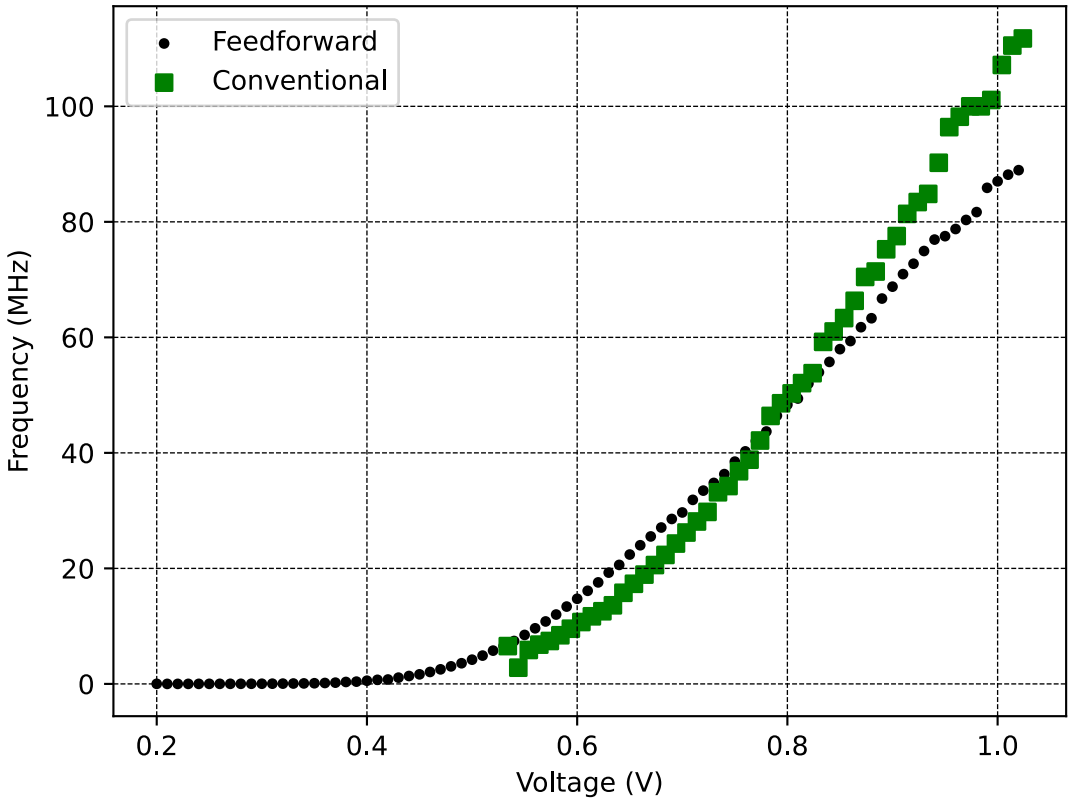


Figure 52: Plot of measurements of conventional VCO and FF VCO. Plot shows Input Voltage vs. Output Frequency

vary between $1.8\text{V} \cdot 5\text{mA} = 9\text{mW}$ and $1.8\text{V} \cdot 6\text{mA} = 10.8\text{mW}$, sampling rate is $1\mu\text{s}$ and the area occupied is $11\mu\text{m} \cdot 10.42\mu\text{m} = 114.6\mu\text{m}^2$ (mentioned in chapter 4).

A table summarizing these values are found in table 9, 10 and 11.

5.3.3 Comparison

A plot that shows the FF VCO from chip 4 and conventional VCO also from chip 4 can be seen in figure 52. Pros and cons between the two VCOs will be discussed in chapter 6.

	Min.freq(MHz)	Max.freq(MHz)	lin.reg
Conv	$\sim$17.3	$\sim$112	$\sim 0.65V - \sim 1.01V$
FF	$\sim$9.6	$\sim$89.3	$\sim 0.56V - \sim 1.01V$

Table 9: Part 1 of performance characteristics of FF VCO and conventional VCO from chip 4, with 1.8 VDD.

