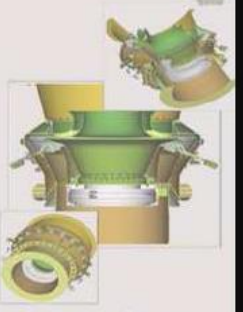
Faculty of Engineering - Cairo University
Credit Hour System Programs

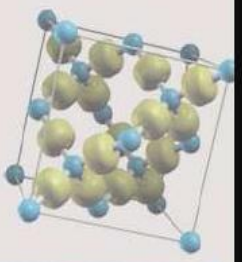


Communication and Computer Engineering
(CCE)

Graduation Project Report
Spring 2013

Digital *TV* Tuner Front End Design

Part B: *PLL* and *VGA*

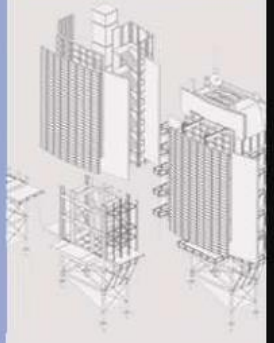


Prepared by

1. Ahmed Mahmoud Gaber (1082029)
2. Ahmed Tarek Ibrahim (1082054)
3. Ayman Hassen Dorrah (1081154)
4. Mohammad El-Sayed Khalifa (1082002)

Supervised by

Dr. Mohamed Aboudina



Acknowledgments

It is a pleasure to acknowledge the assistance received from several individuals during the preparation of this graduation project. No project of this magnitude could be completed without professional help, thus we are grateful to thank:

Dr. Mohamed Aboudina

from the Department of Electronics and Electrical Communications Engineering, Cairo University – for his sincere support and encouragement during the implementation of the project.

Table of Contents:

List of Tables:

List of Figures:

Part B-1: Phase Locked Loops:

Accurate frequency synthesizers are required in all *RF* communication systems. Most *RF* synthesizers employ the concept of "***Phase Locking***" to achieve the required frequency accuracy. In this part, we will talk about our design of the *PLL* for the Digital *TV* system.

The digital televisions (*DTV*) lead the way to a new era in television (*TV*) entertainment. Since *DTV* utilizes the frequency band more efficiently and provides the audience with better experience of watching *TV*, many countries all over the world have planned to move their *TV* broadcasting systems from analog to digital in the near future and have announced their choice for the *DTV* standard. Currently, the following three major *DTV* standards, which basically occupy the same frequency band, are adopted: Advanced Television Systems Committee (*ATSC*), Digital Video Broadcasting-Terrestrial (*DVB-T*), and Integrated Services Digital Broadcasting-Terrestrial (*ISDB-T*).

There are many challenges in designing a multi standard *DTV* tuner, particularly in the design of the frequency synthesizer. The first and the most difficult one is the wide tuning range. The Synthesizer needs to cover from *48* to *862MHz* to generate the required local oscillation (*LO*) signal in a direct conversion receiver, which means that it must have a tuning range larger than *200%*.

The second challenge is the strict phase noise requirement. Because of the wide tuning range the voltage-controlled oscillator (*VCO*) in the frequency synthesizer needs to have a very large *VCO* gain (K_{VCO}) and, thus, becomes sensitive to the noise voltage on the control line. Therefore, we have to minimize the *VCO's* phase noise as much as possible. This problem is typically solved by using two or three *VCOs* to cover the wide frequency range. Nevertheless, this approach draws more power and increases the total chip area, which is not a satisfactory solution.

The third challenge is the different channel spacing for each standard, which is *6MHz* in *ATSC*, *6, 7*, and *8MHz* in *DVB-T*, and *6* and *3/7MHz* in *ISDB-T*. It is hard to fulfill all these frequency resolution requirements using an integer-*N* phase-locked loop (*PLL*), unless a small reference frequency is used. However, narrow channel spacing leads to large division ratio that inevitably raises the in-band phase noise. Therefore, fractional dividers can be used to solve this problem.

Chapter 1 provides mathematical derivations for the phase models that are used to model the performance of the different types of *PLLs*. It presents the drawbacks of each type and how they can be treated. A brief comparison between the types of *PLLs* is also included. This chapter concludes with the design procedures of the loop filter used in the *PLL* for the *TV* Tuner System. These design procedures guarantee the maximum phase margin possible and highest suppression of ripples on the control voltage. The building blocks that constitute the *PLL* of the *TV* Tuner system are designed and implemented in the chapters two through five.

Chapter 2 is divided in two main sections; the first one explains the circuit of the phase detector and its implementation. It also presents the drawbacks of the phase detector and

demonstrates the need for the superior phase/frequency detectors .The second section emphasizes the difference between the Phase frequency detectors and the phase detectors and discusses the disadvantages of the *PFD*. The chapter concludes with proposing a digital circuit that solves the dead zone problem.

Chapter 3 discusses the importance of the charge pump in *PLLs* and its impact in controlling the range of frequencies by driving the *VCO*. In addition, this chapter reviews the basic challenges in the design of a charge pump. These challenges include the limited output swing of most charge pumps. A feedback-based solution is presented that improves the output swing with highly-matched up and down currents. Finally, a wide-swing charge pump is designed using 0.13 μm *CMOS* technology where simulations prove the superiority in the output swing when compared with other *CP* designs.

Chapter 4 provides a comprehensive review of the different candidate *VCOs* used for the *TV* Tuner system. The chapter starts by explaining how a circuit can maintain an oscillating output with constant frequency and how this oscillation frequency can be tuned by a control voltage is presented. Then, the concept of *LC VCOs* is introduced together with design procedures and simulation results of the three most famous *LC VCO* topologies. A brief comparison between these three topologies is also provided. This comparison demonstrates the need for Ring oscillators to cover the wide tuning range of frequency operation. Therefore, design procedures and simulation results are also provided for three distinct Ring VCO topologies. The chapter concludes with a brief comparison between the three Ring *VCO* topologies and a four-stage ring topology is demonstrated as the best topology for the *TV* Tuner System.

As for frequency dividers, they achieve frequency multiplication allowing the *PLL* to be used as a frequency synthesizer. Chapter 5 discusses the different aspects of frequency dividers design. It starts by introducing the possible approaches for the basic building block of the divider and discussing the pros and cons of each of them. Some techniques that enable modifying circuits in order to reach modulus other than *2* are also presented. Then, the most popular and commonly used frequency divider which is called "*Pulse Swallow Divider*" is implemented. Its theory of operation is studied to gain insight on how to modify it to reach the most beneficial results. Finally, a complete design of a frequency divider for the *PLL* of the TV tuner according to *ASTC* standards is implemented.

On the other hand, there is another class of *PLLs* namely fractional *PLLs* that allow using a finer frequency resolution without employing a smaller reference frequency or loop bandwidth which minimizes the noise on the control line of the *VCO.* Chapter 6 starts by discussing the theory of fractional-N *PLLs* and their different aspects such as spurs elimination and noise shaping. Then, it illustrates the implementation of first and second order digital $\sum\Delta$-modulators and their basic components. Some techniques for quantization noise reduction are also introduced and the concept of dithering is defined. The chapter also includes a proposal for a new technique that eliminates spurs to a great extent. Finally, the chapter concludes by proposing a fractional-N *PLL* frequency planning along with a convenient design for the frequency divider.

1 Mathematical Modeling of *PLLs*

As will be discussed in the following chapters, the output phase of the *VCO* needs time to change (an impulse on the control voltage is needed to achieve an instantaneous output phase difference). Suppose that the *VCO* oscillates with the same frequency as an ideal reference but with a certain phase difference, it is required to adjust the output phase of the *VCO* (through the control voltage) and eventually nullify this phase difference. The *PLL* is said to have achieve phase-lock if the phase difference is constant (not necessarily zero) with time. This ensures that the *VCO* phase tracks the reference phase (output phase is locked to the input phase).

To achieve phase-lock, we need to increase (decrease) the frequency of the *VCO* until the phase difference vanishes. Then, we should change back the frequency of the *VCO* to be the same as that of the reference frequency. This is illustrated in Figure 1.1. At t equals to t_O, V_{CONT} is increased to increase the frequency of the *VCO*. When the phase difference becomes zero -at t equals to t_1- V_{CONT} is decreased back to its initial value and the two signals remain in phase-lock (having equal frequencies and no phase difference).

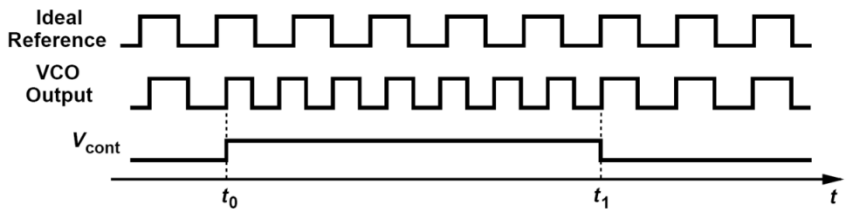


Figure 1.1: Achieving zero phase difference between an ideal reference and the output of a *VCO* by increasing its frequency and changing it back.

To realize the waveforms shown in Figure 1.1, a negative feedback loop is required. Figure 1.2 illustrates a simple negative feedback loop (*PLL*), where the *PD* is used to compare the reference and *VCO* phase difference and determine when this difference reaches zero. If the loop has a sufficient gain, the phase difference (error) is minimized in the steady state.

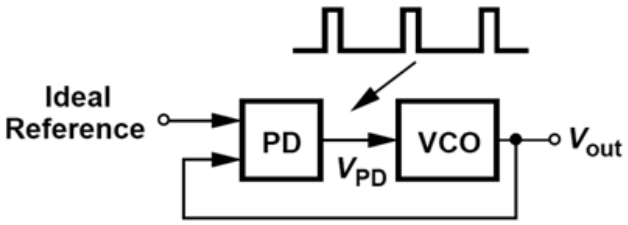


Figure 1.2: A simple *PLL*.

The *PLL* in Figure 1.2 suffers from a very crucial drawback which is modulating the *VCO's* output frequency as the *PD* generates repetitive pulse on the control voltage. This *VCO's* frequency modulation results in very large sidebands which contradicts the requirement of the *PLL* as a frequency synthesizer. Therefore, a loop filter is introduced between the *VCO* and the *PD* to provide a clean control voltage and suppress any modulating pulses. Figure 1.3

illustrates the *PLL* after the introduction of the low pass filter (*LPF*) which is realized by means of resistances and capacitances.

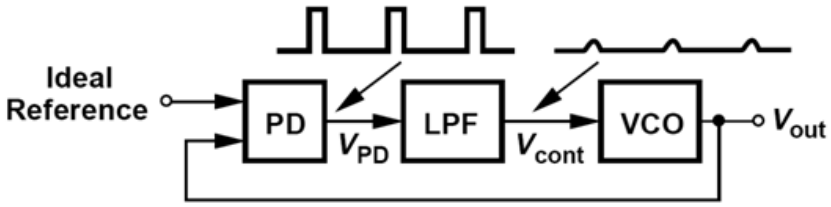


Figure 1.3: Introduction of a *LPF* to the simple *PLL* in order to remove the frequency-modulating components generated by the *PD*.

The *LPF* is necessary to the *PLL* because this feedback loop (as explained later) has a finite loop gain. As a result, the loop suffers from a finite phase difference in the steady state which means that the *PD* will also produce pulses in the steady state. Therefore, the *LPF* is required to suppress the pulses even after locking of the *PLL* and prevent the V_{CONT} from being disturbed. As for *PLLs* with infinite loop gain, there are some nonlinearities and non-idealities that also require the presence of the *LPF* to provide a clean V_{CONT} to the *VCO*.

There are mainly two approaches for modeling the performance of *PLLs* namely Voltage-Domain models and Phase-Domain models. Voltage-Domain models simulate voltages and signals that can also be measured and observed physically in the laboratory. For example, the output of a voltage-domain *VCO* is a periodic signal that resembles a clock. However, voltage-domain models require an extensive amount of time to run properly and simulate the nonlinearities in the system. This is caused by the fact that voltage-domain models describe the nonlinear effects by including the actual shapes of the *RF* waveforms (including spurs and harmonic locking when the *PLL* locks on to a harmonic of the reference frequency).

On the other hand, Phase-Domain models are exceptionally faster than their voltage-domain counter parts because they do not simulate any waveform effects. They can capture the nonlinearities in the system but the simulation is still behavioral. For example, the output of a phase-domain *VCO* is a voltage numerically equal to the output phase. Thus, phase-domain models do not account for inter-stage loading, improper bias and device parasitics. In spite of all of this, we have chosen to analyze the *PLL* using a Phase-Domain model because it is easier to understand and implement without losing any of the properties and the characteristics of the system. A brief summary of the differences between the Voltage and Phase Domain Models is provided in Table 1.1. In the following section we begin our analysis with the simplest *PLL* design (Type-I *PLLs*).

Table 1.1: A brief comparison between Voltage and Phase Domain Models.

	Phase-Domain Models	Voltage-Domain Models
Principle	No simulation of waveforms. Only behavioral simulation	Simulation of actual waveforms of voltages and signals
Complexity	Very simple, easy to understand and implement	Very complex because the nonlinearities in the waveforms are included
Simulation Time	Exceptionally fast	Require an extensive amount of time to run properly and simulate
Accuracy	Do not lose any properties or characteristics in the system.	Simulate the nonlinearities in the system.

1.1 Type-I *PLLs*:

As expressed in section 1.1.1, Type-I *PLLs* have a single pole at the origin in their open-loop transfer function. For Type-I *PLLs*, phase-locking is achieved when the difference between the output and reference phases is constant. As a consequence, this results in the output having the same frequency as the reference input. This property is clearly shown by the following equations:

$$\emptyset_{out}(t) - \emptyset_{in}(t) = constant \qquad \text{Equation 1.1}$$

$$\therefore \frac{d\,\emptyset_{out}}{dt}(t) = \frac{d\,\emptyset_{in}}{dt}(t)\,. \qquad \text{Equation 1.2}$$

On the other hand, if the phase difference between the output and reference signals varies with time, the loop is said to be “unlocked” i.e. the output does not track the input which is undesirable. This may occur if the *VCO* frequency is very far from the reference frequency at the start of operation.

As shown in Figure 1.4, when the loop is in the lock state, the output and reference frequencies are equal (using dividers later they will be multiples of each other) but a finite phase difference ($\Delta\Phi_1$), the *PD* generates pulses regularly (with width $\Delta\Phi_1$), the *LPF* suppresses the pulses and extracts the average level (the residual perturbation is called the ripple) and the *VCO* takes this *DC* voltage and produces an oscillating waveform with the required frequency. To further suppress these ripples a lower cut-off frequency *LPF* has to be used.

It is clear that the signal changes dimensions around the loop, i.e. the input to the *PD* and the output of the *VCO* are phase quantities whereas the output of the *PD* and *LPF* and the input to the *VCO* are voltage quantities

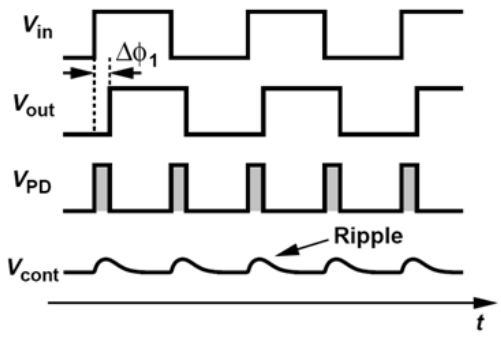


Figure 1.4: The change in signal dimensions across different nodes.

Another type of loops is called frequency-locked loops (*FLLs*) where the *PD* is replaced with a frequency detector (*FD*).As shown in Figure 1.5, this loop will never achieve phase-lock because in the steady state the frequency difference will not be zero (due to the finite loop gain and frequency offsets in the *FD*). Instead there will be a constant frequency difference between the reference and the *VCO* output. Which results in a time-varying phase difference and the loop will never acquire lock.

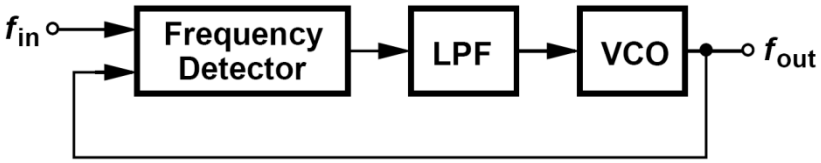


Figure 1.5: A frequency-locked loop (*FLL*).

Figure 1.6 illustrates the response of a Type-I *PLL* to a small positive frequency step $\Delta\omega$ in the reference frequency at $t = t_O$. The reference frequency is now greater than the output frequency, the reference phase accumulates faster and the phase difference starts to increase. As a result, the *PD* starts to generate wider pulses and the *DC* level of the control voltage (V_{CONT}) starts to increase. Consequently, the *VCO* output frequency increases and approaches the reference frequency until the width of the *PD* repetitive pulses starts to decrease and the loop eventually settles at the new frequency.

It is also clear that in order for the loop to acquire lock, the output and reference frequencies must be equal (frequency acquisition) (in general they must be multiples of each other) and the phase difference between $\Phi_{REFERENCE}$ and Φ_{OUT} must settle to a suitable constant value (phase acquisition).

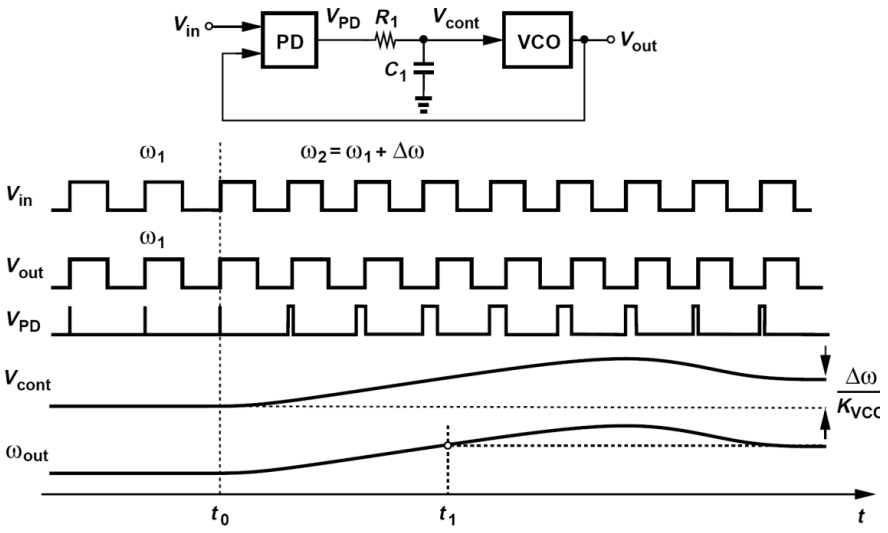


Figure 1.6: Response of a Type-I *PLL* to an input frequency step.

1.1.1 Phase-Domain Model for Type-I *PLLs*:

In general, *PLLs* are nonlinear but we will use a linear approximation for the transient response to determine tradeoffs in *PLL* designs by developing a linear transfer function. The transfer function of *PLLs* quantitatively determines the propagation of slow and fast input phase changes to the output. For example as shown in Figure 1.7, the output phase tracks the input phase for slow variations but loses track for faster input variations. The transfer function will determine this effect quantitatively.

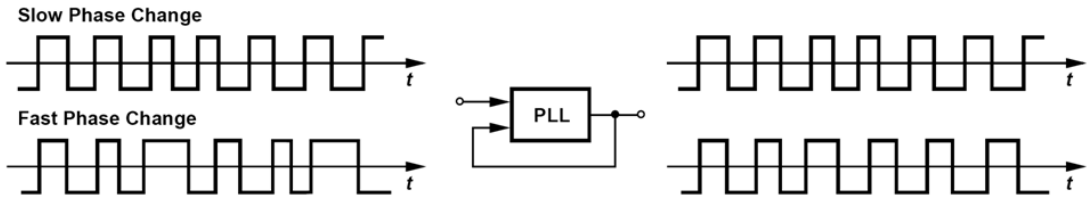


Figure 1.7: *PLLs* response to different input phase changes.

Figure 1.8 illustrates a typical phase-domain model for Type-I *PLLs*. The average voltage (V_{PD}) -which models the repetitive pulses- is generated by simply subtracting the output phase (Φ_{OUT})from the input phase (Φ_{IN})and amplifying the result by the gain of the *PD* (K_{PD} *(V/rad)*). This average voltage is applied to a first-order *LPF* and the result is used as the *VCO* input (V_{CONT}). The *VCO* is modeled as a block with a voltage input and a phase output (to be connected to the *PD* which only senses phase variations). As derived in the following chapters, the transfer function of the *VCO* block in the phase-domain is simply modeled as "K_{VCO}/s" where K_{VCO} is the gain or sensitivity of the *VCO* measured in *(rad/s)/V*.

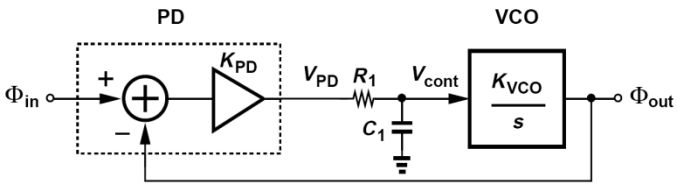


Figure 1.8: Phase-Domain Model for Type-I *PLLs*.

The open-loop transfer function is expressed as

$$G(s) = K_{PD}\left(\frac{1}{R_1C_1s+1}\right)\frac{K_{VCO}}{s}. \qquad \text{Equation 1.3}$$

This results in having the closed-loop transfer function of

$$\frac{\Phi_{OUT}}{\Phi_{IN}} = \frac{G(s)}{1+G(s)} = \frac{K_{PD}K_{VCO}}{R_1C_1s^2+s+K_{PD}K_{VCO}}. \qquad \text{Equation 1.4}$$

As expressed in Equation 1.3, the open-loop transfer function has a single pole at the origin which is caused by the *VCO*. Consequently this *PLL* is called "Type-I *PLL*". Following our intuition, for s ≈ 0 (slow input variations) the output phase tracks the input one ($\Delta\Phi_{OUT} = \Delta\Phi_{IN}$) as expressed in Equation 1.4.

Although the output phase tracks slow variations in the input phase ($\Delta\Phi_{OUT} = \Delta\Phi_{IN}$), the static phase difference between the input and the output is not zero ($\Phi_{OUT} \neq \Phi_{IN}$). Equation 1.4 can also be applied to frequency quantities where the output frequency closely tracks slow frequency variations but fails to track faster ones.

By comparing Equation 1.4with the following typical second-order transfer function from control theory

$$\textbf{Closed Loop TF} = \frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}. \qquad \text{Equation 1.5}$$

We could easily obtain

$$\xi = \frac{1}{2}\sqrt{\frac{\omega_{LPF}}{K_{PD}K_{VCO}}} \qquad \text{Equation 1.6}$$

$$\omega_n = \sqrt{K_{PD}K_{VCO}\omega_{LPF}} \qquad \text{Equation 1.7}$$

where

$$\omega_{LPF} = \frac{1}{R_1C_1} \qquad \text{Equation 1.8}$$

such that ξ denotes the damping factor and ω_n is the natural frequency of the Type-II *PLL* system. To be provide a well-behaved overdamped (or critically damped) response, the damping factor (ξ) is chosen to be *1/√2* or larger.

Figure 1.9 illustrates the Bode plots of the open-loop transfer function. It is clear that as K_{VCO} increases (ξ decreases), the gain crossover frequency (unity-gain frequency) increases whereas the phase plot remains unchanged. As a result, the Phase Margin is reduced i.e. the system becomes less stable.

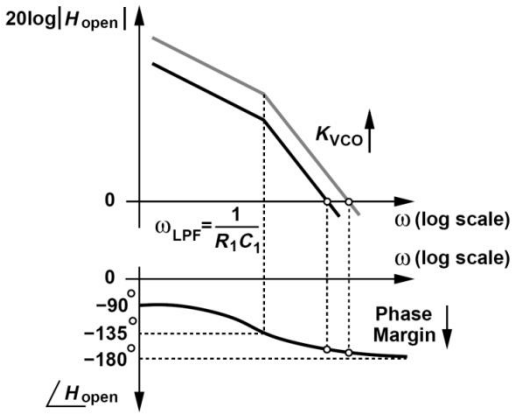


Figure 1.9: Bode plots of Type-I *PLLs* for different values of K_{VCO}.

1.1.2 Frequency Multiplication in Type-I *PLLs*:

As shown in section 1.5.3, frequency multiplication is one of the main applications of *PLLs*. Frequency multiplication is generating an oscillating waveform with frequency that is a multiple (integer or fraction) of a reference frequency. As shown in

Figure 1.9, this is achieved by dividing the output frequency of the *PLL* before feeding it back to the phase detector. The divide by modulus "*M*" circuit is implemented as explained in 1. In the steady state, the loop is locked and the feedback frequency (ω_F) is equal to the reference frequency (ω_{IN}). Consequently, the output frequency is "*M*" multiples of the input frequency ($\omega_{OUT} = M\ \omega_{IN}$).

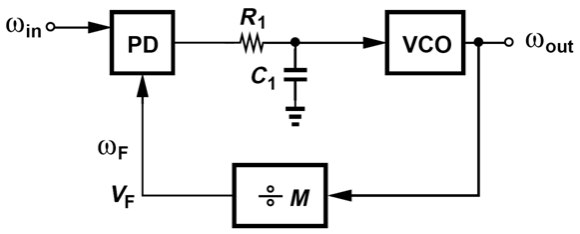


Figure 1.10: Frequency multiplication using a *PLL*

The *PLL* shown in Figure 1.10 can be used as a frequency synthesizer (as shown in section 1.5.3). This is achieved by changing the divider modulus (*M*). If *M* changes by one, the output frequency changes by ω_{IN}. Thus, the output frequency is varied with steps of ω_{IN} by changing the modulus of division and thus frequency synthesis is achieved.

The phase-domain model can be readily modified to include the effect of the modulus *M* as follows

$$\boldsymbol{Openloop\ TF} = \frac{\boldsymbol{G(s)}}{\boldsymbol{M}} = \frac{\boldsymbol{K_{PD}}}{\boldsymbol{M}}\left(\frac{\boldsymbol{1}}{\boldsymbol{R_1C_1s+1}}\right)\frac{\boldsymbol{K_{VCO}}}{\boldsymbol{s}} \qquad \text{Equation 1.9}$$

$$\frac{\boldsymbol{\Phi_{OUT}}}{\boldsymbol{\Phi_{IN}}} = \frac{\boldsymbol{G(s)}}{\boldsymbol{1+\frac{1}{M}G(s)}} = \frac{\boldsymbol{K_{PD}K_{VCO}}}{\boldsymbol{R_1C_1s^2+s+\frac{1}{M}K_{PD}K_{VCO}}} \qquad \text{Equation 1.10}$$

$$\boldsymbol{\xi = \frac{1}{2}\sqrt{\frac{M\omega_{LPF}}{K_{PD}K_{VCO}}}} \qquad \text{Equation 1.11}$$

$$\boldsymbol{\omega_n = \sqrt{\frac{K_{PD}K_{VCO}\omega_{LPF}}{M}}} \qquad \text{Equation 1.12}$$

where

$$\boldsymbol{\omega_{LPF} = \frac{1}{R_1C_1}} \qquad \text{Equation 1.13}$$

such that ξ designates the damping factor and ω_n is the natural frequency of the Type-II *PLL* system. As expressed in Equation 1.11, the frequency division by *M* results in a weaker feedback which leads to a slower response and a larger phase difference in the steady state.