	Slope	INL	DNL	Avg.INL	Avg.DNL
Conv	2.53MHz/10mV	5.20MHz	3.6MHz	2.4MHz	1.17MHz
FF	1.77MHz/10mV	2.5MHz	2.42MHz	1.33MHz	0.47MHz

Table 10: Part 2 of performance characteristics of FF VCO and conventional VCO from chip 4, with 1.8 VDD.

5.3.4 Counter

Unfortunately, the counter was unable to be tested properly. It has been observed that the counter and the MUX is functional, as one of the output pins managed to go high (proving that the counter has managed to have at lease one bit high). However, the PCB was produced with an error as the digital output pins was shifted one row to the left. In other words, instead of connecting pin 58 to 66 on the PCB, pin 57 to 65 was connected instead. As pin 66 is PIN0, it became shorted to GND on the PCB, and thus can never go high. If the 66th pin ever goes high, a short between digital VDD and GND occurs and the whole chip gets burnt. Therefore, the only thing that is proven with the counter is that it is functional.

Specification	FF VCO	conventional VCO
Resolution	79.7 MHz	110MHz
SQNR	36.3	38.3
Max Power conumption	10.8mW	9mW
DR	9.3	6.5
ENOB	NA	NA
Sampling rate	1μs	1μs

Table 11: Part 3 of performance characteristics of FF VCO and conventional VCO from chip 4, with 1.8 VDD

6 Discussion

6.1 Figures of Merit

6.1.1 Resolution

As mentioned in chapter 3, the resolution is defined by the VCO frequency and the sampling frequency. As the sampling frequency is of the FF VCO and conventional VCO is the same, it's all dependent on the maximum frequency of the VCOs. As the conventional VCO has a higher maximum frequency than the FF VCO, the conventional VCO should have more resolution than the FF VCO.

However, the FF VCO can in theory have a higher oscillation frequency than the conventional VCO, which will be further discussed in later section (see section)

6.1.2 SQNR

When calculating the SQNR, one can use the formula that's mentioned in equation 7. Unfortunately, there is no information regarding the analog bandwidth ABW as the input was always assumed to be constant. This technically means that the analog bandwidth is 0 (or 1Hz if you count DC as a signal), which results in some strange numbers from equation 7. Therefore, another equation from [6] (which was the original equation cited in [3]) will be used instead (see equation 10). In this equation, M_q is the quantizer resolution which is described in equation 11, where $f_{tune} = f_0$, $f_s = f_s$, OSR is the oversampling ratio and N_{mp} is the number of the VCO phase. It's possible to see that $6.02M_q \approx 6\log_2(\frac{f_0}{f_s})$ if $N_{mp} = 1$. With this in mind, a modified version of equation 10 is made, as shown in equation 12. Assuming that the sampling frequency is 1μs and that $OSR = 10$ results in the numbers shown in table 11. Here, one can see that the SQNR is higher in the conventional VCO than the FF VCO, which makes sense given that the conventional VCO has a higher resolution than the FF VCO.

$$\begin{aligned}\mathrm{SQNR} = {} & 6.02M_q - 3.41 + 30\log \mathrm{OSR} \\ & + 20\log\left(\mathrm{sinc}\left(\frac{1}{2\mathrm{OSR}}\right)\right)\end{aligned} \tag{10}$$

$$M_q = \log_2 \left(K_{\text{vco}} 2AN_{\text{mp}} T_s\right) = \log_2 \left(\frac{f_{\text{tune}} N_{\text{mp}}}{f_s}\right) \tag{11}$$

$$\begin{aligned} \text{SQNR} = 6\log_2(\frac{f_0}{f_s}) - 3.41 + 30\log \text{OSR} \\ + 20\log\left(\text{sinc}\left(\frac{1}{2\text{OSR}}\right)\right) \end{aligned} \tag{12}$$

6.1.3 Voltage Range

The conventional VCO has a linear voltage range from 0.65V to 1.01V, while the FF VCO has a linear voltage range from 0.56V to 1.01V. It's clear that the FF VCO has a higher voltage range than the conventional VCO.

6.1.4 Dynamic Range

As mentioned in chapter 3, the dynamic range is defined as the ratio between the highest and lowest frequency in the linear range. Looking at the max and min frequencies from table 9, one can calculate that the DR of conventional VCO is $\frac{112}{17.3} = 6.47$ and the DR of FF VCO is $\frac{89.3}{9.6} = 9.3$. Therefore, the FF VCO has a higher DR.