1.1.3 Drawbacks of Type-I *PLLs*:

Type-I *PLLs* have three major drawbacks. First, as expressed in Equation 1.11, there is a tight relation between the damping factor (ξ) which affects the loop stability and the corner frequency of the *LPF* (ω_{LPF}). This is proved to be an undesirable characteristic. For example, to suppress the ripples on the control line (V_{CONT}) the *LPF* is required to have a small corner frequency (ω_{LPF}) whereas a small ω_{LPF} results in a less stable *PLL*. This tradeoff is eliminated in Type-II *PLLs*.

The second drawback is having a limited lock range or "acquisition range". For example, if the initial frequency of the *VCO* is very far from the reference frequency, the loop may never achieve lock. This is mainly caused by the Phase Detector (*PD*) which provides very little information when the input and output have different frequencies. This is also eliminated by Type-II *PLLs*.

The third drawback is having a finite phase difference between the output and the input in the steady state. This phase difference can be eliminated by having an infinite loop gain as will be seen in Type-II *PLLs*.

1.2 Type-II *PLLs*:

Type-II *PLLs* deal with all the drawbacks of Type-I *PLLs* described in 1.1.3. Type-II *PLLs* widen the acquisition range by adding a Frequency Detector to the loop of the *PLL*. A frequency detector by itself will not be enough (due to the residual frequency difference between the input and the output in the steady state as shown in Figure 1.5). The loop must also lock the phases. Consequently, it is required to have a circuit that operates as a Frequency Detector when the input and output have different frequencies, and work as a Phase Detector at equal input frequencies to further lock the phases. Such a circuit is called a Phase/Frequency Detector (*PFD*) and is implemented as shown in 0.

When a *PFD* is employed in a *PLL*, the problem of having a limited acquisition range is eliminated completely. As shown in Figure 1.11, the *LPF* extracts the *DC* content of each of Q_{Ab} and Q_B. Then, the differential amplifier (A_1) extracts the difference between (Q_A - Q_B) them and amplifies it. The result is then passed to the *VCO* as a control voltage (V_{CONT}).

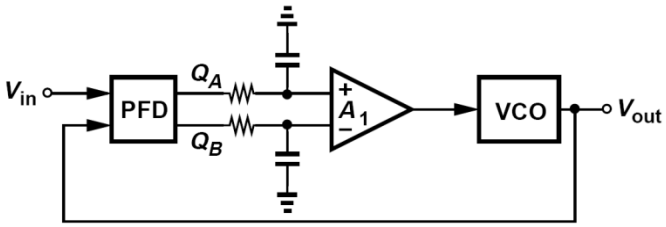


Figure 1.11: Introduction of a *PFD* to a Type-I *PLL*.

At startup, the *PFD* acts as a Frequency Detector and brings the output frequency towards that of the input. Once they are close enough to each other, the *PFD* acts as a Phase Detector and locks the output phase to that of the input (acquires phase-lock).

On the other hand, type-II *PLLs* solve the problem of having a tight relation between the damping factor and the corner frequency of the *LPF* by employing a Charge Pump (*CP*). The charge pump is directly implemented as explained in 1. The *PFD/CP/Capacitor* cascade (shown in Figure 1.12) has an infinite gain i.e. it produces a ramp-like output voltage (V_{CONT}) in response to a constant phase difference between the output and the input (behaves like an integrator).

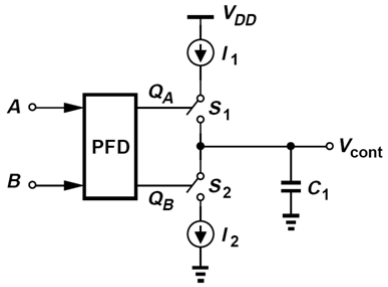


Figure 1.12: Cascade of a PFD, CP and a capacitor.

1.2.1 Phase-Domain Model for Type-II PLLs:

Figure 1.13 illustrates an attempt for a Charge-pump *PLL*. This loop has a zero phase difference between the output and the input in the steady state because a finite phase difference leads to an unbounded value for the control voltage (V_{CONT}) (Infinite gain).

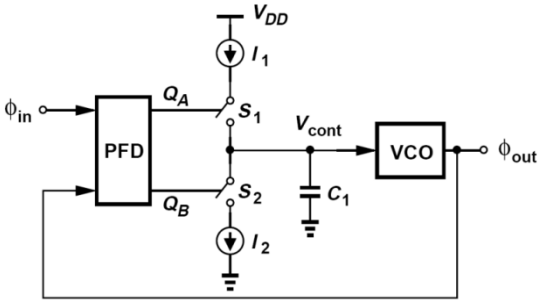


Figure 1.13: An attempt to design a Charge-Pump *PLL*

Equation 1.14 describes the transfer function of the *PFD/CP/Capacitor* cascade (shown in Figure 1.12) as obtained in **Error! Reference source not found.** which confirms that the cascade works as an integrator, where $\Delta\Phi_{PFD}$ is the phase difference between the two inputs of the *PFD*.

$$\frac{V_{CONT}}{\Delta\Phi_{PFD}} = G_1(s) = \frac{I_P}{2\pi C_1}\frac{1}{s}. \qquad \text{Equation 1.14}$$

Using Equation 1.14, the closed-loop transfer function for the *PLL* shown in Figure 1.13 can be expressed as

$$\frac{\Phi_{OUT}}{\Phi_{IN}} = \frac{G_1(s)\frac{K_{VCO}}{s}}{1 + G_1(s)\frac{K_{VCO}}{s}} = \frac{I_P K_{VCO}}{2\pi C_1 s^2 + I_P K_{VCO}}. \qquad \text{Equation 1.15}$$

This design is called a Type-II *PLL* because its open-loop transfer function has two poles at the origin which means that it has two ideal integrators.

There is a major drawback with the *PLL* developed in Figure 1.13which is revealed by Equation 1.15. The system has two pure imaginary poles on the $j\omega$ axis. This indicates that the system is oscillatory (unstable system because we have two lossless integrators in a loop).

This problem can be simply solved by making one of the integrators lossy. One way to achieve this is by inserting a resistance in series with the capacitor (C_1). This way the system is stabilized and the resulting circuit is called a Charge-Pump *PLL* (*CPPLL*). A typical *CPPLL* is illustrated in Figure 1.14.

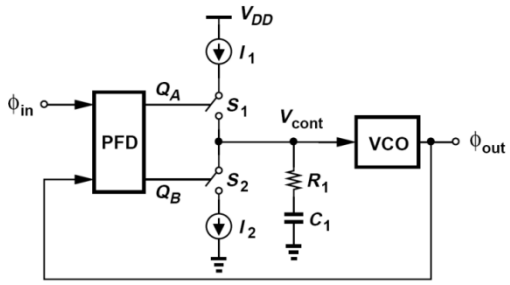


Figure 1.14: A typical Charge-Pump *PLL*.

The *PFD/CP/LPF* cascade has a transfer function given by

$$\frac{V_{CONT}}{\Delta\Phi} = G_1'(s) = \frac{I_P}{2\pi}\left(\frac{1}{C_1 s} + R_1\right) \qquad \text{Equation 1.16}$$

where $\Delta\Phi$ is the phase difference between the two inputs of the *PFD*.

Using Equation 1.16, the closed-loop transfer function of the *PLL* shown in Figure 1.14 can be obtained as

$$\frac{\Phi_{OUT}}{\Phi_{IN}} = \frac{G_1'(s)\frac{K_{VCO}}{s}}{1 + G_1'(s)\frac{K_{VCO}}{s}} = \frac{\frac{I_P K_{VCO}}{2\pi C_1}(R_1 C_1 s + 1)}{s^2 + \frac{I_P}{2\pi}K_{VCO}R_1 s + \frac{I_P}{2\pi C_1}K_{VCO}}. \qquad \text{Equation 1.17}$$

By comparing the denominator of Equation 1.17 with the typical second order characteristic equation (Denominator of the transfer function expressed in Equation 1.5) we can readily obtain

$$\xi = \frac{R_1}{2}\sqrt{\frac{I_p C_1 K_{VCO}}{2\pi}} \qquad \text{Equation 1.18}$$

$$\omega_n = \sqrt{\frac{I_P K_{VCO}}{2\pi C_1}} \qquad \text{Equation 1.19}$$

where ξ denotes the damping factor and ω_n is the natural frequency of the Type-II *PLL* system.

It is clear from Equation 1.18 that as C_1increases (to decrease the corner frequency of the *LPF* and suppress the ripples on the control voltage) the damping factor (ξ) also decreases. This trend is exactly the opposite of that found in Type-I *PLLs* and expressed by Equation 1.6. Thus, we have removed the previous tradeoff between stability (ξ) and ripple suppression (C_1).

The closed-loop poles are now complex (not pure imaginary anymore) and can be expressed as

$$\omega_{P_{1,2}} = \left(1 - \xi \pm \sqrt{\xi^2 - 1}\right)\omega_n \qquad \text{Equation 1.20}$$

There is also a closed-loop zero which is found at

$$\omega_Z = -\frac{\omega_n}{2\xi} \qquad \text{Equation 1.21}$$

The open-loop transfer function provides a very useful insight on frequency compensation (stabilization). It is simply obtained as

$$G_1'(s)\frac{K_{VCO}}{s} = \frac{I_P K_{VCO}}{2\pi}\left(\frac{R_1 C_1 s + 1}{C_1 s^2}\right) \qquad \text{Equation 1.22}$$

It has a real left-half-plane open-loop zero at $-1/(R_1 C_1)$ which has been added to the open-loop transfer function for frequency compensation (to stabilize the *PLL*). Figure 1.15 illustrates the Bode plots of the system before compensation (Figure 1.13) and after compensation (Figure 1.14).

It is clear that the system before compensation has no phase margin (unstable). As for after compensation, the phase curve increases due to the added zero and the magnitude curve decreases at a smaller rate which leads to a high phase margin.

It is also noted that as K_{VCO} increases the magnitude curve is shifted upwards and the phase curve remains unchanged. This moves the gain cross-over frequency to the right and increases the phase margin. This behavior which follows Equation 1.18 is directly opposite to the behavior experienced by the Type-I *PLLs* (expressed by Equation 1.6).

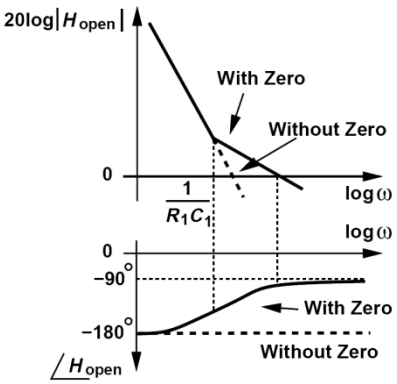


Figure 1.15: Bode Plots of the open-loop transfer function of a typical Charge-Pump *PLL* before and after compensation.

It is important to state that the addition of the resistance (R_1) creates significant ripple effects ($\Delta V_{CONT} = I_P R_1$) and causes large jumps on the control line even in the locked state.

1.2.2 Phase Margin of Type-II *PLLs*:

The expression for the Phase Margin of the second-order Type-II *PLL* expressed in Figure 1.14 can be derived from its open-loop transfer function expressed in Equation 1.22. The Bode plots of the open-loop transfer function follow the behavior presented in Figure 1.16.

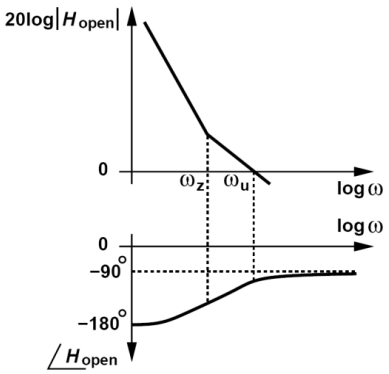


Figure 1.16: Bode Plots of the open-loop transfer function of a second-order Type-II *PLL*.

To determine the Phase Margin, we must first determine the gain cross-over frequency (unity-gain frequency ω_u). It is computed by equating the magnitude of Equation 1.22 to one and determining the corresponding value of the frequency ($s = j\omega_u$) as follows

$$\left|\frac{I_P}{2\pi}\left(R_1+\frac{1}{sC_1}\right)\frac{K_{VCO}}{s}\right|^2_{s=j\omega_u} = 1\,. \qquad \text{Equation 1.23}$$

After some careful mathematical manipulations and using ($R_1C_1\omega_n = 2\xi$), the following expression is obtained for the gain cross-over frequency (ω_u)

$$\omega_u = \sqrt{2\xi^2+\sqrt{4\xi^4+1}}\,\omega_n\,. \qquad \text{Equation 1.24}$$

Using Equation 1.24, the Phase Margin can be readily obtained as

$$PM = \tan^{-1}\frac{\omega_u}{\omega_Z} = \tan^{-1} R_1C_1\omega_u = \tan^{-1}\left(2\xi\sqrt{2\xi^2+\sqrt{4\xi^4+1}}\right). \qquad \text{Equation 1.25}$$

Equation 1.26 results in a *PM* of 76^O at $\xi = 1$ and a *PM* of 65^O at $\xi = 1/\sqrt{2}$. The values of the *VCO* gain (K_{VCO}), the *CP* current and the *LPF* are chosen to achieve a certain Phase Margin (ξ) which guarantees a high level of loop stability.

The left of Figure 1.17 illustrates the effect of increasing R_1 on the magnitude and phase plots of the open-loop transfer function. It is clear that increasing R_1 leads to decreasing the position of the zero (ω_Z) and increasing the gain cross-over frequency (ω_u) which increases the *PM* and stability considerably. On the other hand, increasing C_1 (right part of Figure 1.17) leads to decreasing the position of the zero (ω_Z) but nearly nothing happens to the gain cross-over frequency. This leads to a small increase in the PM. This means that the PM (ξ) has a stronger dependence on R_1 than on C_1.

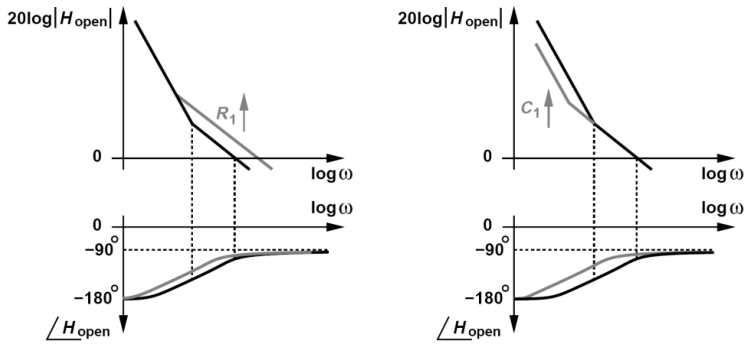


Figure 1.17: Effect of increasing R_1(on the left) and increasing C_1(on the right) on the Bode Plots.

1.2.3 Transient Response of Type-II *PLLs*:

The closed-loop transfer function of Type-II *PLLs* (Equation 1.17) may be used to determine the transient response of the system. This is achieved by assuming that the time constant of the loop is calculated as

$$\boldsymbol{Time\ Constant} = \frac{\mathbf{1}}{\boldsymbol{\xi\omega_n}} = \frac{\mathbf{4\pi}}{\boldsymbol{R_1 I_P K_{VCO}}}. \qquad \text{Equation 1.26}$$

This quantity is used extensively to measure the settling speed of the loop (especially if ξ is close to one). In general, for different values of ξ, the time constant lies between $1/(\xi\omega_n)$ and $1/(2\xi\omega_n)$.

1.2.4 Limitations of Continuous-Time Approximation:

By design, Charge-Pump *PLLs* are discrete-time (*DT*) systems because the *CP* is turned off for a fraction of the period and the loop is broken. However, in our analysis we approximated this *DT* response with a continuous time (*CT*) one to simplify the analysis without losing any of the properties of the system. As a result, the analysis done so far is true only when the control voltage (V_{CONT}) and the phase of the VCO do not change significantly from one input cycle to the next i.e. the Time constant obtained in Equation 1.26 should be much larger than the input (reference) period. In practice, loops with a time constant smaller than the input period experience an underdamped behavior and in some cases never acquire lock (and the *CT* approximation is no longer valid).

1.2.5 Frequency Multiplication in Type-II *PLLs*:

As shown in section 1.5.3, frequency multiplication is one of the main applications of *PLLs*. Frequency multiplication is generating an oscillating waveform with frequency that is a multiple (integer or fraction) of a reference frequency. Similar to Type-I *PLLs*, Type-II *PLLs* will multiply the reference frequency by a factor of "*M*" if a frequency divider with modulus "*M*" is placed in the feedback path of the loop. Figure 1.18 illustrates the frequency multiplication circuit for Type-II *PLLs*. The divide by modulus "*M*" circuit is implemented as explained in 1.

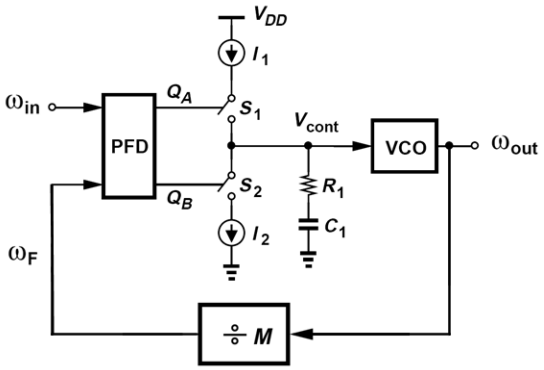


Figure 1.18: Frequency multiplication in Type-II *PLLs*.

In the steady state, the loop is locked and the feedback frequency (ω_F) is equal to the reference frequency (ω_{IN}). Consequently, the output frequency is "*M*" multiples of the input frequency ($\omega_{OUT} = M\,\omega_{IN}$). Thus, frequency multiplication is successfully achieved.

It is clear that the *PLL* shown in Figure 1.18 can also be used as a frequency synthesizer (as shown in section 1.5.3). This is achieved by changing the divider modulus (*M*). If *M* changes by one, the output frequency changes by ω_{IN}. Thus, the output frequency is varied with steps of $\omega_{IN}$by changing the modulus of division and thus frequency synthesis is achieved.

The phase-domain model can be readily modified to include the effect of the modulus *M* as follows

$$\frac{\Phi_{OUT}}{\Phi_{IN}} = \frac{\frac{I_P K_{VCO}}{2\pi C_1}(R_1 C_1 s + 1)}{s^2 + \frac{I_P}{2\pi}\frac{K_{VCO}}{M}R_1 s + \frac{I_P}{2\pi C_1}\frac{K_{VCO}}{M}} \qquad \text{Equation 1.27}$$

$$\xi = \frac{R_1}{2}\sqrt{\frac{I_p C_1}{2\pi}\frac{K_{VCO}}{M}} \qquad \text{Equation 1.28}$$

$$\omega_n = \sqrt{\frac{I_P}{2\pi C_1}\frac{K_{VCO}}{M}} \qquad \text{Equation 1.29}$$

where ξ denotes the damping factor and ω_n is the natural frequency of the Type-II *PLL* system.

As expressed in Equation 1.18, the frequency division of K_{VCO} by *M* results in a weaker feedback path which leads to a slower response and makes the loop less stable. In order to counteract the effect of *M*, it is required to increase I_P or C_1.

The performance of Type-II *PLLs* can be enhanced further by increasing the order of the loop filter. The loop filter expressed in the left part of Figure 1.19 (R_1 *and* C_1) is considered to be inefficient and does not suppress the ripples enough even during the lock state. This is evident by the fact that there must be an instantaneous jump in the control voltage ($\Delta V_{CONT} = I_P R_1$) every time the *PFD* produces a pulse. These ripples are even present in the lock state because of small skews and propagation mismatches within the *PFD*. As shown in the right part of Figure 1.19, the ripples have positive and negative pulses occurring at a constant rate leading to significant modulation of the output of the *VCO*. Hence, additional suppression of

these ripples is extremely necessary. In the following section, we use a second order loop filter to further suppress these ripples.

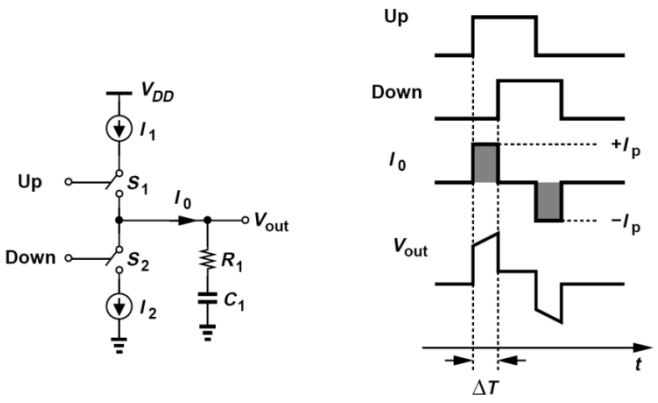


Figure 1.19: A *CP/LPF* cascade (on the left) and the effect of skews between the Up and Down reset pulses on the control voltage (on the right).

1.3 Higher Order Loops:

In this section, we introduce a second order loop filter to the *PLL*, to further suppress the ripples on the control voltage. The left part of Figure 1.20 illustrates a typical approach to further suppress the ripples on the control voltage by adding a second capacitor which increases the order of the *LPF* (becomes a second-order *LPF*). This provides a new low-impedance path to the ground which absorbs the output of the Charge-Pump. When a current pulse of width ΔT is produced by the *CP*, it charges C_2 (approximated as a ramp in the right part of Figure 1.20). When the current pulse is over, C_2 starts discharging and loses some of its stored charge to C_1. This causes an exponential decay in the control voltage. It is required to have C_2as large as possible to extremely suppress the ripples on V_{CONT}.

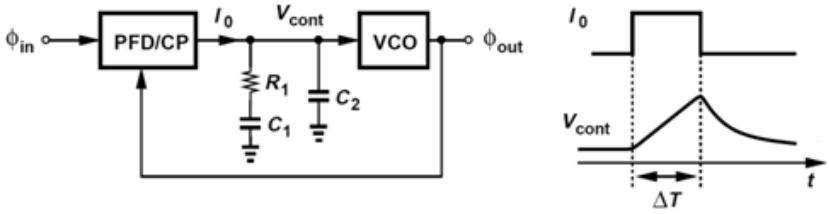


Figure 1.20: A Third-Order *PLL* employing an additional capacitor (on the left) and its effect on suppression of the ripples on V_{CONT} (on the right).

1.3.1 Phase-Domain Model for Third-Order Type-II P*LLs*:

The transfer function of the new cascade *PFD/CP/LPF* expressed in the left part of Figure 1.20 can be readily obtained by modifying Equation 1.16 to include the effect of the second-order *LPF* as follows

$$\frac{V_{CONT}}{\Delta\Phi} = G_1''(s) = \frac{I_P}{2\pi}\left(\left(\frac{1}{C_1 s} + R_1\right) // \frac{1}{C_2 s}\right). \qquad \text{Equation 1.30}$$

The open-loop transfer function of the third-order *PLL* expressed in the left part of Figure 1.20 can be easily derived using Equation 1.30 as follows

$$\boldsymbol{Open-loop\ TF} = \frac{\mathbf{1}}{\boldsymbol{s^2}} \frac{\boldsymbol{I_P K_{VCO}}}{\boldsymbol{2\pi}} \frac{\boldsymbol{(1+R_1 C_1 s)}}{\boldsymbol{(1+R_1 C_{EQ} s)}} \frac{\mathbf{1}}{\boldsymbol{(C_1+C_2)}} \qquad \text{Equation 1.31}$$

where $C_{EQ} = C_1C_2/(C_1+C_2)$.

It is observed that the open-loop transfer function has an additional pole at $1/(R_1C_{EQ})$ which degrades the stability of the loop and decreases the phase margin. Therefore, due care has to be given to the choice of C_2in order to maximize the phase margin.

1.3.2 Phase Margin of Third-Order Type-II *PLLs*:

The expression for the Phase Margin of the third-order Type-II *PLL* expressed in the left part of Figure 1.20 can be derived from its open-loop transfer function expressed in Equation 1.31. The magnitude and phase plots of the open-loop transfer function follow the behavior presented in Figure 1.21 for ($\omega_u > \omega_{P2}$) on the left and ($\omega_{P2} > \omega_u$) on the right. There is an obvious degradation in the Phase Margin and stability caused by the addition of C_2which caused an additional pole ω_{P2} ($\omega_{P2} = 1/(R_1C_{EQ})$).

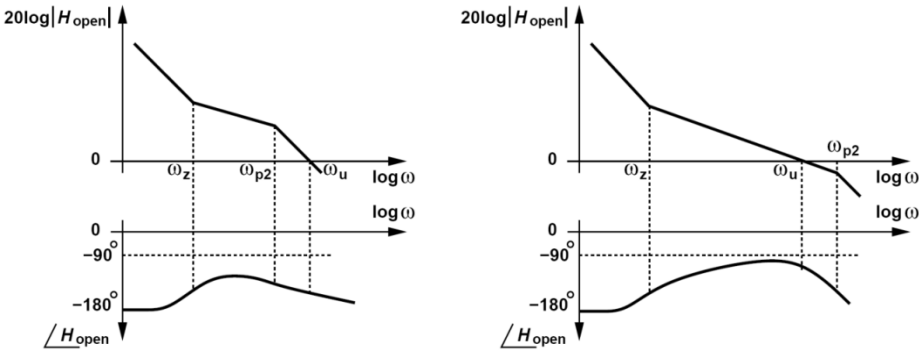


Figure 1.21: Bode Plots of the open-loop transfer function of a third-order Type-II *PLL* where ω_U is the gain cross-over frequency for ($\omega_u > \omega_{P2}$) on the left and ($\omega_{P2} > \omega_u$) on the right.

It is clear from Figure 1.21 that to achieve a higher Phase Margin, it is required to haveω_{P2}much higher than ω_u. In this case, the magnitude of the gain cross-over frequency can be approximated to be the same as in Equation 1.24.

After determining the value of the gain cross-over frequency, the Phase Margin can be computed as follows

$$PM = \tan^{-1}\frac{\omega_u}{\omega_Z} - \tan^{-1}\frac{\omega_u}{\omega_{P2}} = \tan^{-1} R_1C_1\omega_u - \tan^{-1} R_1C_{EQ}\omega_u\ . \qquad \text{Equation 1.32}$$

By substituting for ω_u by the value obtained in Equation 1.24 the *PM* can be simplified as

$$PM = \tan^{-1}\left[4\xi^2\left(1+\frac{1}{32\xi^2}\right)\right] - \tan^{-1}\left[4\xi^2\frac{C_{EQ}}{C_1}\left(1+\frac{1}{32\xi^2}\right)\right] \qquad \text{Equation 1.33}$$

This result is only valid for ω_{P2}much higher than ω_u.

The max Phase Margin can be obtained by differentiating Equation 1.32 with respect to ω_u which yields

$$PM_{MAX} = \tan^{-1}\left(\frac{C_1/C_2}{2\sqrt{1+C_1/C_2}}\right) \qquad \text{Equation 1.34}$$

at

$$\omega_u = \frac{1}{R_1 C_1}\sqrt{1+\frac{C_1}{C_2}}. \qquad \text{Equation 1.35}$$

The corresponding value of ξ can also be obtained (by differentiating Equation 1.33) to obtain

$$\xi = \frac{1}{2}\sqrt[4]{\frac{C_1}{C_{EQ}}}. \qquad \text{Equation 1.36}$$

As shown in Figure 1.22, it is clear that there is a limitation on the value of R_1 used in this design. This is caused by the fact that as R_1 increases,ω_{P2} $(1/(R_1C_{EQ}))$ decreases and may even become smaller than ω_u (ω_u is expressed in Equation 1.24).