6.1.5 ENOB

There was an attempt at finding ENOB through simplifying the SNDR to SNR. Unfortunately, there are some missing parameters that haven't been measured in order to calculate the SNR, which is then used to calculate the ENOB. The frequency gain K_{VCO} has been found by measuring the slope of the VCOs, if one can assume that the slope of the regression lines are the frequency gain. The amplitude of the output frequency A can also be measured, which varies from VDD down to 900mV, depending on the voltage of the input. The problem arises when trying to find the phase noise L, since it's not a straightforward method to measure the phase noise. Having a given frequency at the input is simple in theory: one can create a frequency with a given amplitude with a waveform generator and connect it to the VCO ADC input. However, one often measure the phase noise with the Fast Fourier Transform (FFT), which couldn't be done in time. As a result, ENOB hasn't been measured.

6.1.6 Power Consumption

When it comes to power consuption, the FF VCO has a higher power consuption than the conventional VCO. The reason for this is simply that the FF VCO has more inverters to power up than the conventional VCO. In the conventional VCO only five inverters need to be powered up, whereas in the FF VCO four direct path inverters and four feedforward inverters need to be provided with power for the FF VCO to function properly. Sizing on the NMOS and PMOS don't make a lot of a difference, as every inverter has the same width. Only the direct path inverter different dimensions, which has twice the length of the feedforward inverters and the conventional inverters. If one uses the square law model to calculate the current (square law model being $I_d = \frac{1}{2}k_n V_{ov}(1 - \lambda(V_{ds} - V_{ov}))$), four direct path inverters use more current than one inverter in the conventional VCO. Combine the four DP inverters with the FF inverters, and one can see that the FF VCO use slightly higher amount of current than the conventional VCO. However, the FF VCO can also be functional at a lower VDD than the conventional VCO at the cost of lower max frequency and lower voltage range. So although the power consumption is higher in the FF VCO than the conventional VCO, the FF VCO can use a lower VDD than the conventional VCO.

6.1.7 Sampling Rate

Both VCOs has the same sampling rate, so there is not much to discuss here when comparing the VCOs. Going back to the equation for SNQR (equation 7), the deciding factors are the VCO frequency f_0 and the analog signal bandwidth ABW as the sampling frequencies are the same for FF VCO and conventional VCO. Therefore, not much can be said of which VCO benefits more of an increase in sampling rate.

6.1.8 Area

Size is important in a chip, and a discussion of the VCOs' dimensions cannot be avoided. As the FF VCO consists of more inverters and transistors than the conventional VCO, it's also bigger than the conventional VCO. One can compare their sizes by looking at figure 32 and figure 33. If one looks at the VCOs themselves, then one can't deny that the FF VCO is bigger than the conventional VCO. More spesifically, the FF VCO is aprroximately $5\mu m$ wider than the conventional VCO. This is without considering the neccesary routing on the conventional VCO, which would reduce the width difference from $5\mu m$ to approximately $4\mu m$ (considering that routing

is done optimally, which wasn't done in the conventional VCO in this thesis). Lengthwise, both of the VCOs have a height of approximately $10\mu m$, which is mostly due to the body contacts of the transistors.

Something worth noting is that the VCO size can be compressed due to enabling transistors to share source/drain, something that has been used in this thesis, but not everywhere. On the conventional VCO and FF VCO layout (figure 32 and 33 respectively), one can see that it's possible in theory to put the inverters more closely together and make the NMOSes and PMOSes share source/drain. In practice, this isn't possible due to different sizes between PMOS and NMOS, which makes is so that the larges between the two (in this case, the PMOS) sticks out a bit too much on one side. This results in an inverter that is vertically aligned on one side, but not the other. If both sides were vertically aligned, then sharing source/drain everywhere would be possible. Unfortunately, it's not possible to do it here as PMOS and NMOS have different dimension, which means the minimum spacing requirement between vias routing are dictating the size of the VCOs instead. Nevertheless, having transistors share source/drain makes it so that the size of both VCOs can be compressed to the point where the size differences are barely noticable.

Another observation to be made is that although the conventional VCO uses less area, it also less area efficient. Looking at the layout from figure 33 and figure 32, one can see that although the FF VCO uses more area, it also has more area density to the conventional VCO. Since the conventional VCO has an odd number of inverters, there is always going to be some empty room in the layout. As the die produced if often square-shaped, the die area required to implement five inverters in a ring and four inverters in a ring is often requires the same die area. Of course, there is no denying that the FF VCO do require more area. However, it's not by a huge margin and the area efficiency of the FF VCO does make up for the increase in area.

6.1.9 Linearity

In terms of linearity, the conventional VCO has a max INL of 5.20MHz and a max DNL of 3.6MHz and the FF VCO has an INL of 2.5MHz and a DNL of 2.42MHz. Looking at the maximum values, the FF VCO has a better linearity than the FF VCO. Even when it comes to the average INL and DNL values, the FF VCO is more linear than its conventional counterpart.

7 Conclusion

In this thesis, a VCO-Based ADC was made as an alternative to the SS ADC. The VCO-Based ADC consist of a VCO, a counter and a register, with the VCO being the main focus of in the VCO-Based ADC. Two VCOs has been made in this thesis, which is the Feedforward VCO (FF VCO) amd the conventional VCO. The FF VCO is the VCO that's the important one among the two VCOs, as the conventional VCO is used as a comparison for the FF VCO. The VCO-Based ADC prototype has been designed in TSMC 180nm process, taped out and packaged in a JLCC-84 package. The FF VCO has a height of 11μm and a width of 10.42μm excluding transmission gates and current mirrors, with a linear input voltage range of 0.56V to 1.01V, a maximum oscillation frequency of of 89.3MHz, a minimum frequency of 9.6MHz, a Dynamic Range of $\frac{89.3\text{MHz}}{9.6\text{MHz}} = 9.3$, a max INL of 2.5MHz and a max DNL of 2.42MHz. The average INL and DNL of the conventional VCO are 2.4MHz and 1.17MHz respectively and the average INL and DNL are 1.33MHz and 0.47MHz respectively. The Maximum power consumption of the conventional VCO is 9mW and the Maximum power consumption of the FF VCO is 11mW. The resolution of the conventional VCO is based on the maximum and minimum oscillation frequency, which is 110MHz. For the FF VCO, the resolution is around 79.7MHz.

In comparison, the conventional VCO has a height of 10.61μm and a width of 7.14μm excluding transmission gates and current mirrors, with a linear input voltage range of 0.65V to 1.01V, a maximum oscillation frequency of 112MHz, a minimum oscillation frequency of 17.3MHz, a Dynamic Range of $\frac{112\text{MHz}}{17.3\text{MHz}} = 6.5$, a max INL of 5.20MHz and a max DNL of 3.6MHz.

The counter implemented in this thesis is a 12 bit ripple counter, which was realized with 12 JK-flip flops. The JK flip-flops were realized with three NAND gates, a D-flip-flop and an inverter. The ripple counter is able to count signals as fast as 1.5GHz, thereby making sure that it doesn't bottleneck the system. The counter is controlled with digital signals from a PYNQ Z1 FPGA that controls when the counter resets and when the counter starts counting.

The registers connected to the counter are implemented with 12 D flip-flops, which are then connected to a three MUXes. Control signals from a FPGA then control which bits the MUXes send out.

VHDL code was created in order to control the reset and read of the counter, which was implemented successfully in the PYNQ Z1 FPGA.

A PCB was designed to send in supply, signals and to measure outputs of the VCO-Based ADC system. It's possible to measure the VCO and make it

work, but it can't be done for the digital part due to a design error.

Future Work

Remove PMOS tail transistor

As Baert and Dehaene's paper states that having both PMOS and NMOS tail transistors is redundant [1], removing the PMOS tail transistor might be a good idea. The benefits include less area used on the die, since there is no need for current mirrors and transmission gates for the PMOS tail transistors. This might cause a slight decrease in tuning range and imbalance in duty cycle, but the tradeoff is worth it considering the area saved with the PMOS tail transistor removed.