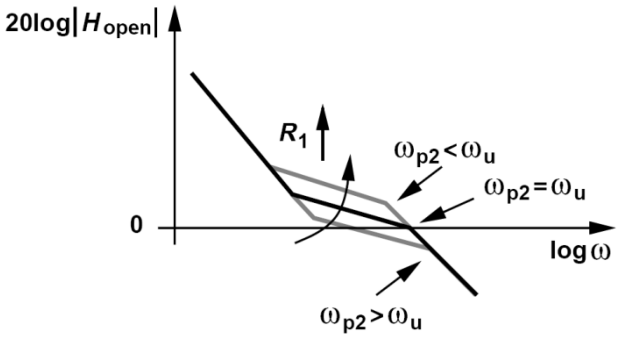


Figure 1.22: Effect of increasing R_1on the magnitude plot of the open-loop transfer function of a Third-Order Type-II *PLL*.

Also, as R_1increases, the combination of $R_1$and C_1vanishes in comparison with C_2 and the loop falls back to a Type-I *PLL*(shown in Figure 1.8) which is inherently unstable (two lossless integrators in a loop causing oscillations). Therefore, it is required to determine the upper limit on R_1which is reached when $\omega_{P2} \approx \omega_u$. This upper limit can simple be calculated by setting $\omega_{P2} > \omega_u$ as follows

$$\frac{1}{R_1 C_{EQ}} \geq 2\xi\,\omega_n \geq \frac{R_1 I_P K_{VCO}}{2\pi}. \qquad \text{Equation 1.37}$$

After some mathematical manipulation, the upper limit on R_1can be obtained as

$$R_1^2 \leq \frac{2\pi}{I_P K_{VCO} C_{EQ}} \qquad \text{Equation 1.38}$$

And

$$\frac{C_2}{C_1 + C_2} \leq \frac{1}{4\xi^2}. \qquad \text{Equation 1.39}$$

The above lower bound on ω_{P2}can be achieved if $\xi = 0.8$ *to* 1 and $C_2 \approx 0.2C_1$.

After satisfying the above condition, the values of the *VCO* gain (K_{VCO}), the *CP* current and the *LPF* are chosen to achieve a certain Phase Margin (ξ) which guarantees a high level of loop stability.

As for the closed-loop transfer of the third-order *PLL*, it can be readily obtained using Equation 1.30 as

$$\frac{\Phi_{OUT}}{\Phi_{IN}} = \frac{G_1''(s)\frac{K_{VCO}}{s}}{1 + G_1''(s)\frac{K_{VCO}}{s}} = = \frac{\frac{I_P K_{VCO}}{2\pi}(R_1 C_1 s + 1)}{s^3 R_1 C_1 C_2 + (C_1 + C_2)s^2 + \frac{I_P}{2\pi} K_{VCO} R_1 C_1 s + \frac{I_P}{2\pi} K_{VCO}} \qquad \text{Equation 1.40}$$

1.3.3 An Alternative Second-Order Loop Filter:

Figure 1.23 suggests an alternative second-order *LPF* to suppress the ripples on the control line (V_{CONT}). Although the ripple at "*X*" is large as before, the *LPF* made up of $R_2$and C_2suppresses it before presenting it to the *VCO*.

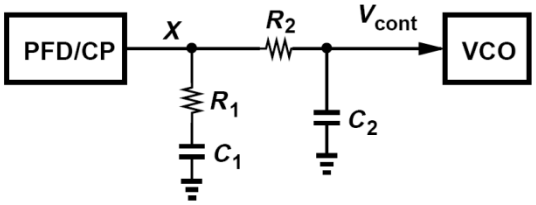


Figure 1.23: An Alternative second-order *LPF* to suppress the ripples on the control voltage.

The Phase Margin of the open-loop transfer function for the *PLL* in Figure 1.23 can be obtained as

$$PM \approx \tan^{-1}(4\xi^2) - \tan^{-1}\left(4\xi^2 \frac{R_2 C_2}{R_1 C_1}\right) \qquad \text{Equation 1.41}$$

where the additional pole is at $1/(R_2C_2)$.

It is important to state that in order to achieve a reasonable *PM*, the additional pole ($1/(R_2C_2)$) must be from *5* to *10* times larger than the zero (ω_z).

Table 1.2 provides a brief summary of the differences between the different types of *PLLs* discussed throughout this chapter. In the following section, we discuss the design procedures followed to design the Loop-Filter employed in the PLL of the TV Tuner System.

Table 1.2: A brief comparison between the different Types of *PLLs*.

	Type-I *PLLs*	**Type-II *PLLs* (2nd Order)**	**Type-II *PLLs* (3rd Order)**
Phase Margin	—	$\tan^{-1}\left(2\xi\sqrt{2\xi^2+\sqrt{4\xi^4+1}}\right)$	$PM=\tan^{-1}\left[4\xi^2\left(1+\frac{1}{32\xi^2}\right)\right]-\tan^{-1}\left[4\xi^2\frac{C_{EQ}}{C_1}\left(1+\frac{1}{32\xi^2}\right)\right]$
ξ	$\frac{1}{2}\sqrt{\frac{M}{R_1C_1K_{PD}K_{VCO}}}$	$\frac{R_1}{2}\sqrt{\frac{I_PC_1}{2\pi}\frac{K_{VCO}}{M}}$	$\frac{R_1}{2}\sqrt{\frac{I_PC_1}{2\pi}\frac{K_{VCO}}{M}}$
ω_n	$\sqrt{\frac{K_{PD}K_{VCO}}{R_1C_1M}}$	$\sqrt{\frac{I_P}{2\pi C_1}\frac{K_{VCO}}{M}}$	$\sqrt{\frac{I_P}{2\pi C_1}\frac{K_{VCO}}{M}}$
Φ_{OUT}/Φ_{IN}	$\frac{K_{PD}K_{VCO}}{R_1C_1s^2+s+\frac{1}{M}K_{PD}K_{VCO}}$	$\frac{\frac{I_PK_{VCO}}{2\pi C_1}(R_1C_1s+1)}{s^2+\frac{I_P}{2\pi}\frac{K_{VCO}}{M}R_1s+\frac{I_P}{2\pi C_1}\frac{K_{VCO}}{M}}$	$\frac{\frac{I_PK_{VCO}}{2\pi}(R_1C_1s+1)}{s^3R_1C_1C_2+(C_1+C_2)s^2+\frac{I_P}{2\pi}\frac{K_{VCO}}{M}R_1C_1s+\frac{I_P}{2\pi C_1}\frac{K_{VCO}}{M}}$
Drawbacks	Tight relation between ξ and ω_{LPF}. Limited Lock (Acquisition) Range. Finite phase difference between the output and the input in the steady state.	The first order loop filter used does not suppress the ripples on the control voltage enough during lock state. R_1 creates significant ripple effects and causes instantaneous jumps in the control voltage. Significant Modulation of the output of the VCO	There is a limit on the max value of R_1 $R_1 \leq \frac{2\pi}{I_PK_{VCO}C_{EQ}}$ And C_1 and C_2 $\frac{C_2}{C_1+C_2} \leq \frac{1}{4\xi^2}$

1.4 Design Procedures

As evident in Table 1.2, which compares between the different types of *PLLs*, we conclude that Type-II *PLL* (*3rd* order) has the highest level of performance as it does not suffer from the instantaneous jumps on the control line as the *2nd* order Type-II *PLL*. It also solves the problems of Type-I *PLLs* (finite phase difference in the steady state and limited acquisition range). Consequently we have decided to employ a *3rd* order Type-II *PLL* in our design which suppresses the ripples on the control line to a great extent. In the following, we will design a *3rd* order Type-II *PLL* for out *TV* Tuner Receiver.

To design the *2nd* order loop filter used, we start with the following equations:

$$\xi = \frac{R_1}{2}\sqrt{\frac{I_p K_{vco}\, C_1}{2\pi M}} \qquad \text{Equation 1.42}$$

$$\omega_n = \sqrt{\frac{I_p K_{vco}}{2\pi C_1 M}} \qquad \text{Equation 1.43}$$

Where: M = the largest modulus frequency division.
K_{vco} =sensitivity of the *VCO*.

To ensure a high stability of the *PLL* with an appropriate settling time we set $\xi = 0.9$ and $\omega_n = \frac{\omega_{in}}{25}$, where ω_{in} is the reference frequency used in our design ($\omega_{in} = 2\pi \times 6MHz$ as demonstrated in 1)

As for C_2, we set C_2 equals to $0.25C_1$ to maximize the Phase Margin of the *PLL* as seen in section 1.3.2.

We also have to make sure that the obtained design of the loop filter satisfies the following constraints:

$$R_1{}^2 \leq \frac{2\pi}{I_p K_{vco} C_{eq}} \qquad \text{Equation 1.44}$$

$$\frac{C_2}{C_2 + C_1} \leq \frac{1}{4\xi^2} \qquad \text{Equation 1.45}$$

From our implementation of the *PLL* Building Blocks discussed in the following chapters, we have $K_{VCO} = 5.051GHz/V$, $M=140$(worst case) and $f_{in} = 6MHz$; by substituting in Equation 1.42 and Equation 1.43, the loop filter parameters can be directly obtained as $R_1 = 95\ \Omega$, $C_1 = 12.7\ nF$ and $C_2 = 3.175nF$. It is clear that these values satisfy the constraints presented in Equation 1.44 and Equation 1.45.

The phase margin can be directly calculated from Equation 1.33 as $PM = 39.5°$ which ensures the stability of the system. Figure 1.24 presents the Bode Plots of the open-loop transfer function of our designed *PLL*. The simulation confirms that the phase margin is indeed 40^O.

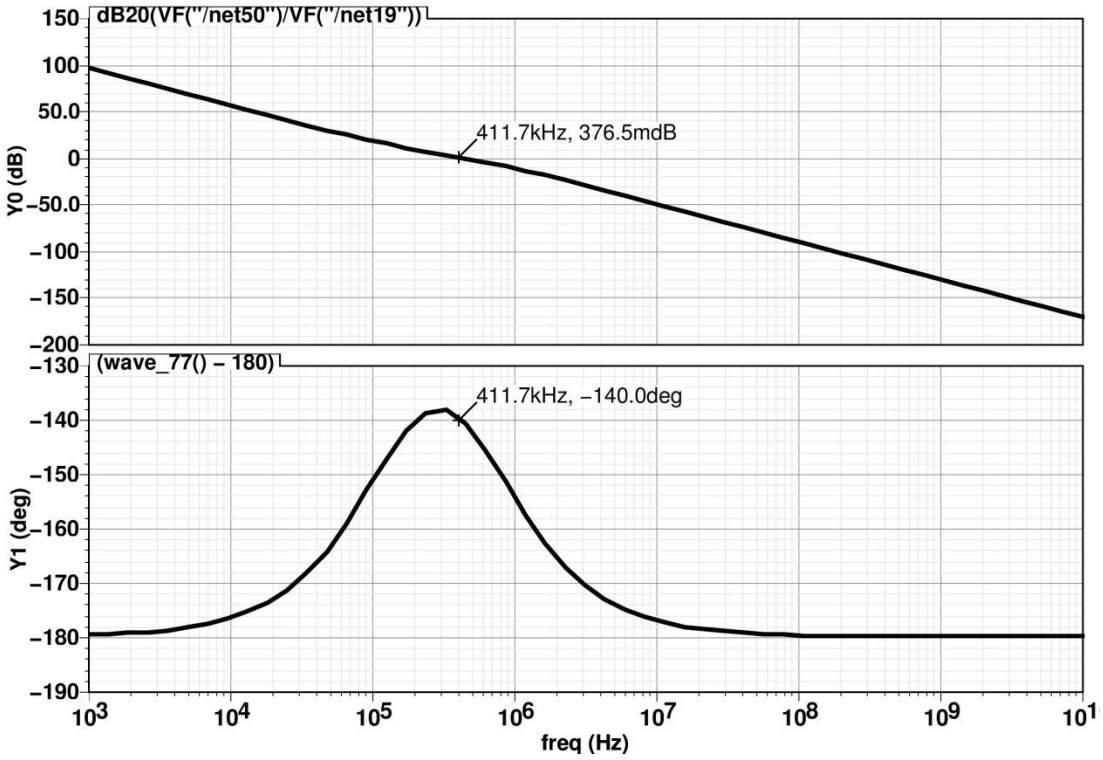


Figure 1.24 system's gain and phase

To conclude, the obtained values of phase margin and ξ ensure the stability of the system under its largest divide ratio (worst case). These values also ensure small settling and response times which guarantees a robust performance of the *PLL* throughout the tuning range of frequency operation. In the following section, we will briefly introduce some of the typical applications of the *PLL*.

1.5 Applications of *PLLs*:

PLLs are required in all *RF* communication systems as they offer a wide variety of applications. In this section, we will briefly mention these applications as follows:

1.5.1 Demodulation:

The *PLL* can serve as a Frequency-Shift Keying (*FSK*) demodulator by using the required signal to be demodulated as the reference signal and the control voltage (V_{CONT}) as an output. As shown in Figure 1.25, when the input (reference) frequency toggles between two different levels, the loop closely follows this change. As a result, the output frequency also toggles between these two levels and so does the control voltage. In this case, the control voltage represents the demodulated bit stream as required.

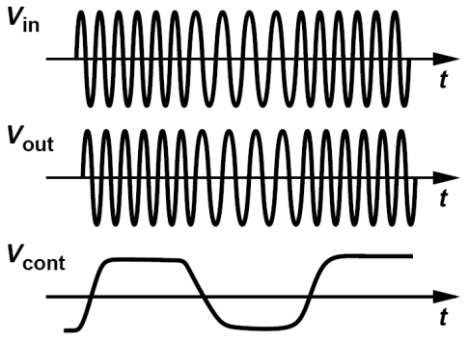


Figure 1.25: *PLL* used as an *FSK* demodulator.

1.5.2 Low-Pass Filtering:

The dynamics of the *PLL* has some interesting properties which enables it to be used as a Low-Pass Filter (*LPF*). If the input (reference) frequency changes at a very fast rate such that the *PLL* never settles, the output frequency will change at a much slower rate than the input frequency. As shown in Figure 1.26, the control voltage never settles and in essence the *PLL* acts as a Low-Pass Filter. Many applications use *PLLs* to reduce the change in frequency of a signal (or phase noise) by employing this low-pass filtering effect.

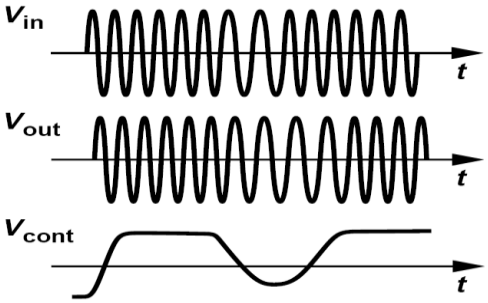


Figure 1.26: *PLL* used as a *LPF*

1.5.3 Frequency Multiplication and Synthesis:

Frequency dividers are used in PLL, mainly, for frequency multiplication. VCOs can typically operate at wide range of frequencies according to its input control voltage level. On the other hand its output frequency is fed back to the PFD to be compared with the crystal oscillator constant frequency. Thus, if the VCO output frequency is fed back directly to the PFD as shown in Figure 1.27, the loop will force the VCO to oscillate at the same frequency of the crystal oscillator in order to settle at zero frequency error. This means that this loop has no advantage over using the crystal oscillator alone.

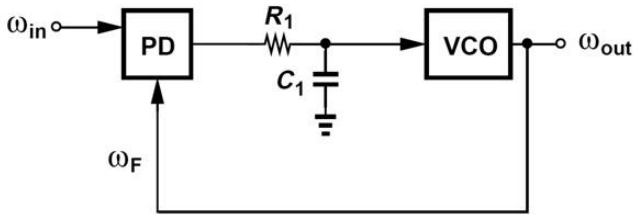


Figure 1.27: PLL without Frequency Divider

In order to breathe life to this loop, VCO frequency shouldn't be sent directly to the PFD. Instead, it is divided by a certain number *M*, using a frequency divider, and the division resulting frequency ω_{out}/M is the one sent to the PFD to be compared with the input reference frequency. In this case the loop forces the VCO to oscillate at a frequency such that the error is zero, i.e., $\omega_{out}/M = \omega_{in}$, hence:

$$\omega_{out} = M\omega_{in} \qquad \text{Equation 1.46}$$

M is called the divide ratio or the modulus.

This concept of frequency multiplication using frequency dividers in a feedback loop is very analogous to voltage amplification using op amps. If we consider the PLL loop without a frequency divider shown in Figure 1.27, it is similar to a buffer whose output voltage is compared directly to its input, so that they both are equivalent. On the other hand, the PLL loop with a frequency divider is analogous to a buffer whose output voltage is divided (attenuated) by a certain value ($1+R_1/R_2$) before returning back to the input, as shown in Figure 1.28, resulting in output that is amplified by the same ratio.

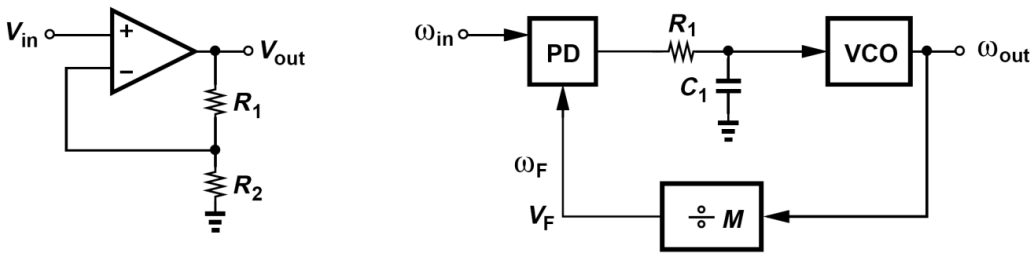


Figure 1.28: Analogy between Frequency Multiplication and Voltage Amplification

Until now, we obtained a PLL whose output frequency is a multiple of the input reference frequency but it is still unique constant value. In order to generate different output frequencies so that we make use of the VCO wide range advantage, the frequency divider is designed with a variable modulus, such that when *M* changes by 1, ω_{out} changes by ω_{in}. This variability of the frequency divider modulus allows the PLL to be used as a frequency synthesizer, which is our main target here. Like any other frequency synthesizer, this one has a group of valid selection inputs; each of them corresponds to a certain frequency output. Since the frequency divider is the block responsible for changing the output frequency value, therefore it is the block receiving the selection inputs.

2 Phase/Frequency Detectors:

Phase-Locked Loops generate a stable oscillating output with frequency that is a multiple of a reference frequency by achieving phase-lock between them. Consequently, there is a need in any PLL for a device that can compare the phases of two oscillating signals. In this chapter, we will start by explaining the concept of phase detectors and how they are implemented. Then, we will discuss Phase/Frequency detectors (PFDs). Finally, we will present a robust architecture that avoids the problem of dead zones faced by common PFDs.

2.1 Phase detector

As shown in Figure 2.1, a Phase Detector circuit simply has two periodic inputs and its output –as an average- is proportional to the phase difference between the two inputs. Ideally, the input/average-output relationship is monotonic and linear (a straight line). The slope of this straight line is called the *PD* "gain" and is denoted by "K_{PD}" (usually measured in *V/rad*). In general, the input/output in Figure 2.1characteristic is non-linear and in some cases even non-monotonic.

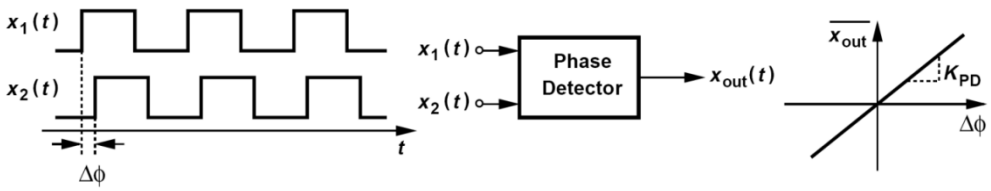


Figure 2.1: An Ideal Phase Detector and its input/output characteristic.

Figure 2.2 illustrates how the *PD* works by showing its output for two periodic inputs with unequal frequencies. It is clear that the phase difference between the inputs varies with time because $x_1$has a lower frequency than x_2. Thus, $x_2$accumulates faster resulting in a time-varying phase difference $\Delta\Phi$ ($\overline{x_{out}}$). It is clear that if $\Delta\Phi$ is equal to zero then $\overline{x_{out}}$ is also zero. This means –as will be shown later- that the *PLL* has achieved locking.

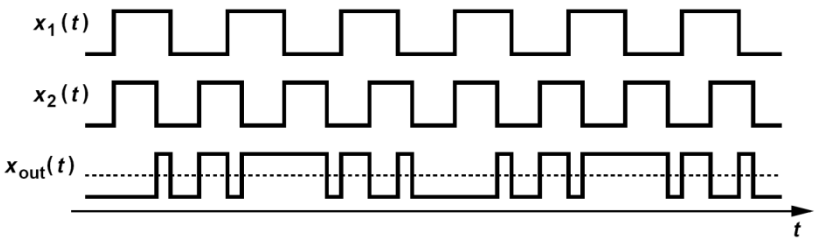


Figure 2.2: Output of A *PD* for two periodic inputs with unequal frequencies.

2.1.1 Implementation:

The output waveform shown in Figure 2.2 is identical to the output waveform for an exclusive-or (*XOR*) gate (shown in Figure 2.3). In other words, an *XOR* has an average output that is proportional

to the input phase difference and thus can be used as a *PD*. As shown in Figure 2.3, the *XOR* can generate pulses having a width equal to *ΔΦ* at both the rising and falling edges of the input waveforms.

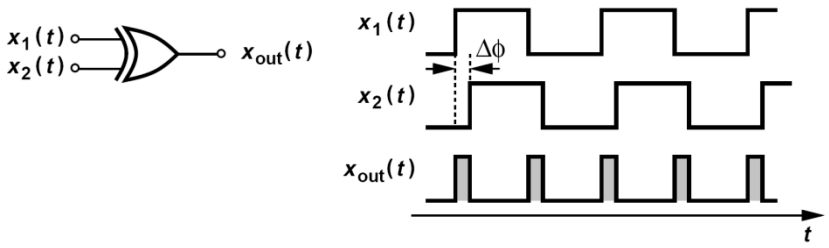


Figure 2.3: The usage of an *XOR* gate as a *PD.*

When the *XOR* gate is used as a *PD*, the input/output characteristic is not a straight line anymore. The characteristic is periodic as shown in Figure 2.4. The average value of the output ($\overline{x_{out}}$) is zero for *ΔΦ* equals to 0^O and it increases to V_{DD} at *ΔΦ* equals to 180^O. As *ΔΦ* exceeds 180^O, $\overline{x_{out}}$ starts decreasing again till it reaches zero at *ΔΦ* equals to 360^O and the characteristic repeats itself.

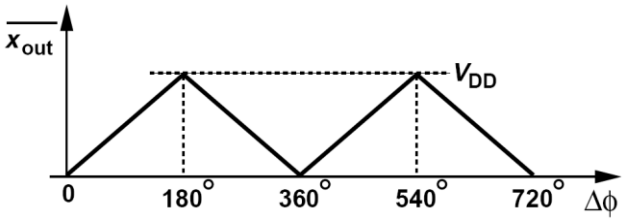


Figure 2.4: Input/output characteristic for an *XOR* gate used as a *PD*.

2.1.2 Drawbacks:

The Phase Detector (*PD*) provides very little information when the input and output have different frequencies (the frequencies of the two inputs signal must be equal). Also, when the PD is employed in the PLL, there is a finite phase difference between the output and the input in the steady state. This phase difference can be eliminated by having an infinite gain as will be seen in the next section. As shown in the previous chapter, the PLL with a phase detector only will not achieve locking so we will use a frequency detector circuit in addition to solve this problem.

2.2 Phase frequency detector

Phase frequency detector is a device that sense two different inputs signals, usually one is from the output of the divider and the other is from a reference crystal oscillator. IT has two outputs which instruct subsequent circuitry on how adjust to lock onto the phase.

Use of a PFD in a phase-locked loop resolves the issue of the limited acquisition range.
At the beginning of a transient, the PFD acts as a frequency detector, pushing the VCO frequency toward the input frequency. After the two are sufficiently close, the PFD operates as a phase detector, bringing the loop into phase lock.

2.2.1 Operation of the PFD:

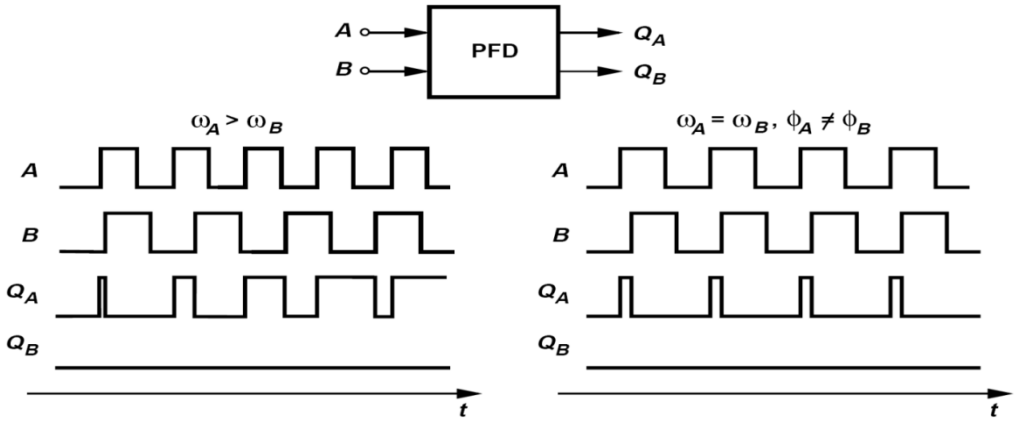


Figure 2.5 Response of PFD to inputs

Figure 2.5 conceptually shows the operation of a PFD. The circuit produces two outputs. Q_A & Q_B and operates based on the following conditions:

1. A rising edge on *A* yields a rising edge on Q_A (if Q_A is low).
2. A rising edge on *B* resets Q_A (if Q_A is high).

First $W_A > W_B$, then Q_A produces pulses while Q_B remains at zero. Conversely, if $W_B > W_A$ then positive pulses appear at Q_B and Q_A remains at zero.
Second If $W_A = W_B$ the circuit generates pulses at either Q_A or Q_B with a width equal to the phase difference between A and B.

2.2.2 State Diagram of PFD

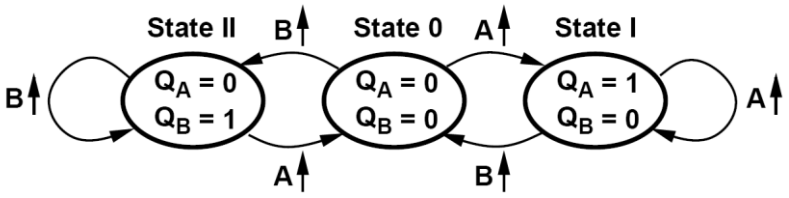


Figure 2.6 PFD state diagram

At least three logical states are necessary:

- Q_A=Q_B=0
- Q_A=0, Q_B=1
- Q_A=1, Q_B=0.

To avoid dependence of the output upon the duty cycle of the inputs, the circuit should be realized as an edge-triggered sequential machine.

2.3 PFD Basic design

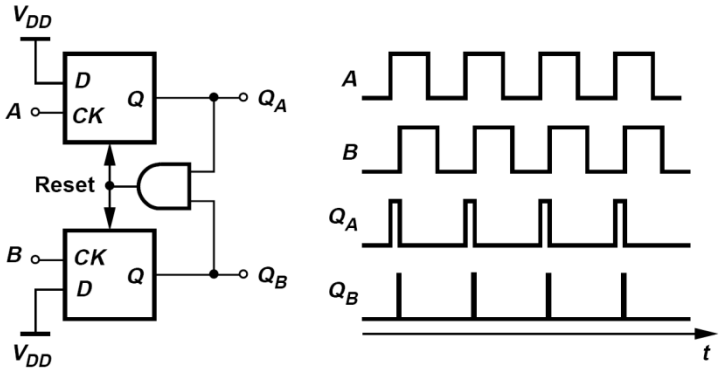


Figure 2.7 PFD implementation

The circuit in Figure 2.7 is the logic implementation of PFD. IT is consists of two edge triggered, resettable D flip-flops with their D inputs tied to logical ONE. Signals A and B act as clock inputs of DFF_A and $DFF_{B,}$ respectively. And the AND gate resets the flip-flops if Q_A=Q_B=1.

2.3.1 Drawbacks:

A. The logic gates delay

When Q_A=Q_B=1 the output of the flip-flops will be high for a duration equal to the delay of AND gate and the reset path as we see in Figure 2.8 ,if A and B are exactly in phase Q_A and Q_B exhibit these narrow “reset pulses “ which increase the ripple on the control of the VCO.

d_1: delay between rising edges of B and Q_B, decided by delay of DFF.
d_2: delay between rising edges of Q_B and RESET, decided by delay of AND gate.

d_3: delay between rising edge of RESET and falling edge of Q_A and B, decided by delay of DFF, usually smaller than d_1.
d_4: delay between falling edges of Q_B and RESET, decided by delay of AND gate.

Total delay=$d_1+d_2+d_3+d_4$.

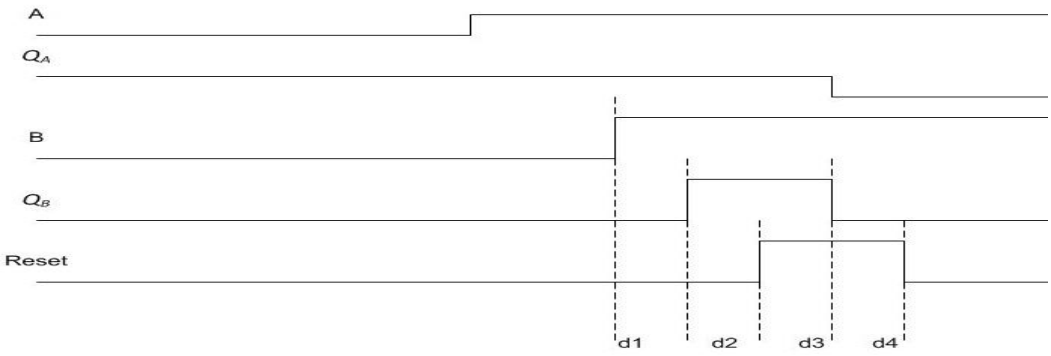


Figure 2.8 Reset process

A. The dead zone

One of the main problem in the PFD circuit is the Dead zone .the dead zone happens when the difference between the two inputs is close to 2π , the rising edge of the leading phase can fall into the reset region .during reset process, the PFD cannot detect the leading signal so it treats the following lagged signal as the leading one and generates reversed phase information to the charge pump as we see in Figure 2.9.This reversed phase information aggravates the cycle slips and elongates the frequency pull in time.

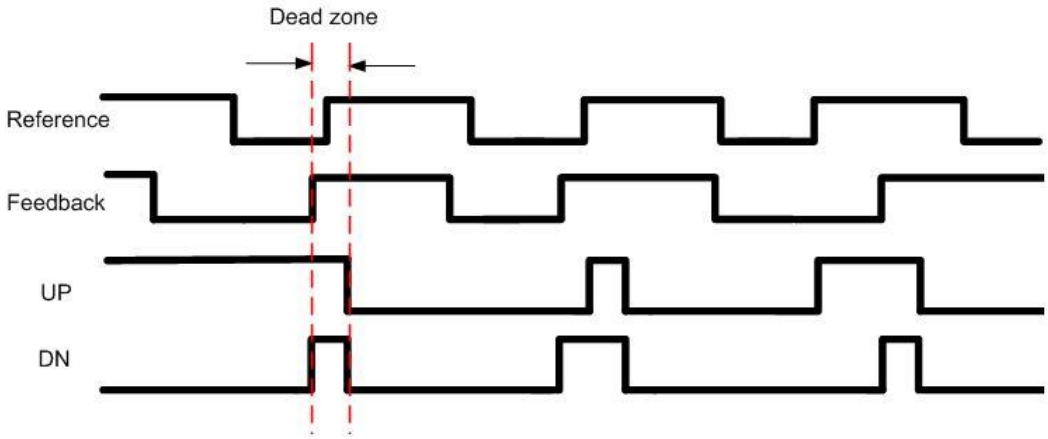


Figure 2.9 Dead zone process

2.4 The Proposed PFD:

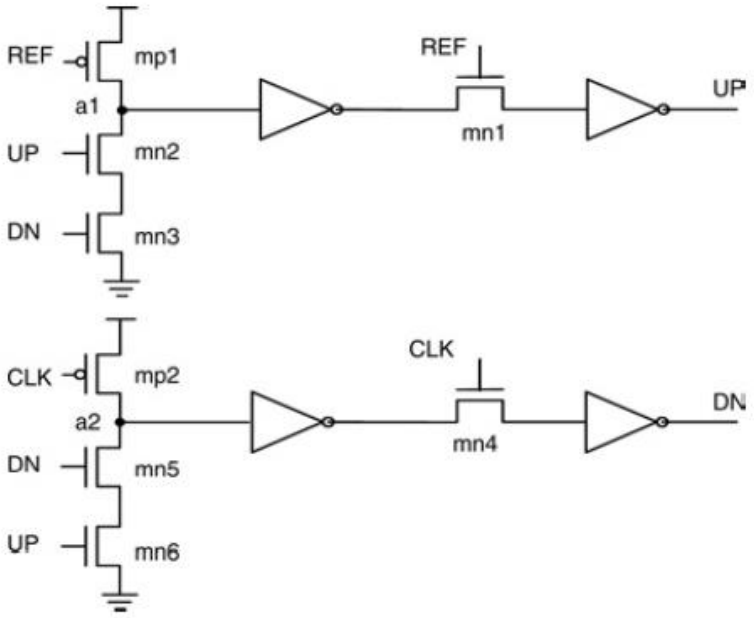


Figure 2.10 Proposed PFD

The proposed PFD has an important feature; which is the number of transistors is small so it decreases the total delay of the logic gates.
Secondly this PFD improves the acquisition speed by eliminating the blind zone near 2π. The circuit operates as follows. Initially the PFDs are in the initial state, and UP and DN signals are logic low. First, the rising edge of REF arrives and the UP signal goes to logic high, which would keep on until the rising edge of CLK arrives. The UP signal would turn on mn_2 and mn_6. Secondly, the rising edge of CLK arrives, so DN also goes to logic high and turns on mn_3 and mn_5. At the same time, both reset paths of the PFD turn on and discharge the points 'a_1' and 'a_2' to logic low. Because now CLK is definitely high and mn_4 turns on, DN can be pulled down rapidly before DN reaches the full value of the supply voltage. If at that time the level of REF is high, mn_1 turns on and UP can be pulled down. This corresponds to the situation that the phase difference of two input signals is in $[0,\pi]$. By contrast, if at that time the level of REF is low, mn_1 turns off and UP holds on, even when point 'a_1' is Discharged to logic low.

This corresponds to the situation that the phase difference of two input signals is in $[\pi, 2\pi]$. When the rising edge of REF leads that of CLK by $[\pi, 2\pi]$, UP would hold on; equally, when the rising edge of CLK leads that of REF by $[\pi, 2\pi]$, DN would also hold on. There would be a flat region in the characteristic, when the phase difference is in $[\pi, 2\pi]$. As shown in Figure 2.11, there is no inversion region near 2π in the characteristic of the proposed PFD.

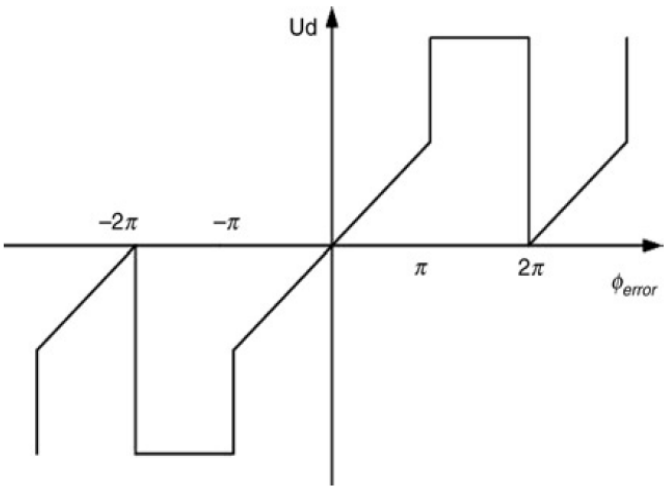


Figure 2.11 transfer characteristics of proposed PFD

2.4.1 Simulation results

This PFD is designed in a 0.13 μm CMOS while operating under a voltage supply of 1.2 V.
in this simulation we made the three cases of the inputs of the phase frequency detector :

1. When the reference frequency > feedback frequency.
2. When the reference frequency < feedback frequency.
3. When the reference frequency = feedback frequency.

The results of this simulation prove that this design overcome the dead zone by setting the output to high when the difference between the leading and the lagging signal exceed π.

1. The frequency of reference (net018) 48 MHz> The feedback frequency (net016) 40 MHz

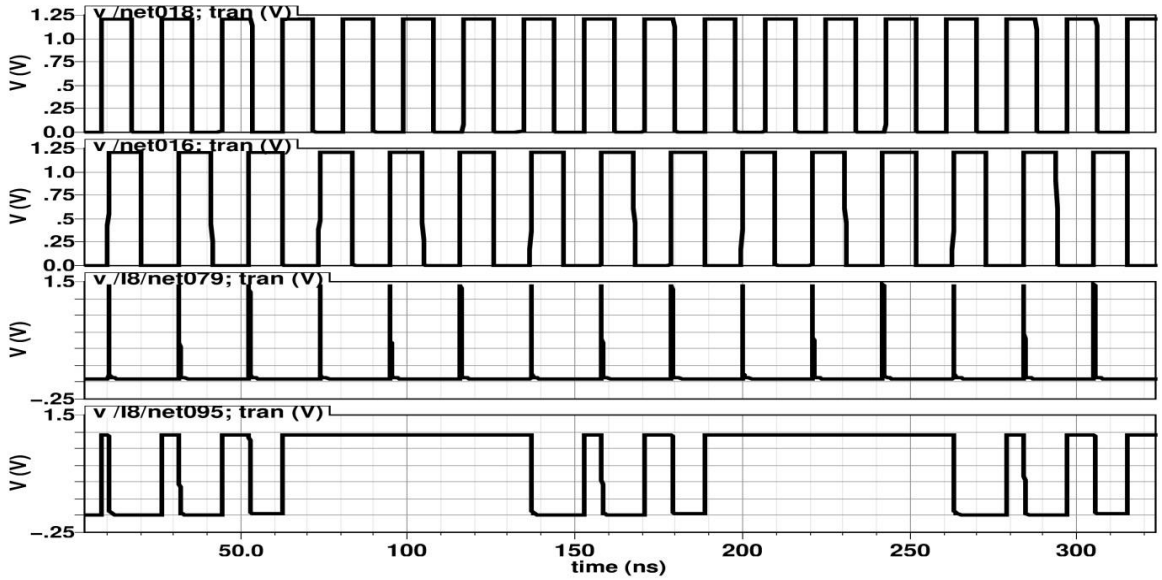


Figure 2.12 simulation result at first case.

∴ The UP signal (net095) produce pulses to accelerate the feedback frequency and the Down signal (net079) remains at zero.

2. The frequency of reference (net018) 48 MHz< The feedback frequency (net016) 56 MHz

∴ The Down signal (net079) produce pulses to slow down the feedback frequency and the UP signal (net095) remains at zero.

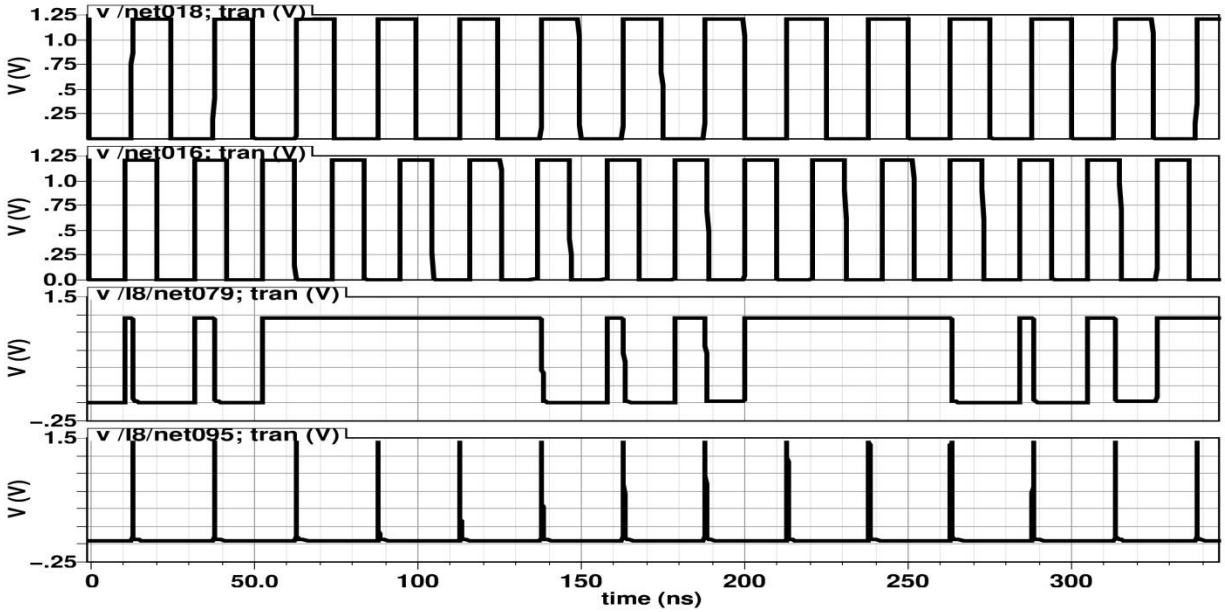
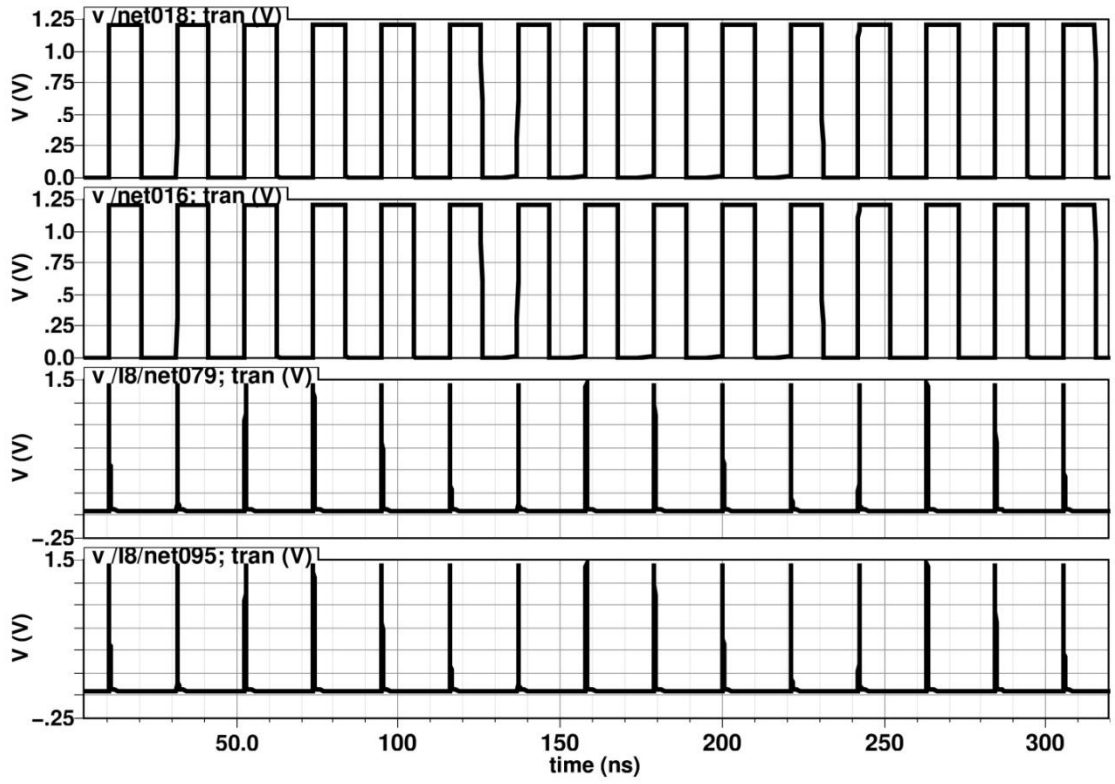


Figure 2.13 simulation result at second case

3. The frequency of reference (net018) 48 MHz= The feedback frequency (net016) 48 MHz

∴ The UP signal (net095) the Down signal (net079) remains at zero; this is the Lock state.



2.5 Conclusion

In this chapter we discuss the implementation of the phase detector circuit and its properties and drawbacks and we concluded on this result that we can't use it in our PLL because we are using multiple frequency ranges so it will no lock.

So we went through the phase frequency detector circuit and we study its specifications and problems and we discuss its basic design showing how it suffer from the logic gates delay and the dead zone problem.

Also we explained the dead zone problem and how its effect the performance of the whole loop.

Finally a new PFD without the blind zone is proposed. This PFD would saturate when the phase difference is in [π, 2π]. Its special characteristic is that it can speed up the acquisition process and Completely avoid the blind zone with a very simple circuit topology.

3 Charge Pumps

In a PLL the phase difference between the reference signal (often from a crystal oscillator) and the output signal is translated into two signals UP and DN. The two signals control switches to steer current into or out of a capacitor, causing the voltage across the capacitor to increase or decrease. In each cycle, the time during which the switch is turned on is proportional to the phase difference; hence the charge delivered is dependent on the phase difference also. The voltage on the capacitor is used to tune a voltage-controlled oscillator (VCO), generating the desired output signal frequency. The use of a charge pump naturally adds a pole at the origin in the loop transfer function of the PLL, since the charge-pump current is driven into a capacitor to generate a voltage (V=I/(SC)). The additional pole at the origin is desirable because when considering the closed-loop transfer function of the PLL, this pole at the origin integrates the error signal and causes the system to track the input with one more order. The charge pump in a PLL design is constructed in integrated-circuit (IC) technology, consisting of pull-up, pull-down transistors and on-chip capacitors. A resistor is also added to stabilize the closed-loop PLL.

3.1 Charge pump overview

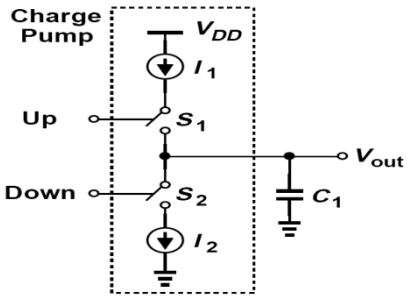


Figure 3.1 simple charge pump

- Switches S_1 and S_2 are controlled by the inputs “UP” and “Down”, respectively.
  A pulse of width ΔT on Up turns S_1 on for ΔT seconds, allowing I_1 to charge C_1 and V_{out} goes up.
  Similarly, a pulse on Down yields a drop in V_{out}.
- If Up and Down are asserted simultaneously, I_1 simply flows through S_1 and S_2 to I_2, creating no change in V_{out}.

3.1.1 PFD/CP operation

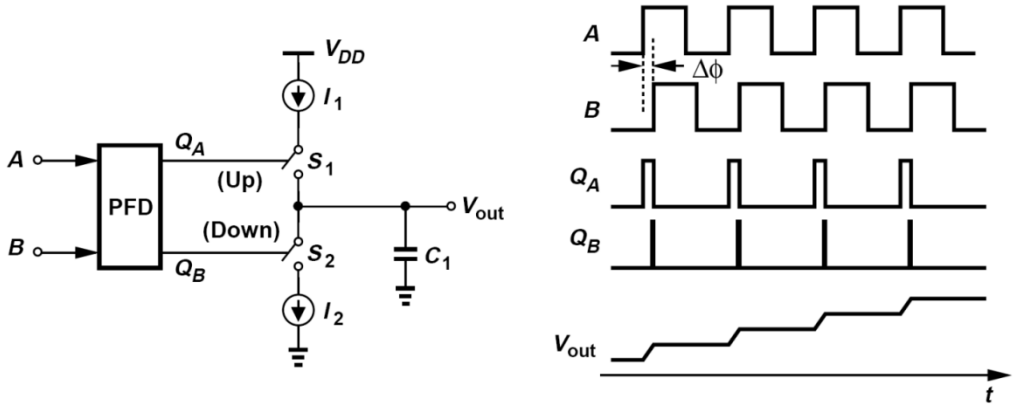


Figure 3.2 PFD/CP cascade

Infinite Gain: An arbitrarily small (constant) phase difference between *A* and *B* still turns one switch on, thereby charging or discharging C_1 and driving V_{out} toward +∞ or -∞.

We can approximate the PFD/CP circuit of figure above as a current source of some average value driving C_1.

For an input phase difference of $\Delta\Phi$ rad = $[\Delta\Phi/(2\pi)] \times T_{in}$ seconds, the average current is equal to I_p $\Delta\Phi/(2\pi)$ and the average slope, $I_p\,\Delta\Phi/(2\pi)/C_1$.

3.1.2 Charge pump transfer function

First if one of the integrators becomes lossy, the system can be stabilized. This can be accomplished by inserting a resistor in series with C_1. As we see in Figure 3.3.

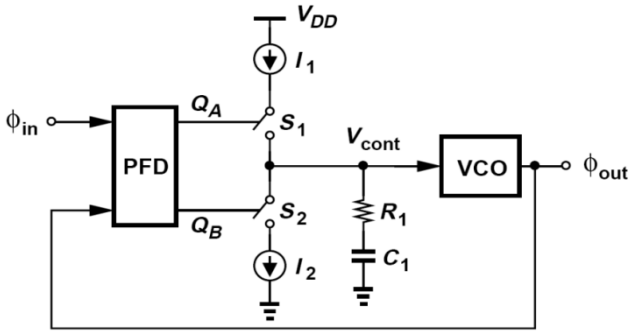


Figure 3.3 CP/ R_1 series with C_1

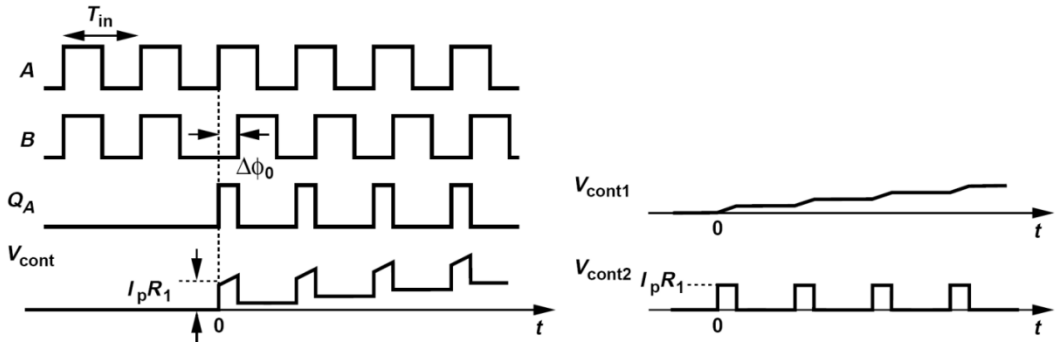


Figure 3.4 Inputs and outputs signal of CP

Approximate the pulse sequence by a step of height $(I_pR_1)[\Delta\Phi_0/(2\pi)]$:

$$Vcont(t) = \frac{\mathrm{Ip}\Delta\Phi\, t}{2\pi \mathrm{C1}}u(t) + \frac{\mathrm{Ip}\Delta\Phi \mathrm{R1}}{2\pi}u(t) \qquad \text{Equation 3.1}$$

$$\frac{Vcont}{\Delta\Phi}(s) = \frac{\mathrm{Ip}}{2\pi}\left(\frac{1}{\mathrm{C1}} + \mathrm{R1}\right) \qquad \text{Equation 3.2}$$

$$\mathrm{H}(s) = \frac{\frac{\mathrm{Ip}Kvco}{2\pi \mathrm{C1}}(\mathrm{R1C1s} + 1)}{s^2 + \frac{\mathrm{Ip}}{2\pi}KvcoR1s + \frac{\mathrm{Ip}Kvco}{2\pi \mathrm{C1}}} \qquad \text{Equation 3.3}$$

3.1.3 Charge pump stability

After writing the denominator of the transfer function as $s^2 + 2\xi\omega_n s + \omega_n^2$
we will find that

$$\xi = \frac{R1}{2}\sqrt{\frac{\mathrm{Ip}Kvco\ \mathrm{C}_1}{2\pi}} \qquad \text{Equation 3.4}$$

$$\omega_n = \sqrt{\frac{\mathrm{Ip}Kvco}{2\pi \mathrm{C}_1}} \qquad \text{Equation 3.5}$$

From Equation 3.4 we can observe that as $C_1 \uparrow \xi \uparrow$ therefore in this design we have a tradeoff between the stability and ripple amplitude.

3.1.4 Charge pump matching

One of the important issues in the charge pump design is the input and output matching first the input matching it's depend on the value of the capacitance seen by the PFD its must be equal for the up and down signal because if it will not be equal one of the signal will be delayed from the other.

The output systematic mismatch: the up and down current must be equal, if they are not equal the current injected into the loop filter will be Δ I +I which so it give a false value of V_{CONT} which will affect the frequency range.