Level shifters on PCB

In order to test the VCO connected with the counter, registers and MUX, off-board level shifter should be avoided due to parasitic capacitance and inductance from the jumper cables, which causes digital signals to become unstable. An easy solution to this problem is to avoid jumper cables entirely by implementing the level shifters directly onto the PCB. This way, the jumper cables are avoided with the additional benefit of having a better connection to GND and 5V VDD, something that should also improve the performance on the level shifters.

Make a new PCB

The PCB made in this thesis had its digital port connected to the wrong places. They should be changed so that the $PIN2$ output is floating instead of being permanently grounded. Even better, the output from the digital part should be fixed so that they are connected to the right outputs. As it is now, the RST output in figure 39 isn't connected to anything, READ is actually connected to RST on the chip, BIT0 is actually READ, and so on.

Measure phase noise and calculate ENOB

ENOB (and thereby SNDR and SNR) couldn't be calculated due the inability to measure the phase noise to calculate phase noise power. As ENOB is possibly one of the most important specification to measure, it would be highly desirable to do so as well.

8 Appendix

VHDL code for digital control signals of counter

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity pmod_ctrl is
  port
  (
    pinx   : in  std_logic_vector(2 downto 0);
    sw     : in  std_logic_vector(1 downto 0);
    mclk   : in  std_logic;
    start  : in  std_logic;
    reset  : in  std_logic;
    lenght : in  std_logic_vector(15 downto 0);
    rst    : out std_logic;
    en     : out std_logic;
    bit0   : out std_logic;
    bit1   : out std_logic;
    data   : out std_logic_vector(2 downto 0)
  );
end entity;

architecture rtl of pmod_ctrl is
  signal rst_end   :  std_logic := '0';
  signal rst_temp  :  std_logic := '0';
  signal en_start  : std_logic := '0';
  signal en_end    : unsigned(15 downto 0) := x"0000";
  signal counter   : unsigned(31 downto 0) := x"00000000";
  signal pressed   : std_logic := '0';
begin

  process(start,reset,mclk)
  begin

    if reset = '0' then
      rst_end  <= '0';
      rst_temp <= '0';
      en_start <= '0';
      en_end   <= x"0000";
      counter  <= x"00000000";
      pressed  <= '0';
    elsif rising_edge(mclk) then
      if start = '1' and counter = x"00000000" and pressed = '0' then
        pressed <= '1';
      end if;
      if pressed = '1' then
```

```
46          counter <= counter + 1;
47          if counter >= x"017D7840" then -- if 25e6
48            pressed <= '0';
49            counter <= x"00000000";
50            en <= '0';
51            rst_end   <= '0';
52            rst_temp  <= '0';
53            en_start  <= '0';
54            en_end    <= x"0000";
55          elsif counter <= unsigned(lenght) + 15 then
56            if rst_end = '0' then
57              rst <= '1';
58              rst_end <= '1';
59            elsif rst_end = '1' and en_end /= unsigned(lenght) and
    counter >= 14 then
60              rst <= '0';
61              en_start <= '1';
62            end if;
63            if en_start = '1' and en_end /= unsigned(lenght) then
64              en <= '1';
65              en_end <= en_end + 1;
66            end if;
67            if en_end = unsigned(lenght) then
68              en <= '0';
69              rst_end   <= '0';
70              en_start  <= '0';
71              en_end <= x"0000";
72            end if;
73          end if;
74      end if;
75    end if;
76  end process;
77
78
79  switches : process(sw,mclk)
80  begin
81    case sw is
82      when "00" =>
83       bit0 <= '0';
84       bit1 <= '0';
85      when "01" =>
86       bit0 <= '1';
87       bit1 <= '0';
88      when "10" =>
89       bit0 <= '0';
90       bit1 <= '1';
91      when "11" =>
92       bit0 <= '1';
93       bit1 <= '1';
94      end case;
```

```
  end process;

  reading : process(pinx)
  begin
    case pinx is
      when "000" =>
       data <= "000";
      when "001" =>
       data <= "001";
      when "010" =>
       data <= "010";
      when "011" =>
       data <= "011";
      when "100" =>
       data <= "100";
      when "101" =>
       data <= "101";
      when "110" =>
       data <= "110";
      when "111" =>
       data <= "111";
    end case;
  end process;
  end architecture;
```

Matlab code for generating a input/output points

```
function [volt,freq] = MeasureVCO()
%[volt,freq] = MeasureVCO()
%  applying voltages to voltage source HPE3631 and reading
% frequencies on HP54622 Oscilloscope
volt=0.5834:0.01:1.024; %chip 4 : conv at 0.534V and max at 1.024V, FF
    at 0.2V and max 1.15V
freq=zeros(size(volt));
j=1;
for v=volt
    HPE3631_SetVolt(1, v,HPE3631_DefaultAdr);
    f=zeros(1,10);
    for i=1:10
        f(i)=HP54622_MeasFreq(1,HP54622_DefaultAdr)
        if f(i) == 9.9*1.0e+37
            f(i)= 0
        end
    end; %for i
    freq(1,j)= mean(f)
    j=j+1
end; %for v
plot(volt,freq);
end
```

Python code for plotting and linear regression

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

def read_data(file):
    openfile = open(file, "r")
    lines = openfile.readlines()
    openfile.close()
    data = []

    for line in lines:
        vals = line.split(";")
        data.append(vals)

    x_val = []
    y_val = []

    for i in range(1,len(data)):
        vals = data[i][0].split(",")
        x_val.append(float(vals[0]))
        y_val.append(float(vals[1]))
    #print(x_val[30])
    return np.array(x_val), (np.array(y_val)/1e6)



def read_data_im(file):
    openfile = open(file, "r")
    lines = openfile.readlines()
    openfile.close()
    data = []

    for line in lines:
        #vals = line.split(";")
        vals = line.split(",")
        data.append(vals)

    x_val = []
    y_val = []

    for i in range(len(data[0])):
        x_val.append(float(data[0][i]))
        y_val.append(float(data[1][i]))

    return np.array(x_val), np.array(y_val)#/1e6


def deriv(x,y):
```

```
    a_list = []
    for i in range(len(x)-1):
        dy = y[i+1]-y[i]
        dx = x[i+1]-x[i]
        a = dy/dx
        a_list.append(a)
    return a_list

def diff(x,y):
    diff_x = []
    diff_y = []
    for i in range(len(x)-1):
        dx = x[i+1]-x[i]
        dy = y[i+1]-y[i]
        diff_x.append(dx)
        diff_y.append(dy)
    return diff_x,diff_y

def myfunc(x):
    return slope * x + intercept


if __name__ == "__main__":
    d = 33 #7 #33
    var = 5 #2 #5
#     x,y = read_data("FF_meas.csv")
#     x,y = read_data("ff_VCO_1_8_V.csv")
#     x1,y1 = read_data("ff_VCO_1_8_V_PEX.csv")
#     x,y = read_data("conv_VCO_smol_freq_inside.csv")
    x,y = read_data_im("ff_real.csv")
    # x,y = read_data_im("conv_real.csv")
#     x = x*1e6
#     print(diff_y)
#     x,y = read_data("ff_smol_freq_v2.csv")
#     deriv_ff = deriv(x,y)
#     deriv_conv = deriv(x1,y1)
#     x,y = read_data_im("conv_real_2.csv")
    y = y/1e6
    diff_x, diff_y = diff(x,y)
#     plt.plot(x,y, color = "black")
    slope, intercept, r, p, std_err = stats.linregress(x[d:], y[d:])
    model = list(map(myfunc, x[d:]))
    print(slope)
    plt.plot(x[d:],model)
    plt.scatter(x,y, label = "Feedforward", color = "black", marker = "