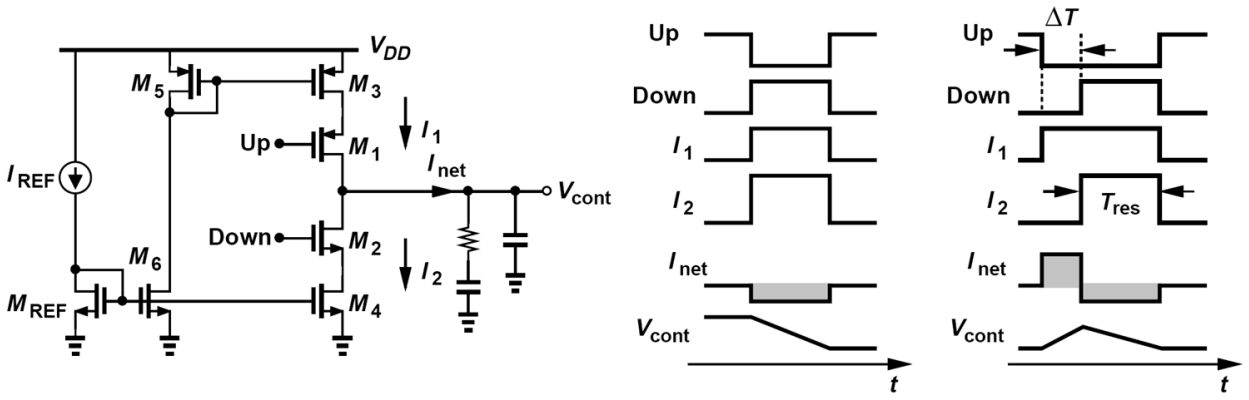


Figure 3.5 systematic mismatch

Different output voltage inevitably lead to opposite change in the drain source voltage of the current source, thereby creating a larger random mismatch between the current mirrors so it must be minimized this mismatch through increase the channel length. Random mismatch is calculated by the following equations:

$$\frac{\Delta I}{I} = \frac{2Vos}{Veff} \qquad \text{Equation 3.6}$$

$$Vos = \frac{Avt}{\sqrt{WL}} \qquad \text{Equation 3.7}$$

3.2 Different charge pump designs:

3.2.1 Basic single ended CP:

Simply put, a charge pump is a circuit that converts the output of the PFD to a current which then flows through the loop filter. The basic block diagram of a generic CP is shown in Figure 3.6.When the UP signal is active, current I_1 charges the output node and thus the voltage on the output capacitor increases, which in turn results in increasing the frequency of the VCO. When the DN signal is active, current I_2 discharges the output node and decreases the output voltage, which in turn results in decreasing the frequency of the VCO. Based on the discussion on the PFD, in a typical PLL, the reference signal is connected to the DN input of the CP, while the UP input is connected to the

proportional signal. The problem with the above CP is that when one switch is off, the voltage difference across its (non-ideal) current source will become zero. For example, assume UP is zero. The voltage of node A will become equal to V_{DD} to make I_1 equal to zero. When the UP signal arrives again, the switch connects node Out to V_{DD} creating an unwanted transient ripple.

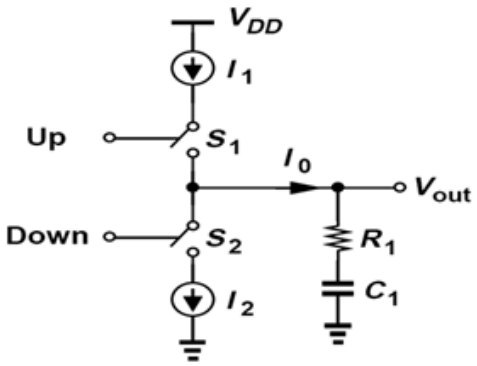


Figure 3.6 Basic single ended CP

3.2.2 Differential CP

The circuit proposed in Figure 3.7 solves the previous problem. Attempt should be made to make the value of the (non-ideal) current sources I_1 and I_2 equal, which could be difficult if the voltage of the node OUT is not $V_{DD}/2$. In this circuit, the inputs are differential, which is another advantage when compared to the circuit of Figure 3.6. The buffer used in this circuit must be designed to be rail to rail. Assume the next charge pump is used with the proposed PFD. When, the signal UP is "1" and the signal DN is "0", the current I1 flows into the output capacitor, thus raising the voltage of the output node. When UP and DN are both "1", ideally the voltage on the output capacitor should not change, as long as I_1 and I_2 are equal. When both UP and DN are "0", I_1 and I_2 flow through the left branch. Thus the current sources are always on, which eliminates the transient ripple.

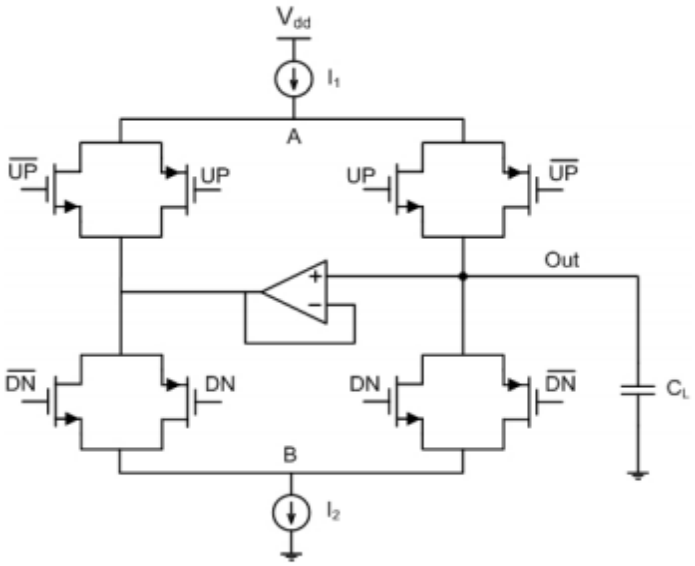


Figure 3.7: Differential CP

Let us assume that at the beginning of the operation, the voltage on the output node is $V_{DD}/2$. Also, assume that the clock is early. The output capacitor is charged with I_1 when pulse Y is "1" and discharged with I_2 when pulse X is "1". In the case of Figure 3.8, since the pulse width of X is greater than that of Y and $I_1 = I_2$, the capacitor is discharged more than it is charged, and as expected the loop filter voltage decreases to decrease the output frequency of the subsequent VCO. The problem with the above charge pump is that as the voltage on the output node decreases, so does the voltage of node B (and to a lesser extent the voltage of node A). Since the current sources are non-ideal, the decrease of the voltage of node B will decrease I_2. Eventually although the capacitor is discharged for more time, it is discharged with less current, and the CP output settles on a value that is more than expected. This limits the swing of the charge pump which in turn limits the PLL pull in range, a highly undesirable effect.

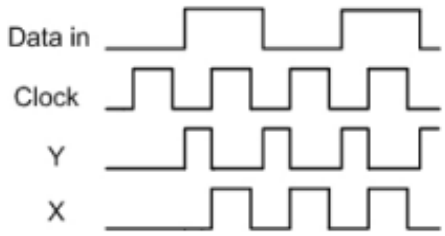


Figure 3.8 CP inputs

3.3 Proposed charge pump

To get the maximum tuning range from the VCO, its control voltage should be able to provide a large swing (ideally rail to rail). Since this is the output of the loop filter which is typically connected to the output node of the charge pump, the CP output voltage characteristic directly affects this swing. The ability of the charge pump to provide a wide swing, ideally rail-to-rail change, is therefore critical. So in our design we use a charge pump characterized by a wide output swing.

The charge pump is designed and simulated in a 0.13 um CMOS process and operates from a 1.2V supply.

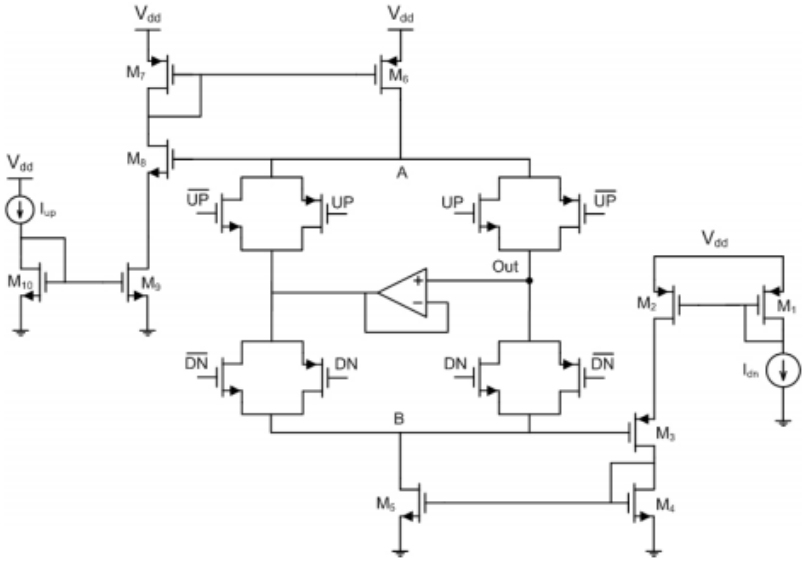


Figure 3.9: wide swing CP

3.3.1 How it works:

One way to solve the problem with the limited swing of the charge pump is to detect the change of the voltage on nodes A and B in Figure 3.7 and to try to keep the currents I_1 and I_2 constant. This is done in Figure 3.9. Assume that the output voltage is decreasing. This in turn decreases the voltage on node B. Now, however, the decrease on the voltage of node B is detected by M_3. This decrease, increases the gate source voltage of the PMOS transistor M_3, thus increasing its current. This current then flows through M_4, which increases its gate source voltage, which increases the gate source voltage of M_5, thus increasing its current. Therefore, although the decrease in the drain voltage of M_5 decreases its current, the increase in its gate-source voltage increases the current, and if the design is done carefully, these two effects will equalize each other, keeping I_2 almost constant. On the other hand, the decrease in the voltage of node A, decreases the gate source voltage of M_8, which decreases its current, which in turn decreases I_1.So the increase in I_1 due to the increase of its drain source voltage is balanced out by the current decrease due to the decrease of the gate source voltage of M_6.

A similar scenario takes place when the clock is late and the CP increases the voltage on the capacitor of the loop filter.

Finally, it is important to design the CP, such that the two mechanisms for changing the current of each current source balance each other out. In order to do this, consider the bottom branch once again. The amplifier can be modeled by a common-source amplifier with source degeneration, with M_3 as the main amplifier.

The gain for such a configuration can be expressed as:

$$A_v = \frac{-g_m R_D r_o}{R_D + R_S + r_o + g_m R_S r_o}$$ Equation 3.8

Where g_m is the Trans conductance of the transistor, R_D models the drain resistance, R_S is the resistance seen at the source and r_o is the transistor output resistance. In Figure 3.9, we have

$$A_v = \frac{-g_{m3}r_{o3} + \frac{1}{g_{m4}}}{\frac{1}{g_{m4}} + r_{o3} + r_{o2} + g_{m3}r_{o2}r_{o3}} \qquad \text{Equation 3.9}$$

The design should be done such that the above equation is equal to $-1/(g_{m5}r_{o5})$, so that the two mechanisms equalize each other. After some manipulations, the result can be given as

$$\frac{W_5^2}{L_5^4} = \frac{W_4^2}{L_4^2} \times \frac{1}{L_2^2} \qquad \text{Equation 3.10}$$

$$\frac{W_6^2}{L_6^4} = \frac{W_7^2}{L_7^2} \times \frac{1}{L_9^2} \qquad \text{Equation 3.11}$$

It should be noted that several conditions should be satisfied in the CP design for proper operation. Firstly, the bottom and the top current sources should be equal. Secondly, Equations (3) and (4) should be satisfied. Thirdly, transistors M_2, M_3, and M_4 and also M_7, M_8, and M_9 should operate in saturation. Also, note that current source transistors, M_5 and M_6, should not be minimum length transistors.

3.3.2 Simulation results:

The charge pump is designed and simulated in 0.13 μm CMOS with a supply voltage of 1.2 V. I_{UP} = I_{DN} = 800μA. The Up and Dn inputs are exactly the same in all cases and the circuits have all been designed to have an equal input capacitance. The Width of the PMOS transistors in the entire CP= 4× Width Nmos Transistors. The length of all the transistors= 3μm to achieve 1% random mismatch.

1.1.1.1 Output swing:

1. The output voltage of the charge pump when the 2 inputs are zeros

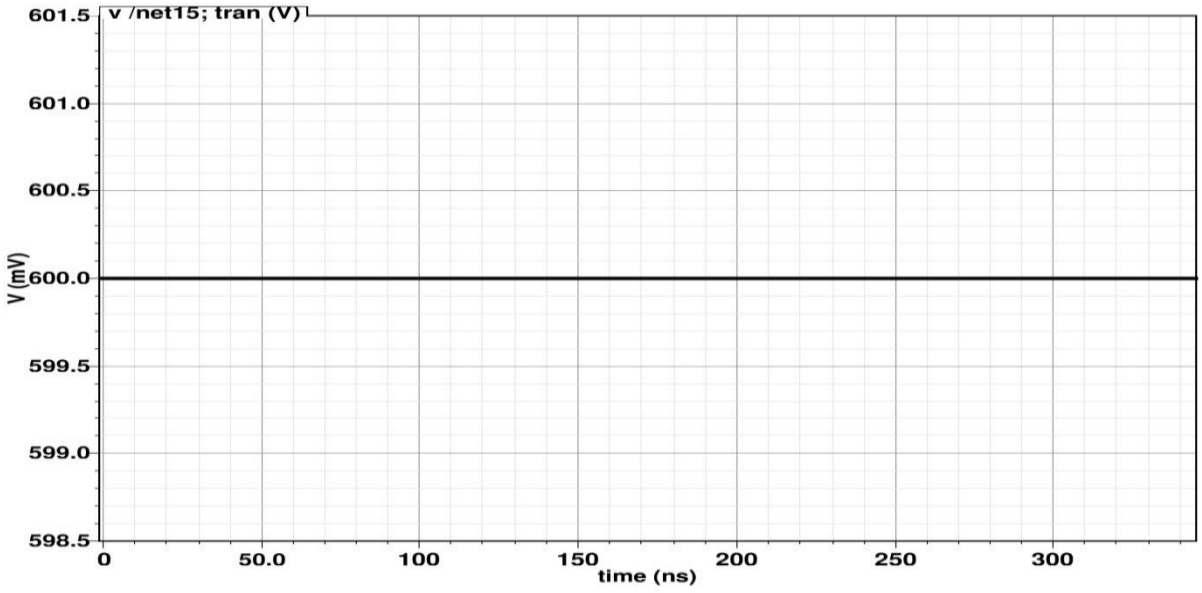


Figure 3.10

2. The output voltage when the DOWN signal generates pulses and the UP remains at zero

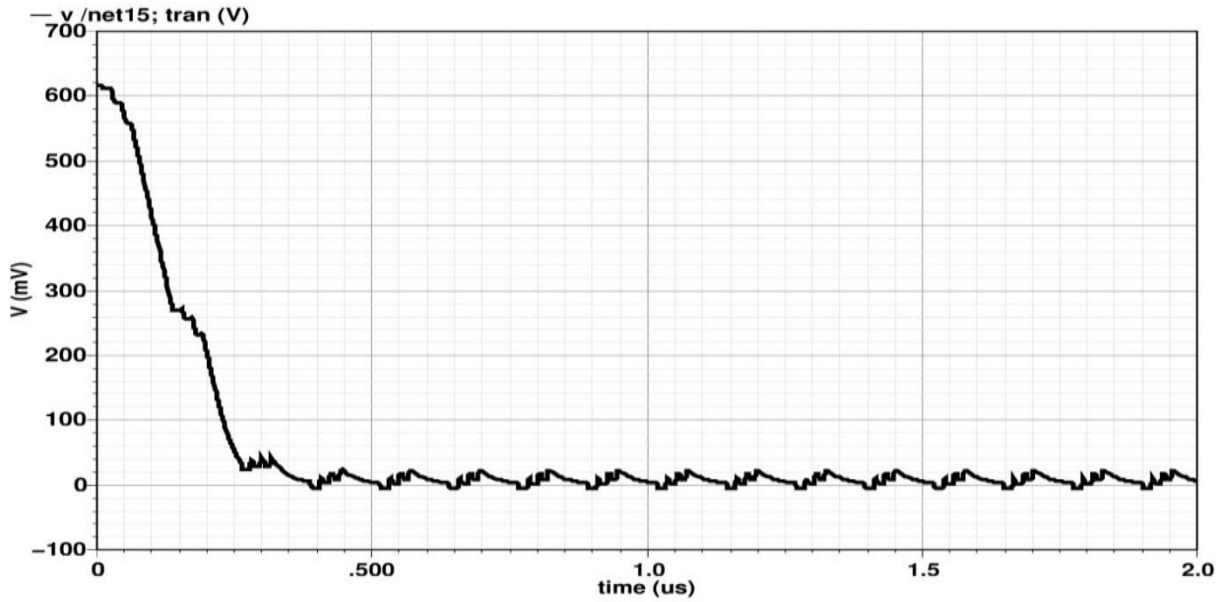


Figure 3.11

3. The output voltage when the UP signal generates pulses and the DOWN remains at zero

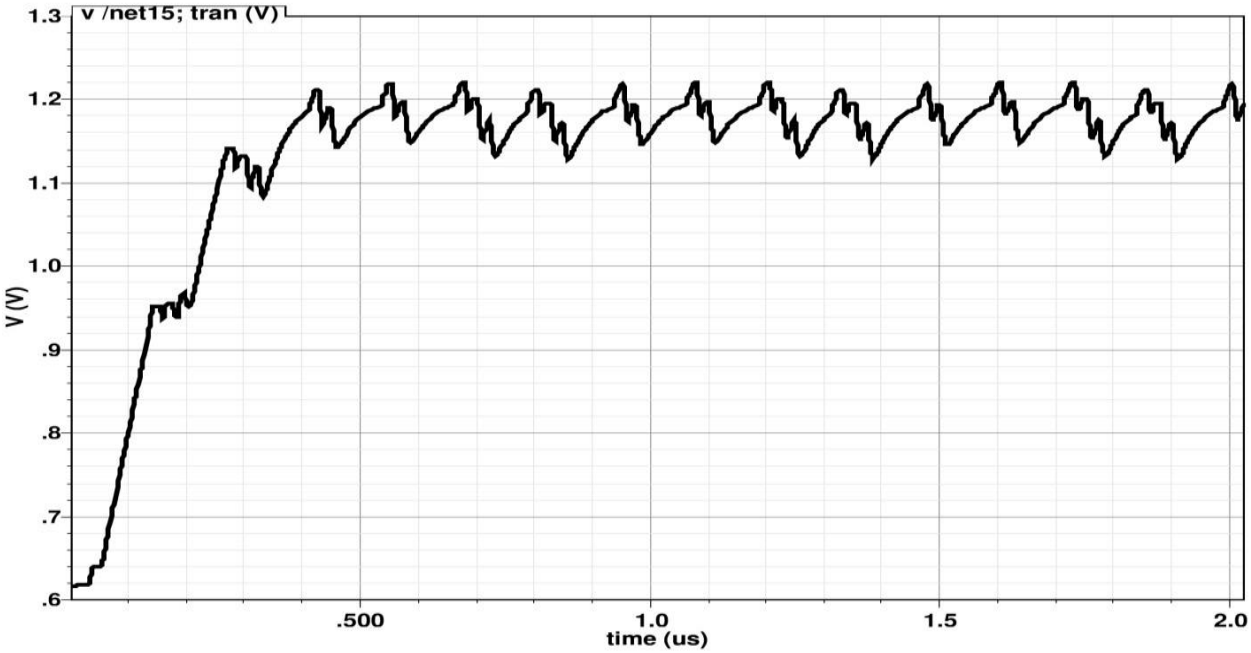


Figure 3.12

1.1.1.2 *Output current matching*

1. The UP current remains at 800μA from 170mV to 990m V.

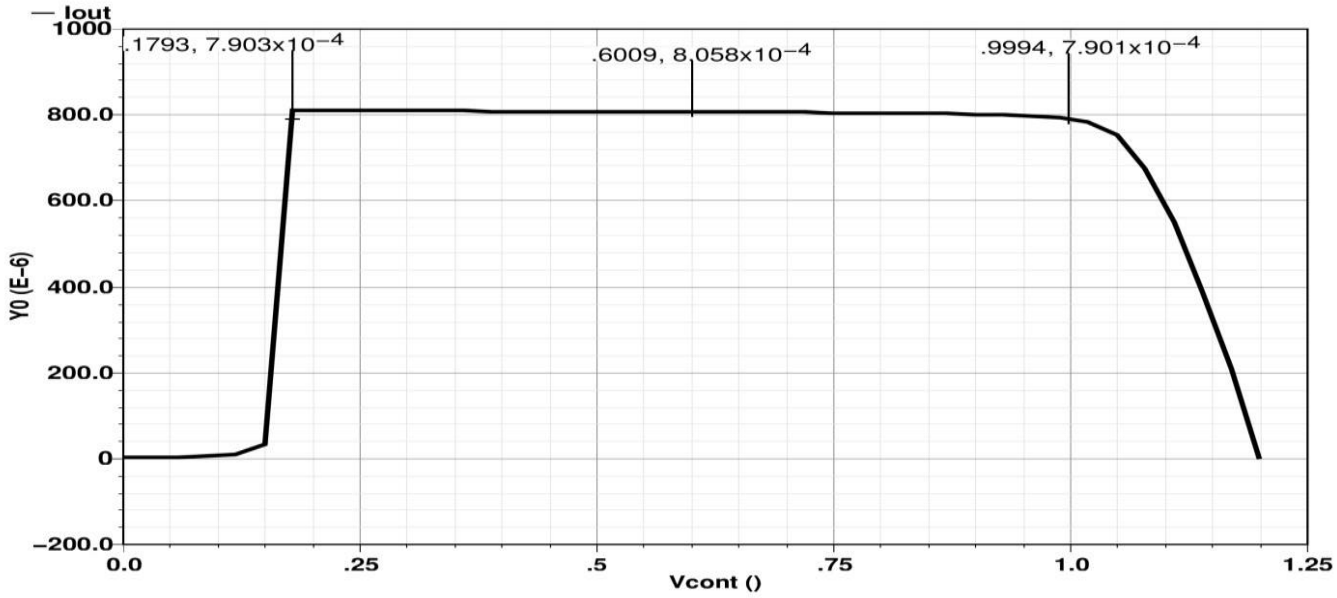


Figure 3.13

2. The DOWN current remains at -800μA from 190mV to 990m V.

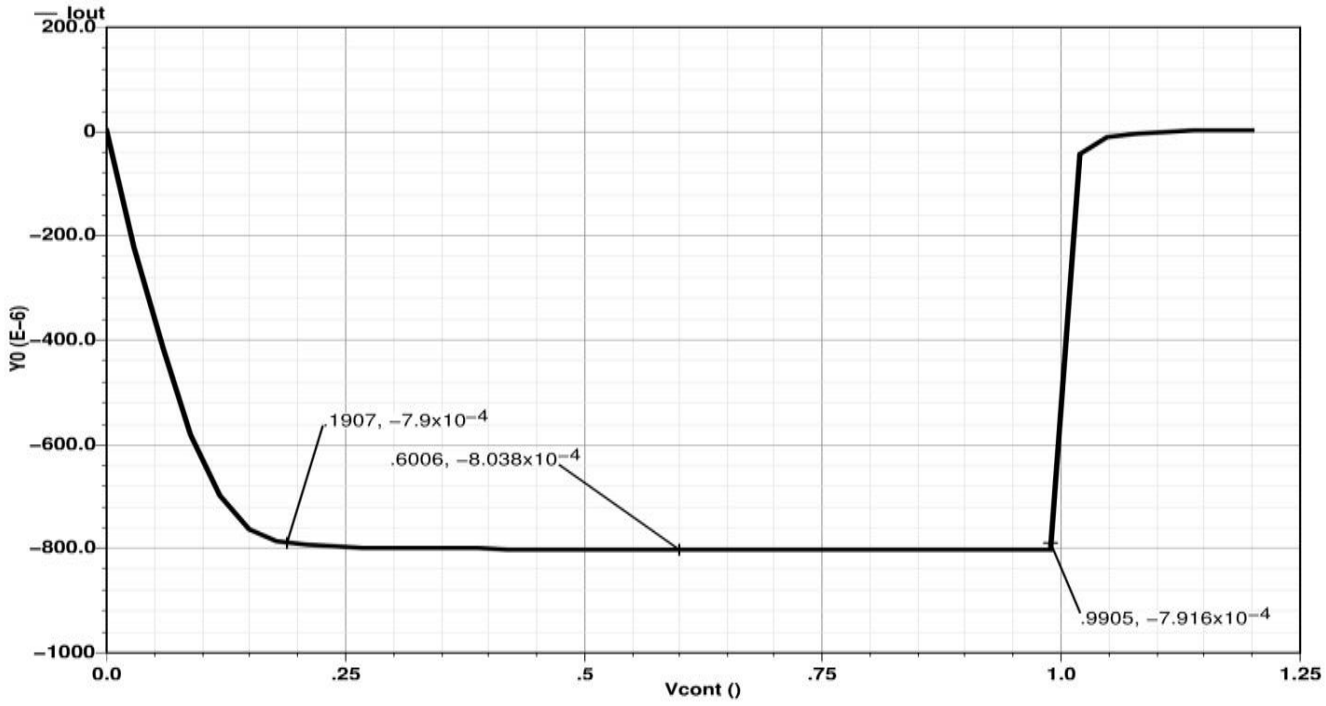


Figure 3.14

1.1.1.3 Output noise:

1. Squared output noise when the input is the UP signal:

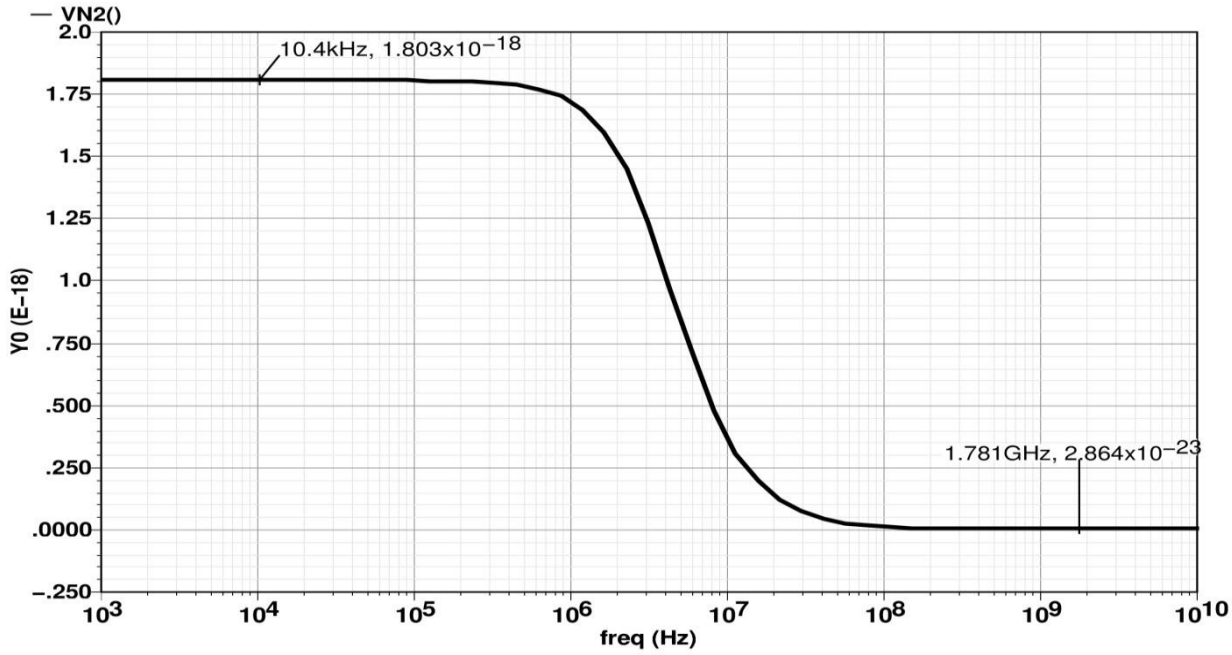


Figure 3.15

2. Squared output noise when the input is the DOWN signal:

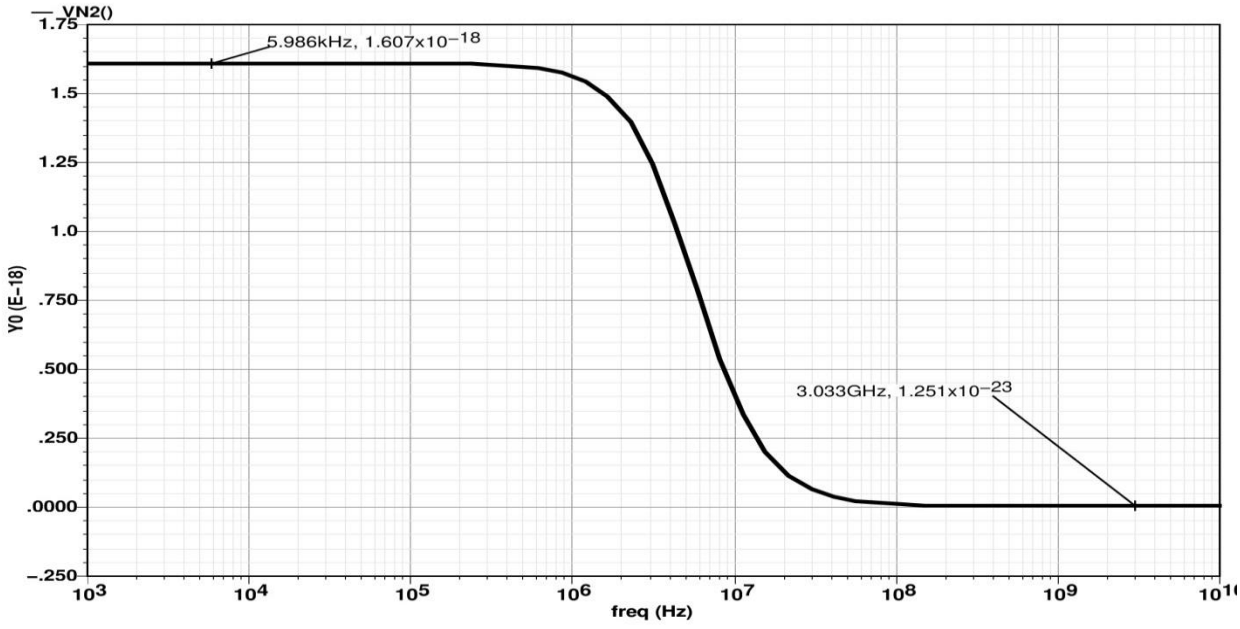


Figure 3.16

3.4 Conclusion

First From the previous sections in this chapter we discuss the importance of the charge pump block in the phase locked loop and its impact in controlling the range of frequencies of driven by the VCO.