    .")
#     plt.scatter(x1[2:],y1[2:], label = "post-PEX", color = "green",
    marker = ".")
#     plt.plot(x,y, label = "Feedforward", color = "blue")
    plt.xlabel("Voltage (V)")
```

```
#    plt.xlabel("time (us)")
#    plt.ylabel("Amplitude (V)")
    plt.ylabel("Frequency (MHz)")
    plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
    plt.legend()
#    plt.savefig("VCO_compar.svg")
#    plt.savefig("FF_meas.svg")
    plt.show()
    plt.plot(x[d:], y[d:]-model)
    plt.plot(x[d:],np.zeros(len(x[d:])))
    plt.xlabel("Voltage")
    plt.ylabel("INL (MHz)")
    plt.show()
    fake, DNL = diff(x[d:], y[d:]-model)
    print(max(y[d:]-model))
    print(max(DNL))
    plt.plot(x[d+1:], DNL)
    plt.plot(x[d+1:],np.zeros(len(x[d+1:])))
    plt.xlabel("Voltage")
    plt.ylabel("DNL (MHz)")
    plt.show()
    INL = []
    y_im = y[d+var:]
    model_im = model[var:]
    for i in range(len(y_im)):
        element = y_im[i]-model_im[i]
        INL.append(abs(element))
    print(INL)
    print(sum(INL) / len(INL))
    for i in range(len(DNL)):
        DNL[i] = abs(DNL[i])
    print(DNL)
    print(sum(DNL) / len(DNL))
"""
    plt.plot(x[0:-1:10],deriv_ff[::10], label = "Feedforward", color =
    "black")
    plt.plot(x1[0:-1:10],deriv_conv[::10], label = "Conventional",
    color = "green")
    plt.xlabel("Voltage (V)")
    plt.ylabel("Slope")
    plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
    plt.legend()
#    plt.savefig("FF_meas.svg")
    plt.show()
"""
"""
    a_list = deriv(x,y)
    print(a_list)
    plt.plot(x[0:-1],a_list, label = "Feedforward", color = "blue")
    plt.xlabel("Voltage (V)")
```

```
    plt.ylabel("stigningstall")
    plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
    plt.legend()
    #    plt.savefig("FF_meas.svg")
    plt.show()
"""
"""
for file in ["conv_VCO_1_8_V_high_freq.csv","conv_VCO_1_8_V_smol_freq.csv",
"conv_VCO_1_8_V_high_freq_PEX.csv", "conv_VCO_1_8_V_smol_freq_PEX.csv",
"ff_VCO_1_8_V_high_freq.csv", "ff_VCO_1_8_V_smol_freq.csv",
"ff_VCO_1_8_V_high_freq_PEX.csv", "ff_VCO_1_8_V_smol_freq_PEX.csv"] :
x,y = read_data(file)
x = x*1e6
plt.plot(x,y, label = "VDD = 1.8V", color = "black")
plt.xlabel("time (us)")
plt.ylabel("Amplitude (V)")
plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
#plt.legend()
#plt.savefig(f"{file[:-4]}.svg")
plt.show()
```

```
    for file in ["ff_VCO_1_8_V.csv", "ff_VCO_1_8_V_PEX.csv", "
    conv_VCO_1_8_V.csv", "conv_VCO_1_8_PEX.csv"]:
        x,y = read_data(file)
        plt.plot(x,y, label = "VDD = 1.8V", color = "black")
        plt.xlabel("Voltage (V)")
        plt.ylabel("Frequency (MHz)")
        plt.grid(color = 'black', linestyle = '--', linewidth = 0.5)
        plt.legend()
        plt.savefig(f"{file[:-4]}.svg")
        plt.show()

"""
```