Also we explained the circuit of the charge pump with all its properties and drawback and we suggest some solutions to enhance its performance.

And we discuss two basic design of the charge pump the single ended CP and the differential CP.

Secondly the basic challenges in the design of a charge pump are reviewed. The limited output swing of a widely used charge pump is discussed and a feedback-based solution to improve the output swing is proposed with highly matched up and down currents.

Finally, a wide-swing charge pump is designed in a 0.13 μm CMOS technology where simulations show improvement in the output swing as compared to other CP designs.

4 Voltage-Controlled Oscillators:

Voltage-Controlled oscillators are extensively used in all *RF* transceivers in both the transmitters and the receivers. This is caused by the fact that every mixer in the system drives its input from a stable oscillator. *VCOs* are typically embedded in a phase-locked loop system (as described in chapter 1). Since we are only designing the receiver of the *TV* tuner system, we are only going to design the *VCO* of the receiver. However, all the tradeoffs and discussion expressed here are equally applicable to the *VCO* of the transmitter.

CMOS oscillators for *TV* Tuner and *RF* systems are typically implemented as "*LC* oscillators" or "Ring oscillators". First, we are going to discuss the different challenges in designing a VCO for the TV Tuner system. Second, we are going to explain how a circuit can sustain an oscillating output using a feedback view of oscillators. Third, we will introduce the concept of tunable oscillators namely Voltage-Controlled Oscillators. Fourth, we will design and implement different *LC* oscillator topologies and conduct a brief comparison between them. Fifth, we will explain the concept of Ring oscillators, implement two distinct topologies and modify the second one to achieve a better performance. Also, a brief comparison between the three circuits is provided. Finally, we will conclude by choosing a topology to be implemented and explain why it is suitable for our *TV* Tuner direct conversion receiver.

4.1 Challenges in the Design of a VCO for the TV Tuner System:

There are many challenges in designing a *VCO* for the Digital *TV* Tuner receiver such as having a strong drive capability (output swing). Also, since we are going to use direct conversion receivers, the *VCO* must achieve a wide range of frequency operation (*48MHz* to *862MHz* which is equivalent to *1:18*) with a high level of signal purity (low phase noise). We also have to include an additional margin to this tuning range to account for temperature and process variations. Thus, this tuning range (*1:20*) is considered to be very wide and -as we will discuss in this chapter- is seldom achieved with a single *VCO*.

The first requirement usually belongs to "interface specifications" which is more serious in transmitters with power amplifiers whereas the second one is related to the "system specifications".

There is also a restriction on the Phase Noise associated with the *VCO's* output. Phase noise is used to express the broadening effect that the electronic devices noise has on the ideal impulse generated by the *VCO*. This effect is highly important and due care has to be given to the design of *VCO's* to keep the phase noise as low as possible. Phase noise usually exhibits tradeoffs with the tuning range and power dissipation imposing more challenges on the design.

As for the output waveform, it is preferred that the waveform generated by the *VCO* has a square-like nature and exhibits abrupt switching to reduce the noise of the Mixers and increase their conversion gain. It is also preferred that the waveform has a 50% duty cycle to suppress undesired effects such as direct feed-through. Also, frequency dividers –which are

driven by the *VCO's* output in typical *PLLs* - are pure digital circuits. Thus, they also require the *VCO* to generate a rail-to-rail oscillating square wave. In practice, the amplitude of the *VCO's* output is designed to be very large and the switching transistors are made very wide so as to produce a nearly rail-to-rail square-like wave.

As shown in Figure 4.1, the oscillator in the *TV* Tuner system is required to drive a mixer and a frequency divider. Thus, the *VCO* must provide a large enough output swing to ensure complete switching of the transistors in the next stage. If this cannot be achieved directly by the *VCO*, a buffer circuit is added to amplify the output swing and drive the next stage which could have large load capacitance.

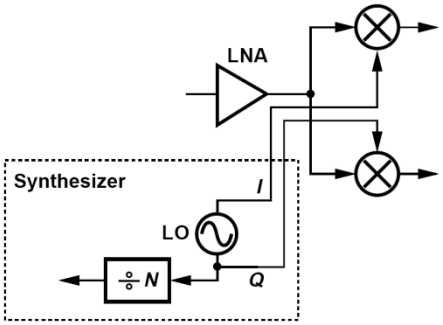


Figure 4.1: A typical *VCO* driving a Mixer and a Frequency Divider.

We will also focus our attention on differential *VCOs* because they usually have a higher performance compared to single ended ones. This is caused by the fact that balanced mixers -that require differential *VCOs*- outperform their unbalanced counterparts in terms of gain, noise and *DC* offsets. In addition, the leakage of the *VCO's* output to the received input is weaker when differential topologies are used.

There are other considerations related to the design of *VCOs* such as Power Dissipation and Supply Sensitivity. In most cases, *VCOs* are power hungry devices and have the highest level of power dissipation in the system. In general, the power dissipation of *VCOs* cannot be significantly reduced because it experiences tradeoffs with other key performance parameters such as phase noise, speed and tuning range. However, we are going to discuss some techniques that lower the phase noise without increasing the power dissipation.

As for Supply Sensitivity, in some cases, the output of the *VCO* is coupled to the supply voltage meaning that the frequency of oscillation is affected by the noise present in the supply. Figure 4.2 expresses a typical supply with flicker noise. This supply noise is translated to Phase Noise and modulates the oscillating frequency which is extremely undesirable. Bypass capacitors and other techniques are used to mitigate this effect.

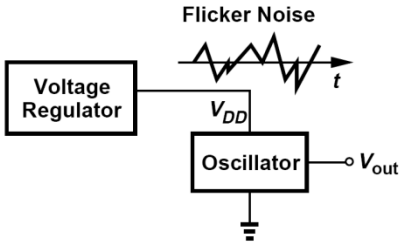


Figure 4.2: Voltage regulator with flicker noise used as a Supplier for an oscillator.

4.2 Feedback View of Oscillators:

To generate a periodic output voltage, oscillators must allow their internal noise to grow and self-sustain the developed periodic signal. In other words, the circuit has no input but can sustain an oscillating output indefinitely. As shown in Figure 4.3, oscillators may be modeled as poorly-designed linear negative feedback systems that have a zero or negative Phase Margin.

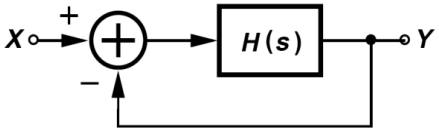


Figure 4.3: An oscillator modeled as a negative feedback

The oscillator modeled in Figure 4.3 has a transfer function given by

$$\frac{Y}{X}(s) = \frac{H(s)}{1 + H(s)}. \qquad \text{Equation 4.1}$$

For the oscillator to oscillate at a frequency of ω_1, according to "Barkhausen's criteria for oscillation", the following two requirements must be achieved

$$|H(s = j\omega_1)| = 1 \qquad \text{Equation 4.2}$$

$$\angle H(s = j\omega_1) = 180^O\,. \qquad \text{Equation 4.3}$$

In this case, $H(s = j\omega_1)$ becomes equal to *-1* and the gain from the input "*X*" to the output "*Y*" approaches infinity. This allows the circuit to amplify its own internal noise component at frequency ω_1 and sustain its oscillation. In other words, the closed-loop system has two imaginary poles at $\pm j\omega_1$ that achieve the required oscillations.

It is important to state that the noise does not have to be injected at the input of the loop. The noise can be present anywhere in the internal circuit of the *VCO* provided that the signal at ω_1 experiences a total phase shift of 360^O as it propagates around the loop. As shown in Figure 4.4, in the steady state, the signal returning (at point *A*) exactly coincides with the starting signal at point *A*.

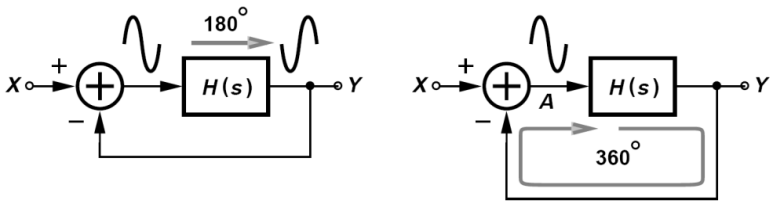


Figure 4.4: Phase shift around the loop to satisfy Barkhausen's Phase criteria.

The $\angle H(j\omega_1)$ component is called a frequency-dependent phase shift whereas the other 180^O phase shift is caused by the negative feedback of the loop (called a *DC* phase shift). This creates a total phase shift of 360^O i.e. a positive feedback that can sustain the oscillations at ω_1 because the loop gain is at least unity as expressed in Equation 4.2. Figure 4.5 illustrates the amplitude growth of a signal at ω_1 as it propagates around the loop.

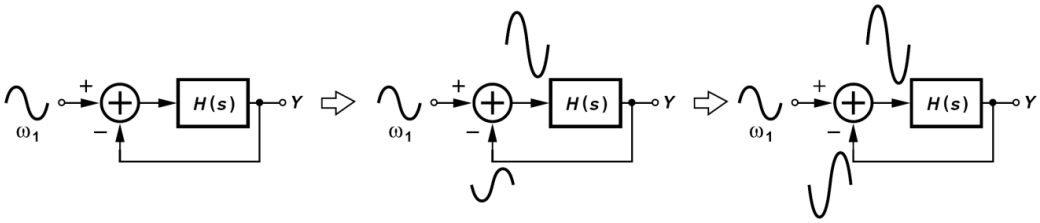


Figure 4.5: A typical oscillator during start-up.

If the loop gain is greater than unity, the amplitude growth occurs at a much faster rate because the returning waveform is amplified by the loop gain. This amplitude growth does not occur indefinitely. Due to nonlinearities in the loop and the saturating behavior of the transistors in the loop, the amplitude growth eventually stops and the oscillating output is maintained with a fixed voltage swing.

To sum up, we can predict the frequency of oscillation by determining the frequency at which both Barkhausen's criteria are satisfied. In practice, Barkhausen's criteria are not sufficient due to the presence of temperature and process variations. Thus, we design the loop gain to be twice or three times the theoretical value. Having discussed how a given oscillator can sustain an oscillating output, we are going to discuss the concept of *VCOs* i.e. tuning this oscillation over a wide range of frequency operation.

4.3 Voltage-Controlled Oscillators:

Most *RF* applications require the oscillator to be tunable over a certain frequency range. In our *TV* Tuner direct conversion receiver, this tuning range is considered to be very wide (*48 MHz* to *862MHz* which is equivalent to *1:20*). This wide tuning range may require more than one tunable oscillator each covering a small part of the whole frequency range. Thus, our goal is to design an oscillator whose frequency can be varied using an electronics signal (control voltage). Such a circuit is called a "Voltage-Controlled Oscillator" and is presented in Figure 4.6. It generates an oscillating output whose frequency is controlled by the input voltage signal (V_{CONT}).

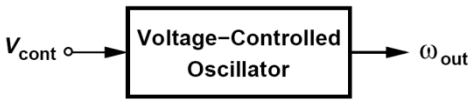


Figure 4.6: A typical *VCO* with an oscillating output whose frequency is controlled by the input voltage signal (V_{CONT}).

Figure 4.7 depicts a linear input/output characteristic of the *VCO*. It conceptually illustrates the desired behavior of a *VCO*. For instance, as the control voltage (V_{CONT}) varies from V_1 to V_2, the output frequency varies from ω_1 to ω_2 (the required tuning range). The slope of this characteristic is denoted as K_{VCO} and is called the "gain" or "sensitivity" of the *VCO* and is measured in *(rad/s)/V*. This characteristic can be expressed by the following simple equation of a straight line

$$\omega_{OUT} = \omega_O + K_{VCO}V_{CONT} \qquad \text{Equation 4.4}$$

where ω_O is the point of intersection of the characteristic with the vertical axis.

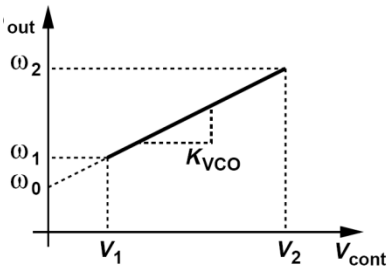


Figure 4.7: Linear Input/output characteristic of a *VCO*.

As explained in chapter 1, it is preferable that this characteristic be as linear as possible. In other words, it is desired to have a nearly constant sensitivity throughout the tuning range.

After explaining the concept of *VCOs*, we are going to discuss how to implement *VCOs* using *CMOS* technology. *CMOS* oscillators for *TV* Tuner and *RF* systems are typically implemented as "*LC* oscillators" or "Ring oscillators". In the next section, we will discuss the concept of *LC* oscillators and implement some of the famous *LC* oscillators' topologies.

4.4 *LC* Oscillators:

LC oscillators' topologies have become the dominant choice in most *RF* applications because of their robust operation. We are going to examine the feasibility of designing a cross-coupled *LC* oscillator for our *TV* Tuner system. We are going to start our discussion by developing a typical differential cross-coupled *LC* oscillator design. We are not going to discuss any single ended topologies because their performance is inferior to their differential counterparts, they have more stringent startup conditions and their noise directly corrupts the oscillation.

Our goal is to design a negative feedback oscillatory system that satisfies both of Barkhausen's criteria (Equation 4.2 and Equation 4.3) at our frequency of interest (ω_O) using *LC*-tuned amplifier stages as shown in the left part of Figure 4.8. It is important to highlight

that C_1 denotes the output capacitance seen at V_{OUT} and R_P is the parallel resistance of the inductor at resonance.

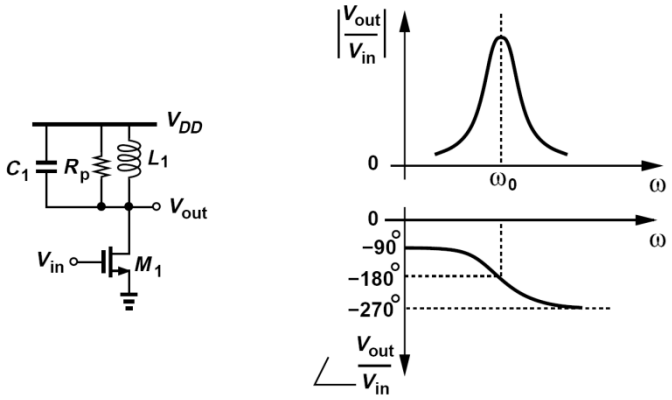


Figure 4.8: An *LC*-Tuned amplifier stage (on the left) and it Bode Plots (on the right).

At low frequencies, the inductor dominates the output load and the gain is given by

$$\frac{V_{OUT}}{V_{IN}} \approx -g_m L_1 s \qquad \text{Equation 4.5}$$

where $|V_{OUT}/V_{IN}|$ is extremely small and $\angle V_{OUT}/V_{in}$ is only -90^O. At the resonance frequency, the inductor and the capacitor cancel each other and the output load becomes R_P. The gain is given by

$$\frac{V_{OUT}}{V_{IN}} = -g_m R_P \qquad \text{Equation 4.6}$$

where $\angle V_{OUT}/V_{in}$ is now -180^O and the loop gain is adequate. At very high frequencies, the capacitor dominates the output load and the gain is given by

$$\frac{V_{OUT}}{V_{IN}} = -g_m \frac{1}{C_1 s} \qquad \text{Equation 4.7}$$

where $\angle V_{OUT}/V_{in}$ approaches -270^O and $|V_{OUT}/V_{IN}|$ is again extremely small. These results are plotted in the right part of Figure 4.8.

It is clear that no frequency satisfies both Barkhausen's criteria as the total phase shift does not even exceed 270^O at any frequency. However, the circuit provides a phase shift of 180^O at the resonance frequency (ω_O) with a possible high gain ($g_m R_P$). By adding an identical stage in cascade as shown in Figure 4.9, the total phase shift at resonance is increased to 360^O. The circuit will sustain the oscillation if the loop gain is greater than or equal to unity which results in the following restriction on g_m

$$(g_m R_P)^2 \geq 1. \qquad \text{Equation 4.8}$$

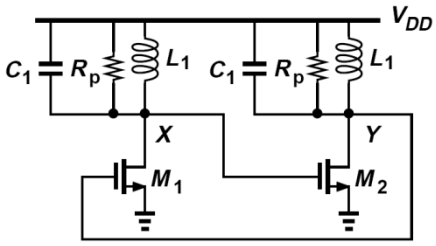


Figure 4.9: Cascading two identical *LC*-Tuned amplifiers to satisfy Barkhausen's Criteria.

If the restriction on g_m is achieved, the circuits own noise component at ω_O is amplified as it propagates around the loop. As shown in Figure 4.10, V_X and V_Y continue to grow while maintaining an 180^O phase shift between them. It is important to highlight that the inductive loads where able to provide an output waveform with peak voltages greater than the supply, while keeping a zero average voltage drop on the inductor. When the amplitude of the output waveform grows significantly, transistors M_1 and M_2 enter the triode region for a fraction of the period. As a result, the loop gain is reduced and the amplitude of V_X and V_Y stops growing.

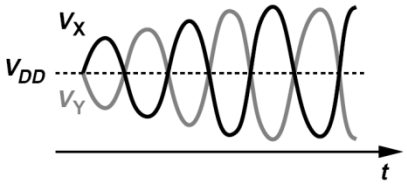


Figure 4.10: Output voltage exceeding the supply because of the inductive load used.

The circuit shown in Figure 4.9 is called a "cross-coupled" LC oscillator because of the connection between transistors M_1 and M_2 and is considered to be the core of most *CMOS* oscillators used in *RF* systems. The circuit can be redrawn as shown in Figure 4.11.

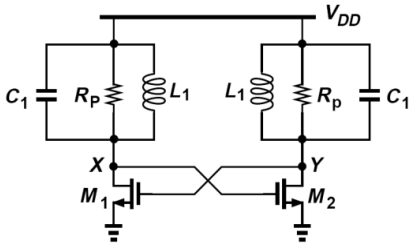


Figure 4.11: A typical cross-coupled *LC* oscillator.

The circuit shown in Figure 4.11 suffers from a critical drawback which is having a poorly-defined bias current. The bias current strongly depends on the mobility, threshold voltage and temperature ($\overline{V_{GS}} = V_{DD}$ and the bias current is determined accordingly). This drawback can be treated by adding a tail current source. This way, M_1 and M_2 operate as a differential pair with a differential output (V_X *and* V_Y). Figure 4.12 illustrates a typical Tail-Biased cross-coupled *LC* oscillator. It is more robust and can be viewed as an inductively loaded differential pair used in a positive feedback loop.

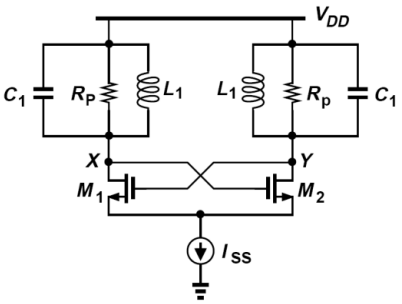


Figure 4.12: A Tail-Biased cross-coupled *LC* oscillator.

The amplitude of oscillation continues to grow until the differential pair enters the triode region for part of the period where the output experiences saturation. If M_1 and M_2 experience complete switching with abrupt edges, large output swings can be achieved and an almost square current waveform is injected into the *LC* pair (as shown in Figure 4.13). Each drain current waveform has a duty cycle of *50%* and thus the average of each is $I_{SS}/2$ and the peak amplitude is also $I_{SS}/2$. This square wave can be expanded using Fourier series into a fundamental signal that is multiplied by a resistance R_P and many harmonics that are effectively attenuated by the *LC* pair's selectivity. Using Fourier expansion, the peak amplitude of the fundamental signal is found out to be $(4/\pi)I_{SS}/2 = 2I_{SS}/\pi$. This signal is multiplied by a resistance R_P yielding a peak differential output swing given by

$$V_{XY} = \frac{4}{\pi} I_{SS} R_P \qquad \text{Equation 4.9}$$

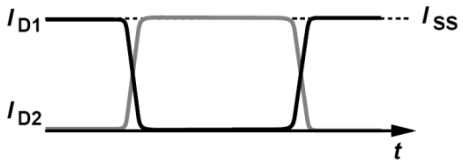


Figure 4.13: Square current waveform injected into the *LC* pair.

It is clear from Figure 4.10 that the output swings exceed the supply voltage for part of the period. This raises concerns about the transistors reliability, as the voltage drop across any two terminals of M_1 or M_2 may exceed the maximum value allowed by the technology used. In other words, the transistors may suffer from excessive stress because of this "above-supply" behavior of the output swing. Thus, the transistors dimensions must be carefully chosen so as to avoid stresses for a given output swing (I_{SS}).

The cross-coupled pair shown in Figure 4.12 also suffers from having high supply sensitivity. This is caused by the fact that the average value of the output voltage is V_{DD}. As shown in Figure 4.14, supply variations directly modulate the output waveform and the oscillation frequency.

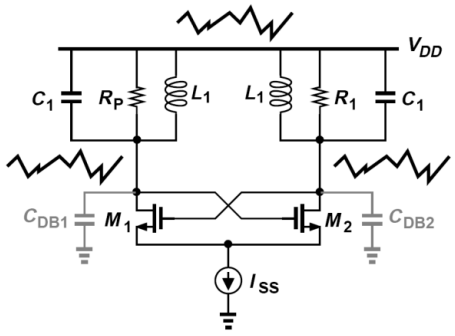


Figure 4.14: Modulation of the output voltage across C_{DB} by the supply voltage variations.

It is important to state that the choice of g_m given by Equation 4.8 to ensure sustaining of oscillations leads to a relatively small output swing. In practice, M_1 and M_2 are designed to experience complete current steering by having g_m much larger than $1/R_P$. In the following, we will attempt to design a Tail-Biased Cross-Coupled *LC VCO* for the receiver of our *TV* Tuner system.

4.4.1 An NMOS Tail-Biased Cross-Coupled LC VCO:

In order to convert the oscillator depicted in Figure 4.12 into a *VCO*, it is required to vary the resonance frequency of the *LC* pair with a certain control voltage. This way, the output frequency will be varied. This resonance frequency is given by

$$\omega_{OSC} = \frac{1}{\sqrt{L_1 C_1}}. \qquad \text{Equation 4.10}$$

Since it is extremely difficult to vary the inductance using an electronic signal, the only way to vary the resonance frequency is by varying the capacitor. This is achieved by adding a varactor which gives a capacitance that depends on the voltage across it (right part of Figure 4.15). An *NMOS* Tail-Biased *LC VCO* is presented in the left part of Figure 4.15 where the varactors (M_{V1} and M_{V2}) are in parallel with the *LC* pair.

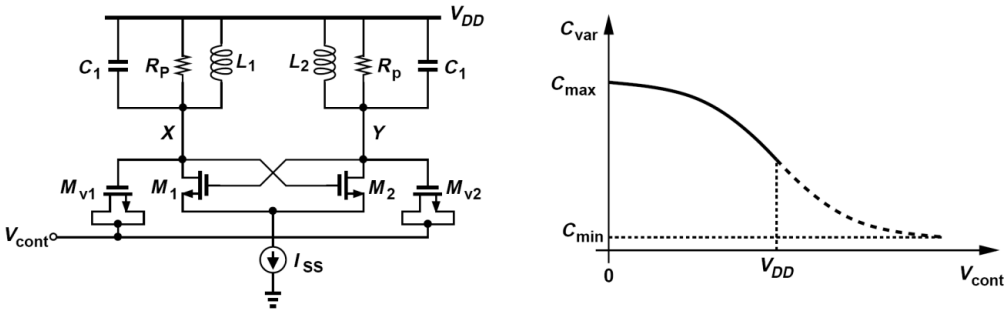


Figure 4.15: An *NMOS* Tail-Biased *LC VCO* using varactors (on the left) and the varactor's capacitance versus the control voltage (V_{CONT}) (on the right).

The average gate voltage of the varactors (M_{V1} and M_{V2}) is equal to V_{DD}. As a result, their gate-source voltage (V_{GS}) is always positive as their control voltage (V_{CONT}) is always smaller than the supply (V_{DD}). As expressed in the right part of Figure 4.15, the varactor's capacitance (C_{VAR}) decreases as V_{CONT} increases from *0* to V_{DD}. In this case, the average

voltage across the varactor varies from V_{DD} to *0* creating a monotonic decrease in C_{VAR}. The new oscillation (resonance) frequency is readily obtained as

$$\omega_{OSC} = \frac{1}{\sqrt{L_1(C_1 + C_{VAR})}}. \qquad \text{Equation 4.11}$$

It is important to state that C_1 does not represent a deliberately added capacitance to the design, on the contrary, the presence of C_1 leads to a narrower tuning range and is desired to be as small as possible. C_1 is added to model the inevitable capacitances seen at nodes *X* and *Y* such as C_{GS}, C_{GD}, C_{DB}, the parasitic capacitance of the inductor and the load capacitance of the next stage.

Another important remark is that the topology described in Figure 4.15 only utilizes about half of the available tuning range of the varactor because V_{CONT} cannot exceed the supply (V_{DD}) and the voltage drop on the varactor is always positive. This problem will be treated in the next section providing a wider tuning range of oscillation.