References

[1] Maarten Baert and Wim Dehaene. “A 5-GS/s 7.2-ENOB Time-Interleaved VCO-Based ADC Achieving 30.5 fJ/cs”. In: *IEEE Journal of Solid-State Circuits* 55.6 (2020), pp. 1577–1587. DOI: 10.1109/JSSC.2019.2959484.

[2] T.C. Carusone, D. Johns, and K.W. Martin. *Analog Integrated Circuit Design*. Analog Integrated Circuit Design. Wiley, 2012. ISBN: 9781118092330. URL: https://books.google.no/books?id=hNvNygAACAAJ.

[3] G. G. E. Gielen, L. Hernandez, and P. Rombouts. “Time-Encoding Analog-to-Digital Converters: Bridging the Analog Gap to Advanced Digital CMOS-Part 1: Basic Principles”. In: *IEEE Solid-State Circuits Magazine* 12.2 (2020), pp. 47–55. DOI: 10.1109/MSSC.2020.2987536.

[4] A. Hajimiri, S. Limotyrakis, and T.H. Lee. “Jitter and phase noise in ring oscillators”. In: *IEEE Journal of Solid-State Circuits* 34.6 (1999), pp. 790–804. DOI: 10.1109/4.766813.

[5] M. Hovin et al. “A narrow-band delta-sigma frequency-to-digital converter”. In: *1997 IEEE International Symposium on Circuits and Systems (ISCAS)*. Vol. 1. 1997, 77–80 vol.1. DOI: 10.1109/ISCAS.1997.608538.

[6] Jaewook Kim et al. “Analysis and Design of Voltage-Controlled Oscillator Based Analog-to-Digital Converter”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 57.1 (2010), pp. 18–30. DOI: 10.1109/TCSI.2009.2018928.

[7] Ko-Chi Kuo. “A Novel Linear-Logarithmic Active Pixel CMOS Image Sensor with wide dynamic range”. In: *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2021, pp. 1100–1103. DOI: 10.1109/MWSCAS47672.2021.9531713.

[8] Deyan Levski, Martin Wäny, and Bhaskar Choubey. “Ramp Noise Projection in CMOS Image Sensor Single-Slope ADCs”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.6 (2017), pp. 1380–1389. DOI: 10.1109/TCSI.2017.2662950.

[9] Junichi Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. Taylor Francis, 2006.

[10] Neelakantan Narasimman and Tony Tae-Hyoung Kim. “Design challenges for VCO based ADCs for ultra-low power operation”. In: *2013 International SoC Design Conference (ISOCC)*. 2013, pp. 249–252. DOI: 10.1109/ISOCC.2013.6864019.

[11] P.A.J. Nuyts, P. Reynaert, and W. Dehaene. *Continuous-Time Digital Front-Ends for Multistandard Wireless Transmission*. Analog Circuits and Signal Processing. Springer International Publishing, 2014. ISBN: 9783319039251. URL: https://books.google.it/books?id=QgS7BAAAQBAJ.

[12] Himchan Park et al. "Low Power CMOS Image Sensors Using Two Step Single Slope ADC With Bandwidth-Limited Comparators amp; Voltage Range Extended Ramp Generator for Battery-Limited Application". In: *IEEE Sensors Journal* 20.6 (2020), pp. 2831–2838. DOI: 10.1109/JSEN.2019.2957043.

[13] Young-Seok Park, Pyung-Su Han, and Woo-Young Choi. "Linear analysis of feedforward ring oscillators". In: *IEICE transactions on electronics* 93.9 (2010), pp. 1467–1470.

[14] B. Razavi. "A study of phase noise in CMOS oscillators". In: *IEEE Journal of Solid-State Circuits* 31.3 (1996), pp. 331–343. DOI: 10.1109/4.494195.

[15] Min-Woong Seo et al. "2.45 e-RMS Low-Random-Noise, 598.5 mW Low-Power, and 1.2 kfps High-Speed 2-Mp Global Shutter CMOS Image Sensor With Pixel-Level ADC and Memory". In: *IEEE Journal of Solid-State Circuits* 57.4 (2022), pp. 1125–1137. DOI: 10.1109/JSSC.2022.3142436.

[16] Johannes Sølhusvik. "Lecture 1 – Camera systems overview". https://www.uio.no/studier/emner/matnat/ifi/IN5350/h21/timeplan/in5350_h21_1_overview_27aug2021.pdf. Accessed: 2022–07-05. 2021.

[17] *TXB0104 4-Bit Bidirectional Voltage-level Translator With Automatic Direction Sensing and ± 15-kV ESD Protection*. TXB0104. Rev. J. Texas Instruments. 2020. URL: https://www.ti.com/lit/ds/symlink/txb0104.pdf?ts=1653490595374&ref_url=https%5C%253A%5C%252F%5C%252Fwww.google.com%5C%252F.

[18] Wikipedia contributors. *Ring oscillator*. [Online; accessed 14-June-2022]. 2021. URL: https://en.wikipedia.org/wiki/Ring_oscillator.

[19] Dag Wisland and Jørgen Andreas Michaelsen. "Phase locked loops". https://www.uio.no/studier/emner/matnat/ifi/IN5220/v22/timeplan/in5220_v22_10_pll.pdf. Accessed: 2022–07-12. 2022.