4.4.1.1 Design Procedures:

In the following, we are going to design a *VCO* that follows the topology in Figure 4.15 and demonstrate the simulation results obtained for this design. We start by designing a *VCO* that can cover only the upper range of frequency operation (*750MHz to 850MHz*) using *1.2V* supply and a power dissipation of *1.2mW*. We also require the output swing to be only from *0.77V* to *1.57V* for the *1.2V* supply so as not to stress the transistors. The design procedure is as follows:

1. Based on the given power dissipation budget, the maximum allowable I_{SS} is *1mA*. Thus, to achieve a voltage swing of *0.8V*, R_P is calculated from Equation 4.9 giving $R_P = \frac{V_{swing}}{\frac{4}{\pi} I_{SS}} = 630\Omega$.
2. Taking this value of R_P, determine the smallest inductor value that gives this parallel resistance at resonance. It will also be the inductor with the highest quality factor. In our design, this inductance value was found to be *5.6nF*.
3. Estimate the dimensions of M_1 and M_2 that achieve nearly complete current steering. This achieves the required voltage swing and satisfies the startup condition *($(g_m R_P)^2 > 1$ giving $g_m > 1.6m\Omega^{-1}$)*. We chose g_m to be five multiples of this value ($8.6m\Omega^{-1}$) to account for temperature and process variations. M_1 and M_2 are also chosen to have the minimum channel length to minimize their capacitance contributions to C_1.
4. Calculate $C_{VAR,\,MIN}$ that allows the oscillator to reach the upper end of the tuning range (*850MHz*). From Equation 4.11, $C_{VAR,\,MIN}$ was found to be *2.5pF*.
5. Using appropriate varactor models, determine the lower end of the tuning range that occurs at $C_{VAR,\,MAX}$ ($C_{VAR,\,MAX}$*= 3.7pF*). It was found that this value is *775MHz*.
6. Apply some fine adjustments to modify the tuning range to achieve our required range of frequency operation (*750MHz to 850MHz*).

This design procedure achieves the maximum possible tuning range for a given range of frequency operation, power dissipation and output voltage swing.

4.4.1.2 Simulation Results:

Figure 4.16 demonstrates the tuning curve achieved by this VCO. For the charge-pump designed in 1 with VCONT from 0.2V to 1V, the tuning range available by this VCO is from *775MHz* to *852MHz*. This tuning range is considered to be very narrow and in the following section we are going to propose a topology that achieves a wider tuning range.

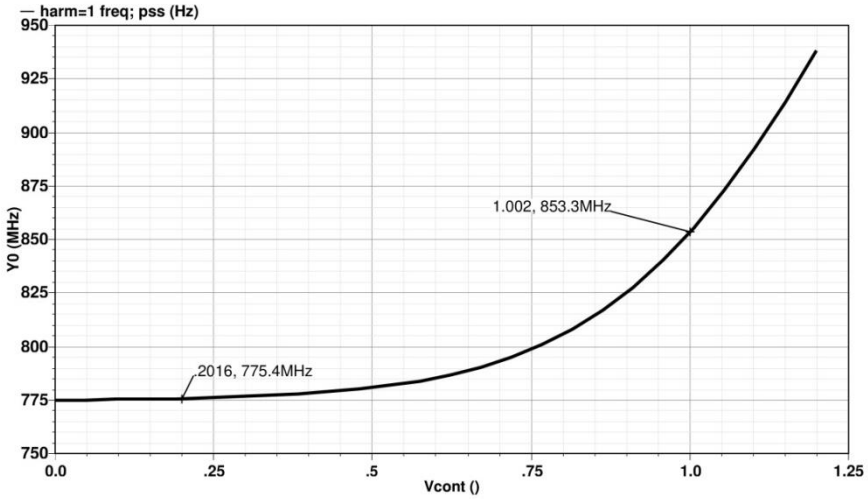


Figure 4.16: The tuning range of an *NMOS* tail-biased cross-coupled *LC* VCO.

Figure 4.17 confirms that the obtained single ended peak-to-peak output swing is *0.8V* as expected from our design procedures. It is clear that this *VCO* provides an output Common-Mode level of V_{DD} which could be an advantage or a disadvantage depending on the next stage.

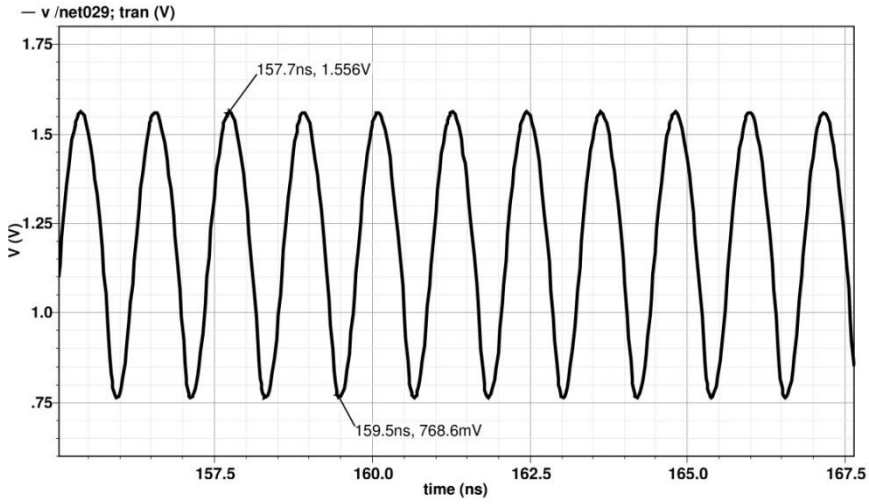


Figure 4.17: The single ended output swing obtained by the NMOS tail-biased *LC VCO*.

On the other hand, Figure 4.18 depicts the differential output swing obtained. It is clear that the generated signal has a clean sinusoidal waveform. This signal is readily applied to a buffer or an inverter stage to obtain a square-like waveform with *50%* duty cycle.

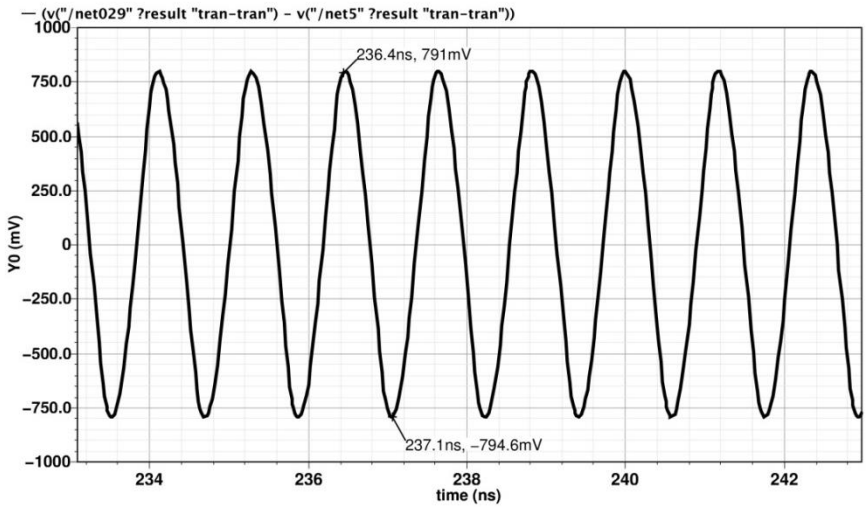


Figure 4.18: The differential output swing obtained by the *NMOS* tail-biased *LC VCO*.

As for the sensitivity (K_{VCO}) of the designed *VCO*, as evident by Figure 4.19, it varies by orders of magnitude across the tuning range. This behavior is undesirable in *PLLs* as the sensitivity is required to be almost constant throughout the tuning range.

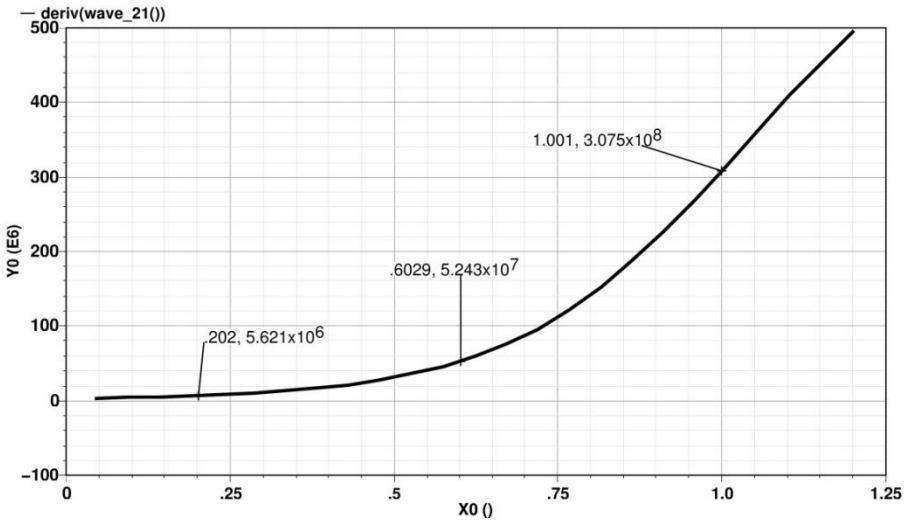


Figure 4.19: The sensitivity of the *NMOS* tail-biased *LC VCO*.

Figure 4.20 presents the power dissipation of the tail-biased *VCO* (*-28.87dB ≈1.2mW*). This result is in confirmation with our initial design.

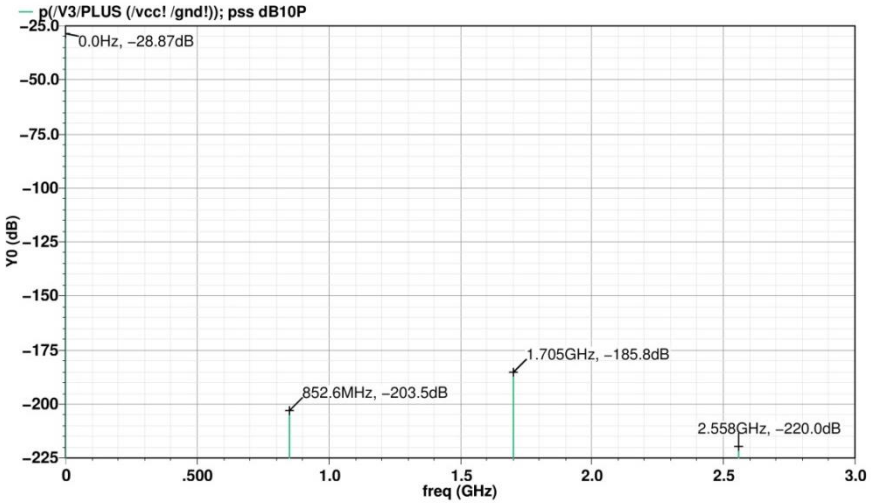


Figure 4.20: The power dissipation of the tail-biased *VCO*.

The power level of the output signal oscillating at frequency *852MHz* is *-6.06dBm* as expressed in Figure 4.21.

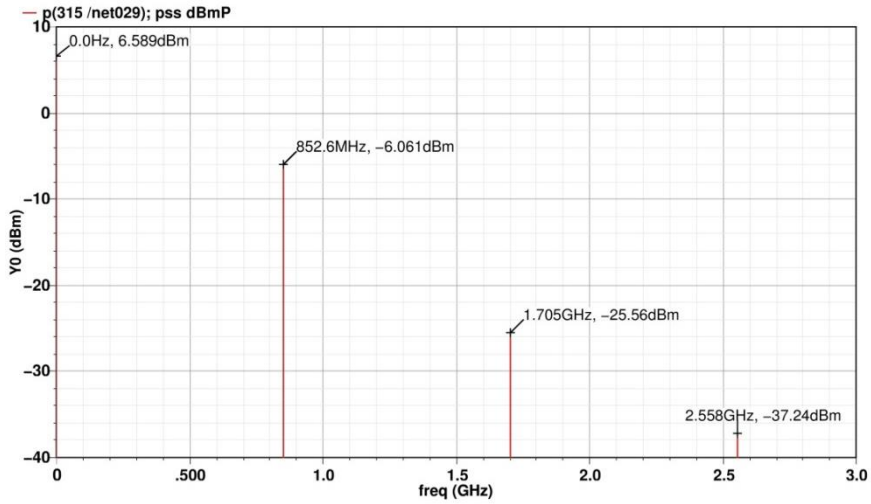


Figure 4.21: The signal power level at *852MHz* operating frequency.

Figure 4.22 shows the output phase noise at different offset frequencies from the carrier frequency (*852MHz*). The phase Noise was found to be *-122.7dBc/Hz* at *1MHz* offset frequency which is considered to be appropriate.

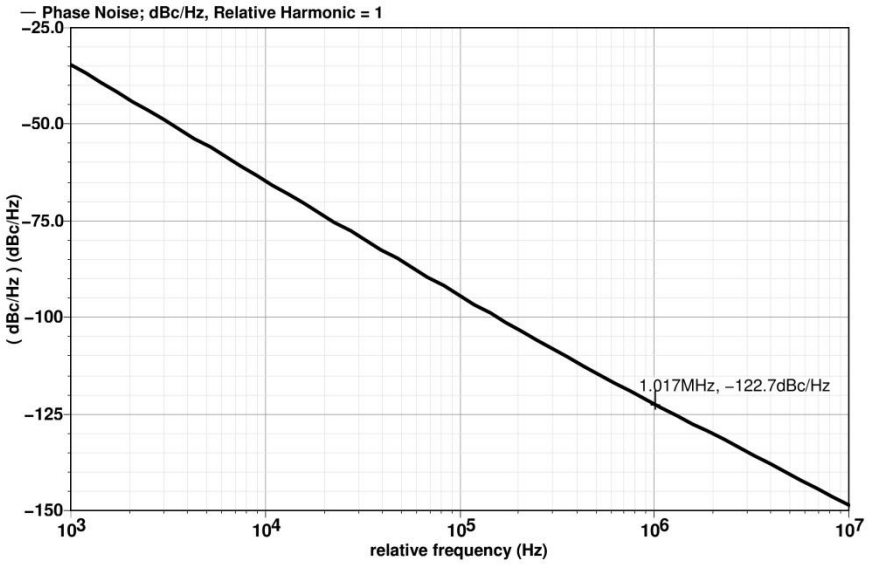


Figure 4.22: Output Phase Noise at different offset frequencies from a carrier frequency at *852MHz*.

It is evident from Figure 4.23 that the output phase noise does not exceed *-120dBc/Hz* for all the frequencies of operation.

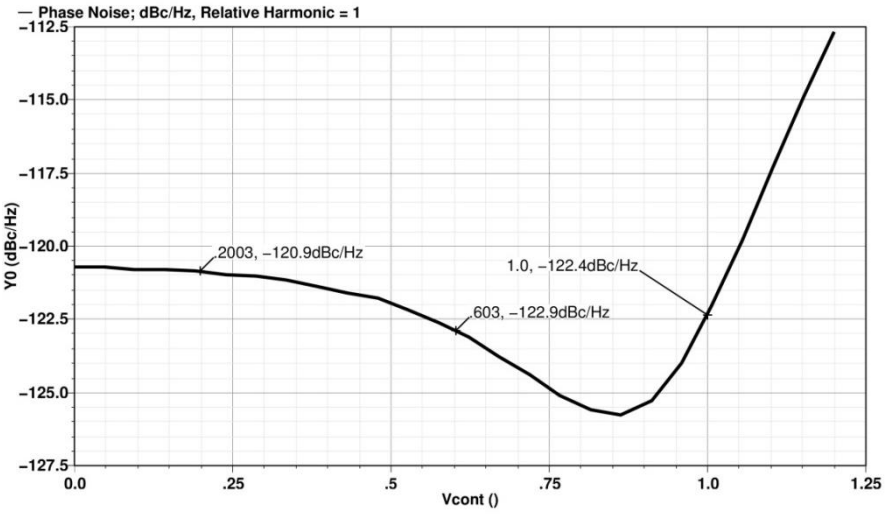


Figure 4.23: Phase Noise at *1MHz* offset frequency for all the frequencies of operation.

In this *VCO*, changing the supply voltage has a strong effect on the frequency of operation. For instance, at V_{DD} equals to *1.2V* the resulting output frequency is *852MHz*. As V_{DD} increases the resulting output frequency decreases and vice versa. This phenomenon is called frequency pushing and is illustrated in Figure 4.24.

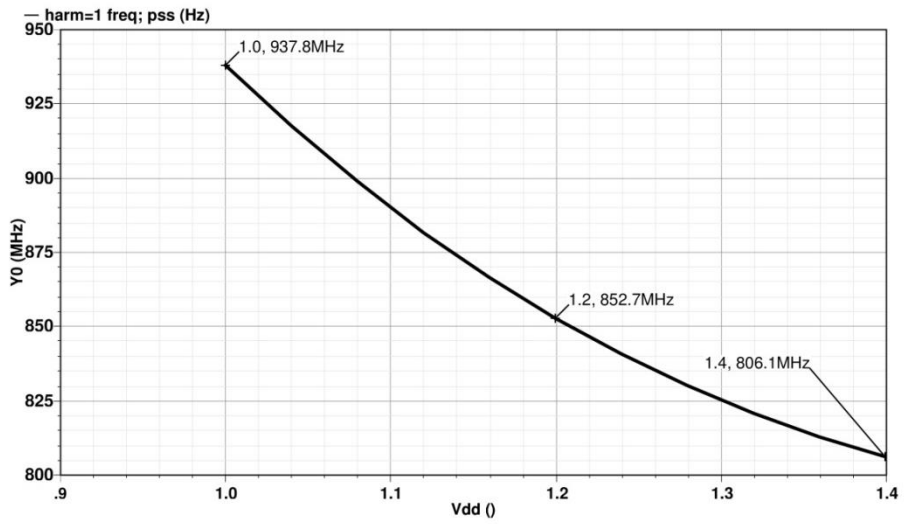


Figure 4.24: The effect of frequency pushing on a carrier frequency at *852MHz*.

4.4.1.3 Findings:

As evident from the ongoing discussion, the *NMOS* tail-biased cross-coupled *LC VCO* topology depicted in Figure 4.15is a robust topology. However, it suffers from two major drawbacks which are having a narrow tuning range and a high phase noise. These two drawbacks are treated in the following sections where the next section proposes a topology that achieves a wider tuning range whereas the one after demonstrates a topology with a much lower phase noise.

4.4.2 An NMOS Top-Biased Cross-Coupled LC VCO:

In this section, we are going to discuss a topology that achieves a much wider tuning range compared to its tail-biased counterpart. The reason for the narrow tuning range available in the tail-biased topology is because the voltage drop across the capacitor is always positive (V_{CONT} cannot exceed V_{DD} as it is usually provided by a charge-pump). This way, only half of the tuning range of the varactor is utilized as demonstrated by Figure 4.25.

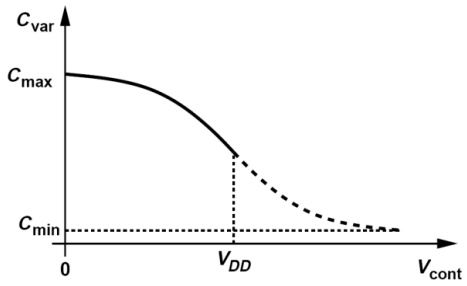


Figure 4.25: Tunability of the varactor's capacitance with the control voltage (V_{CONT}) where only half of the tunining range is utilized by the tail-biased topology.

The top-biased topology depicted in Figure 4.26 defines the bias current by virtue of a top current source. This topology expands the tuning range by exploiting the capacitance range of the varactor corresponding to negative voltage drop ($V_{GS} < 0$). Thus, this topology allows

both positive and negative voltage drops across the varactor. This way, almost the entire tuning range of the varactor is utilized as demonstrated in Figure 4.27.

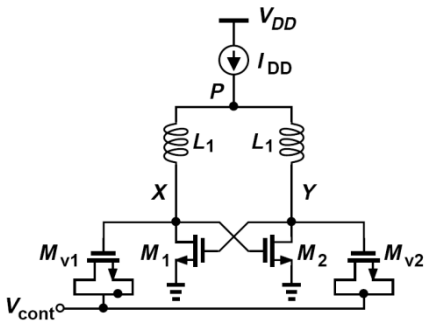


Figure 4.26: A typical *NMOS* Top-Biased Cross-Coupled *LC VCO*.

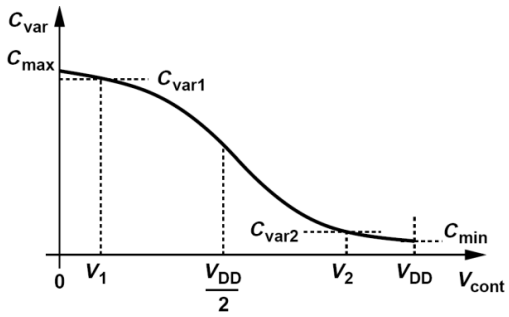


Figure 4.27: Tunability of the varactor's capacitance with the control voltage (V_{CONT}) where almost the entire tuning range is utilized by the top-biased topology.

It is important to state that we have to carefully select the transistors' dimensions such that the Common-Mode level of the output is equal to half the supply voltage ($V_{DD}/2$).

Consequently, as V_{CONT} varies from *0* to V_{DD}, the voltage drop across the varactor (V_{GS}) varies from $+V_{DD}/2$ to $-V_{DD}/2$ (sweeping almost the entire capacitance range as shown in Figure 4.27).

The startup condition of this topology is exactly the same as that of the tail-biased one and is directly calculated from Equation 4.8. The same applies for the frequency of oscillation (Equation 4.11) and the output voltage swing (Equation 4.9).

4.4.2.1 Design Procedures:

In the following, we are going to design a *VCO* that follows the topology in Figure 4.26 and demonstrate the simulation results obtained for this design. As done before, we start by designing a *VCO* that can cover only the upper range of frequency operation (*550MHz* to *850MHz*) using *1.2V* supply and a power dissipation of *1.2mW*. To prevent stressing of the transistors, we require the output swing to be from almost *0.3V* to *0.9V* for the *1.2V* supply. The design procedure is as follows:

1. Based on the given power dissipation budget, the maximum allowable I_{SS} is *1mA*. Thus, to achieve a voltage swing of *0.6V*, R_P is calculated from Equation 4.9 giving $R_P = \frac{V_{swing}}{\frac{4}{\pi} I_{SS}} = 471\Omega$.

2. Taking this value of R_P, determine the smallest inductor value that gives this parallel resistance at resonance. It will also be the inductor with the highest quality factor. In our design, this inductance value was also found to be *6.7nF*.
3. Estimate the dimensions of M_1 and M_2 that achieve an output common-mode level of *0.6V* while also achieving nearly complete current steering. This achieves the required voltage swing and satisfies the startup condition $((g_m R_P)^2 > 1$ giving $g_m > 2.1m\Omega^{-1})$. The value of g_m that satisfies the two requirements was found to be $(3.8m\Omega^{-1})$ which is almost twice the theoretical value. This does not leave a large margin for temperature and process variations. M_1 and M_2 are also chosen to have the minimum channel length to minimize their capacitance contributions to C_1.
4. Calculate $C_{VAR,\,MIN}$ that allows the oscillator to reach the upper end of the tuning range (*850MHz*). From Equation 4.11, $C_{VAR,\,MIN}$ was found to be *1.9pF*.
5. Using appropriate varactor models, determine the lower end of the tuning range that occurs at $C_{VAR,\,MAX}$ ($C_{VAR,\,MAX} = 7pF$). This value was found to be *530MHz*.
6. Apply some fine adjustments to modify the tuning range to achieve our required range of frequency operation (*550MHz* to *850MHz*).

As mentioned before, this design procedure achieves the maximum possible tuning range for a given range of frequency operation, power dissipation and output voltage swing.

4.4.2.2 Simulation Results:

Figure 4.28 demonstrates the tuning curve achieved by this *VCO*. For the charge-pump designed in 1 with V_{CONT} from *0.2V* to *1V*, the tuning range available by this *VCO* is from *530MHz* to *888MHz*. This tuning range is much wider than that of the tail-biased *VCO* as expected.

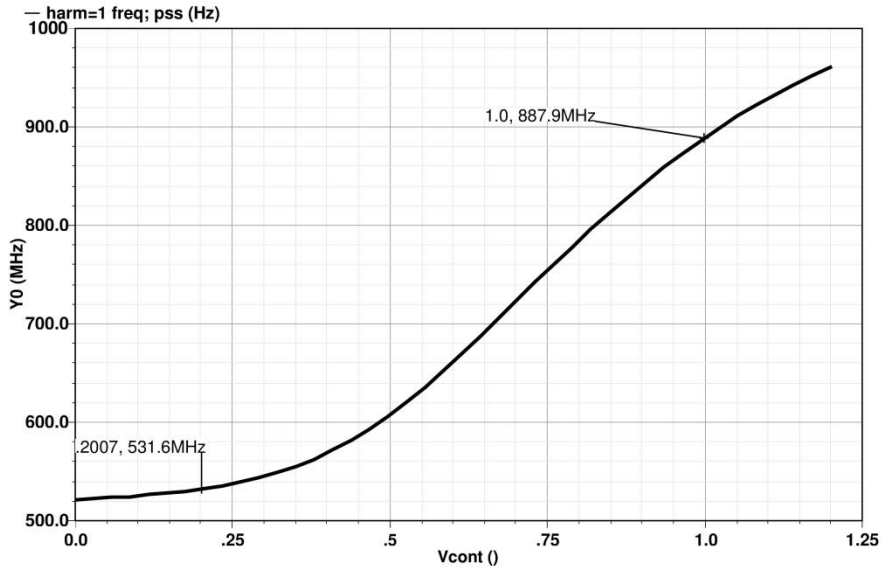


Figure 4.28: The tuning range of an *NMOS* top-biased cross-coupled *LC VCO*.

Figure 4.29 presents the obtained single ended output swing. The upper end of the voltage swing is *0.9V* as designed whereas the lower end reaches a value of almost zero. This is

caused by the fact that M_1 and M_2 achieve complete current switching and the sources of the transistors are connected directly to the ground. This large output swing occurs around a common-mode level of *0.6V* as expected from our design procedures. It is clear that this *VCO* provides an output Common-Mode level of $V_{DD}/2$ by design which could be an advantage or a disadvantage depending on the next stage.

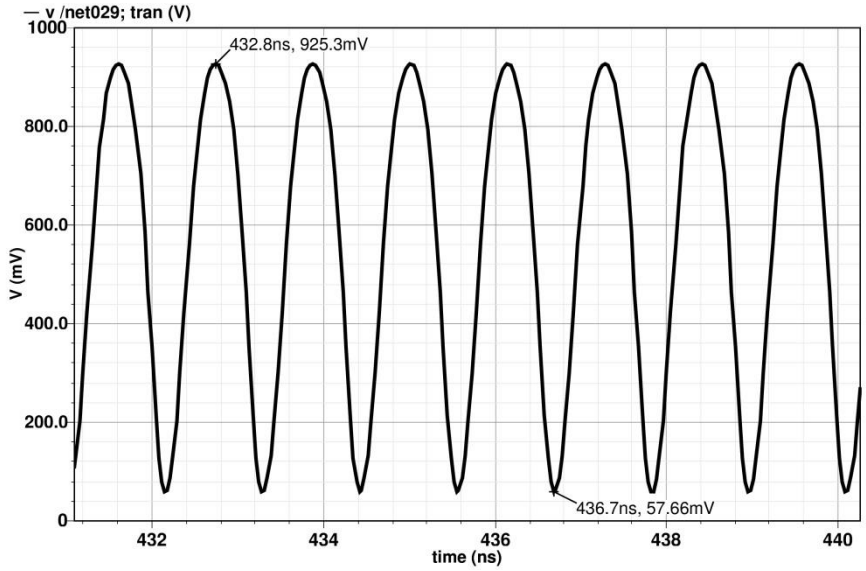


Figure 4.29: The single ended output swing obtained by the *NMOS* top-biased *LC VCO*.

On the other hand, Figure 4.30 depicts the differential output swing obtained. It is clear that the generated signal has a clean sinusoidal waveform. This signal is readily applied to a buffer or an inverter stage to obtain a square-like waveform with *50%* duty cycle.

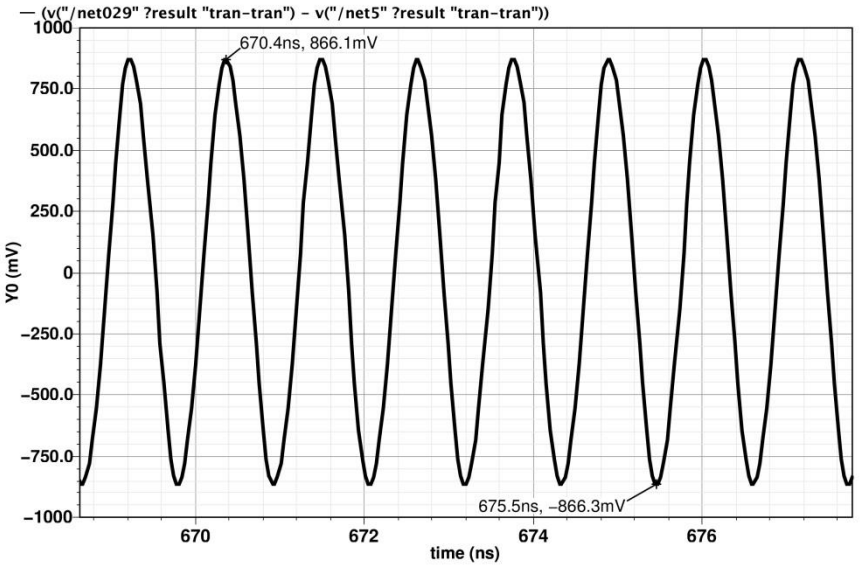


Figure 4.30: The differential output swing obtained by the *NMOS* top-biased *LC VCO*

As for the sensitivity (K_{VCO}) of the designed *VCO*, as evident by Figure 4.31, it increases by a factor of ten then decreases slightly again. It is clear that this behavior is far from linear and is undesirable. On the other hand, the desired behavior in *PLLs* is to have a sensitivity that is almost constant throughout the tuning range.

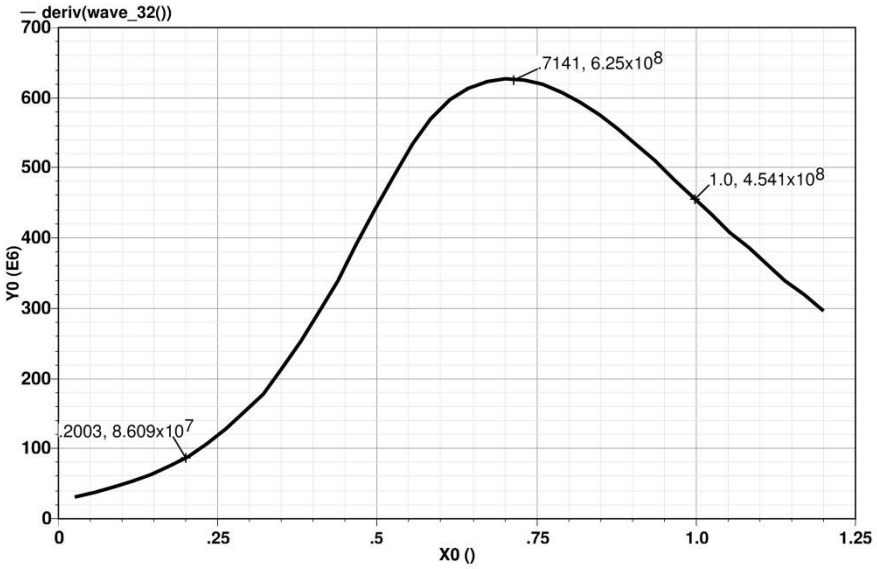


Figure 4.31: The sensitivity of the *NMOS* top-biased *LC VCO*.

Figure 4.32 presents the power dissipation of the top-biased *VCO (-28.91dB≈1.2mW)*. This result is in confirmation with our initial design.

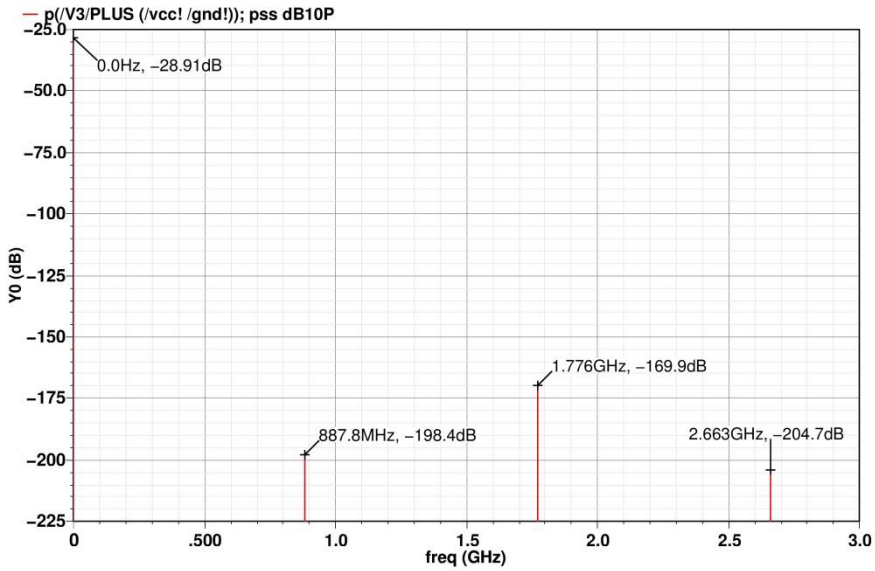


Figure 4.32: The power dissipation of the top-biased *LC VCO*.

The power level of the output signal oscillating at frequency *888MHz* is *-4.05dBm* as expressed in Figure 4.33.

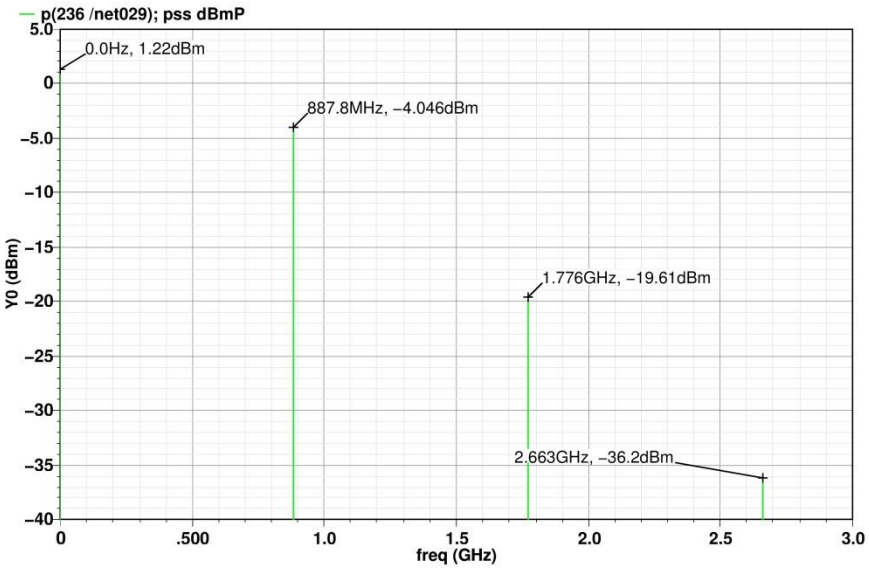


Figure 4.33: The signal power level at *888MHz* operating frequency.

Figure 4.34 shows the output phase noise at different offset frequencies from the carrier frequency (*888MHz*). The phase Noise was found to be *-109.4dBc/Hz* at *1MHz* offset frequency which is much worse than the phase noise of the tail-biased topology (worse by *13dB* for the same power dissipation).

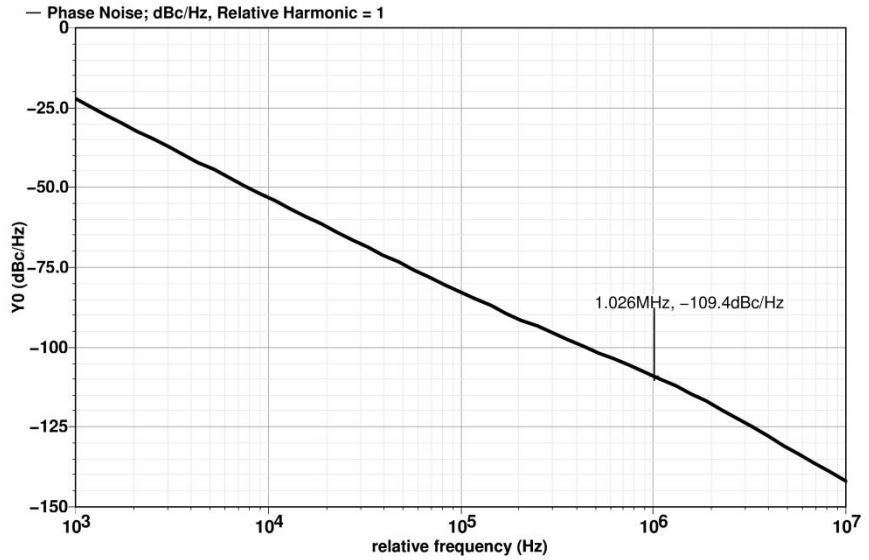


Figure 4.34: Output Phase Noise at different offset frequencies from a carrier frequency at *888MHz*.

It is evident from Figure 4.35 that the output phase noise certainly exceeds *-120dBc/Hz* for all the frequencies of operation. This proves that this topology has a much worse phase noise than its tail-biased counterpart by at least a factor of *13dB*.

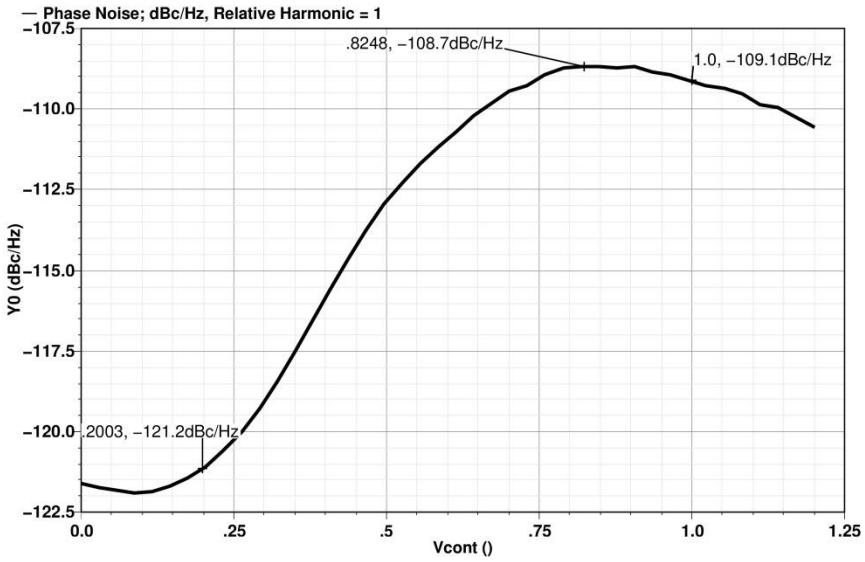


Figure 4.35: Phase Noise at *1MHz* offset frequency for all the frequencies of operation.

In this *VCO*, changing the supply voltage does not a strong effect on the frequency of operation compared to the tail-biased one. For instance, at V_{DD} equals to *1.2V* the resulting output frequency is *888MHz*. As V_{DD} increases the resulting output frequency decreases only slightly and vice versa. As mentioned before, this phenomenon is called frequency pushing and is illustrated in Figure 4.36.

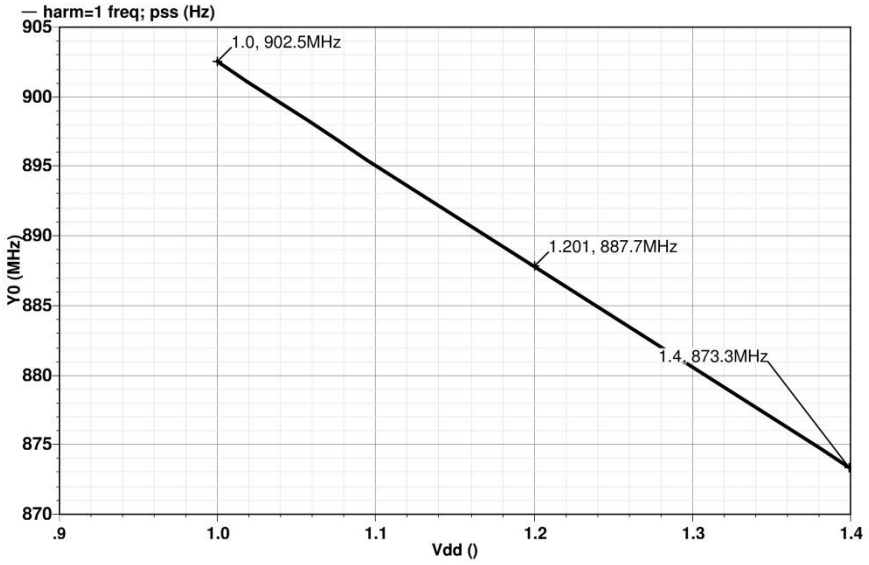


Figure 4.36: The effect of frequency pushing on a carrier frequency at *888MHz*.

4.4.2.3 Findings:

It is obvious that the *NMOS* tail-biased cross-coupled *LC VCO* topology depicted in Figure 4.26 provides a much wider tuning range compared to its tail-biased counterpart. However, it is not a robust topology as it suffers from a major drawback which is having an extremely high level of phase noise compared to the tail-biased one. This is caused by the

modulation of the output common-mode level (varactors' voltage) by the noise found in the current of transistors M_1 and M_2. In other words, this topology suffers from a much higher varactor voltage modulation which results in phase noise.

This modulating effect is prevented in the tail-biased topology because the output common-mode level is stuck by design at V_{DD}. Consequently, we prefer to go back to the tail-biased topology to avoid this modulating effect. In the following section, we propose a *PMOS* tail-biased topology that has a much lower phase noise compared to its *NMOS* counterpart for the same power dissipation.

4.4.3 A PMOS Tail-Biased Cross-Coupled LC VCO:

Many applications require a *VCO* with low phase noise. In this section, we will analyze the performance of a *PMOS* tail-biased cross-coupled *LC VCO* (shown in Figure 4.37). This topology has a much higher noise performance compared to the corresponding *NMOS* topology.

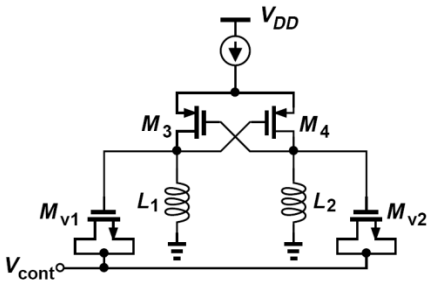


Figure 4.37: A typical *PMOS* tail-biased cross-coupled *LC VCO*.

This topology reduces the phase noise due to flicker noise by replacing the *NMOS* transistors in the topology depicted in Figure 4.15 by *PMOS* transistors (that have substantially less flicker noise). The phase noise of this topology is typically *5dB* to *10dB* lower than that of the *NMOS*.

The startup condition of this topology is exactly the same as that of the tail-biased one and is directly calculated from Equation 4.8. The same applies for the frequency of oscillation (Equation 4.11) and the output voltage swing (Equation 4.9).

4.4.3.1 Design Procedures:

In the following, we are going to design a *VCO* that follows the topology in Figure 4.37 and demonstrate the simulation results obtained for this design. We start by designing a *VCO* that can cover only the upper range of frequency operation (*700MHz* to *850MHz*) using *1.2V* supply and only quarter of the power dissipation of that of the *NMOS* topology (*0.3mW*). We also require the output swing to be only from *-0.15V* to *0.15V* for the *1.2V* supply so as not to stress the transistors. The design procedure is as follows:

1. Based on the given power dissipation budget, the maximum allowable I_{SS} is *250µA*. Thus, to achieve a voltage swing of *0.3V*, R_P is calculated from Equation 4.9 giving $R_P = \frac{V_{swing}}{\frac{4}{\pi} I_{SS}} = 942\Omega$.
2. Taking this value of R_P, determine the smallest inductor value that gives this parallel resistance at resonance. It will also be the inductor with the highest quality factor. In our design, this inductance value was found to be *5.9nF*.
3. Estimate the dimensions of M_1 and M_2 that achieve nearly complete current steering. This achieves the required voltage swing and satisfies the startup condition ($(g_m R_P)^2 > 1$ *giving* $g_m > 1.1m\Omega^{-1}$). We chose g_m to be more than double this value ($2.75m\Omega^{-1}$) to account for temperature and process variations. M_1 and M_2 are also chosen to have the minimum channel length to minimize their capacitance contributions to C_1.
4. Calculate $C_{VAR,\ MIN}$ that allows the oscillator to reach the upper end of the tuning range (*850MHz*). From Equation 4.11, $C_{VAR,\ MIN}$ was found to be *2.1pF*.
5. Using appropriate varactor models, determine the lower end of the tuning range that occurs at $C_{VAR,\ MAX}$ ($C_{VAR,\ MAX} = 5.2pF$). It was found that this value is *658MHz*.
6. Apply some fine adjustments to modify the tuning range to achieve our required range of frequency operation (*700MHz* to *850MHz*).

This design procedure achieves the maximum possible tuning range for a given range of frequency operation, power dissipation and output voltage swing.

4.4.3.2 Simulation Results:

Figure 4.38 demonstrates the tuning curve achieved by this *VCO*. For the charge-pump designed in 1 with V_{CONT} from *0.2V* to *1V*, the tuning range available by this *VCO* is from *685MHz* to *860MHz*. This tuning range is considered to be moderate and it is wider than that obtained by the *NMOS* topology.

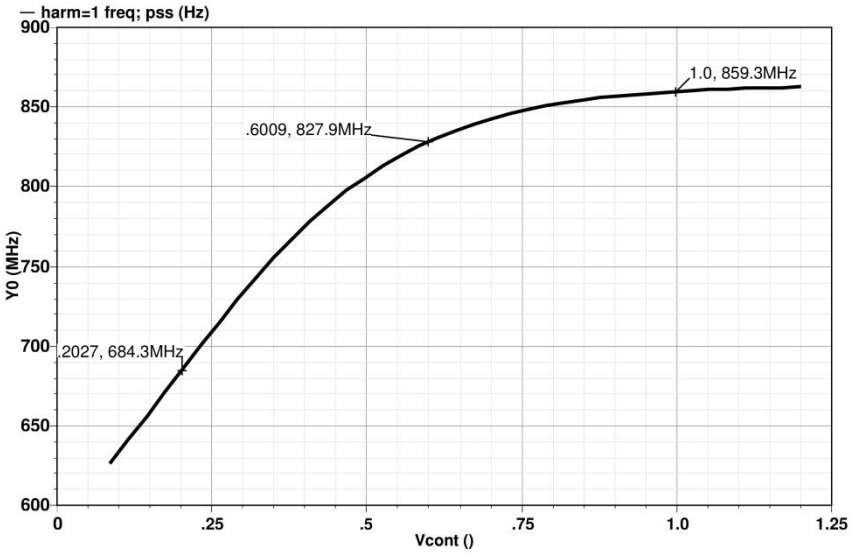


Figure 4.38: The tuning range of a *PMOS* tail-biased cross-coupled *LC* VCO.

Figure 4.39 confirms that the obtained single ended peak-to-peak output swing is indeed *0.3V* as expected from our design procedures. It is clear that this *VCO* provides an output Common-Mode level of *0* which could be an advantage or a disadvantage depending on the next stage.

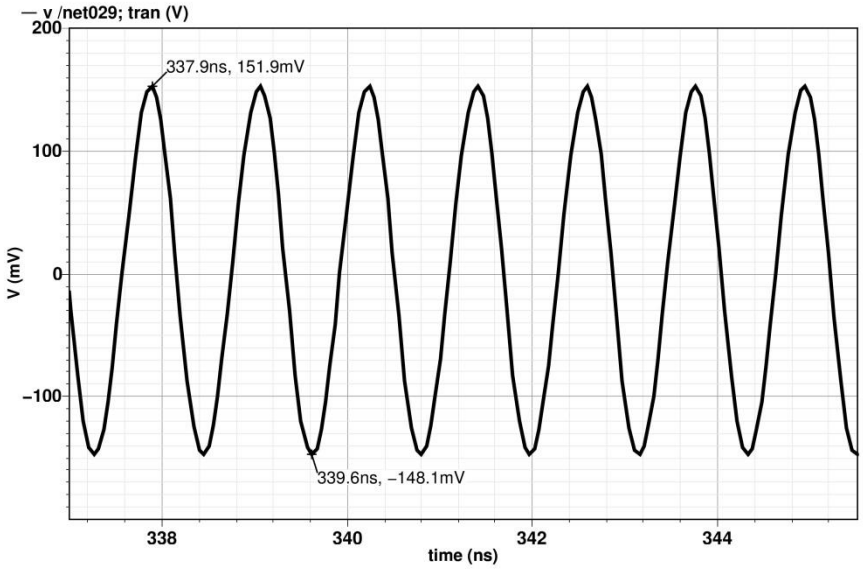


Figure 4.39: The single ended output swing obtained by the *PMOS* tail-biased *LC VCO*.

On the other hand, Figure 4.40 depicts the differential output swing obtained. It is clear that the generated signal has a clean sinusoidal waveform. This signal is readily applied to a buffer or an inverter stage to obtain a square-like waveform with *50%* duty cycle.

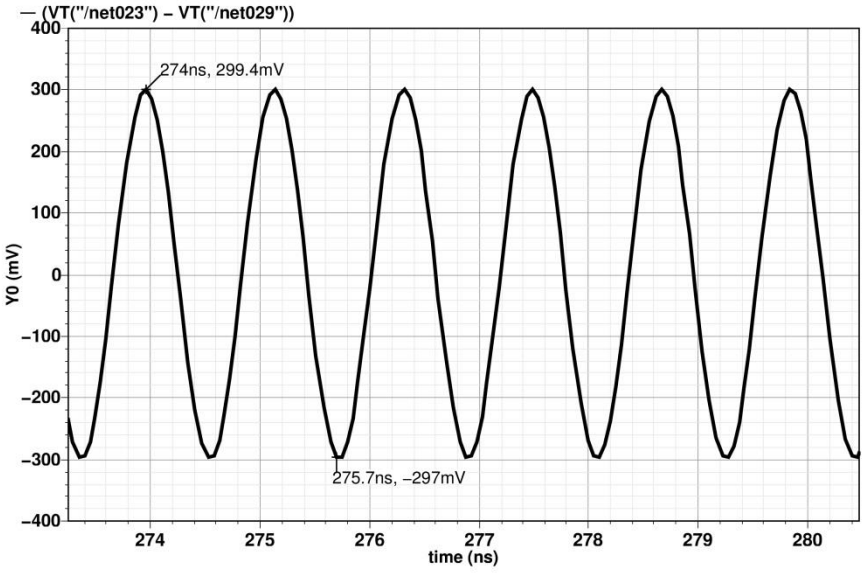


Figure 4.40: The differential output swing obtained by the *PMOS* tail-biased *LC VCO*

As for the sensitivity (K_{VCO}) of the designed *VCO*, as evident by Figure 4.41, it is monotonic and varies by only a single order of magnitude across the tuning range. This behavior is near

the desirable behavior required by *PLLs* as the sensitivity is preferred to be almost constant throughout the tuning range.

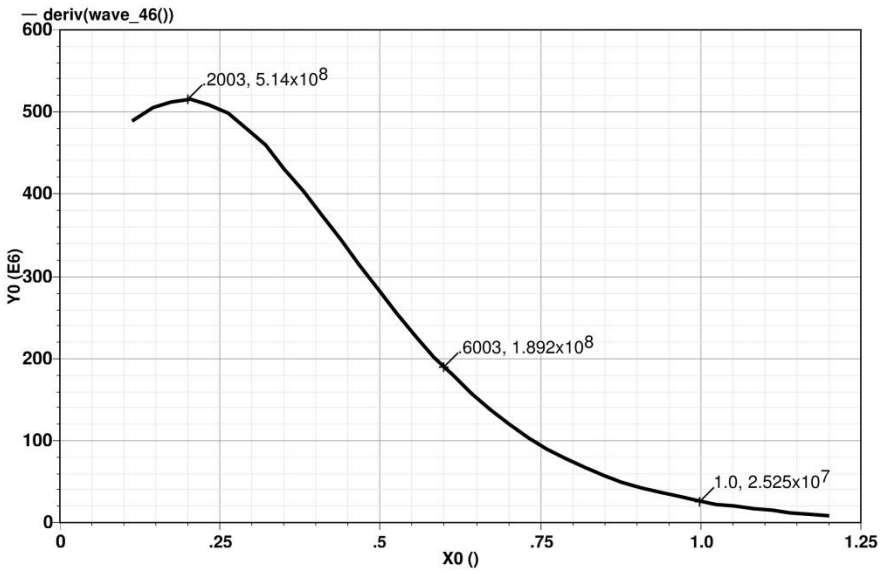


Figure 4.41: The sensitivity of the PMOS tail-biased LC VCO.

Figure 4.42 presents the power dissipation of the *PMOS* tail-biased *VCO (-34,77dB ≈ 0.3mW)* which is only quarter of the power dissipation of its *NMOS* counterpart. This result is in confirmation with our initial design.

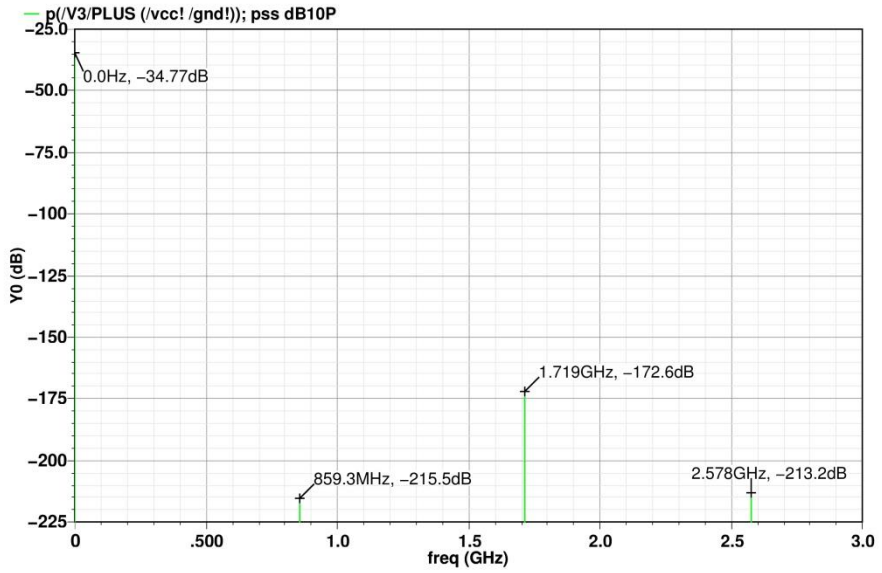


Figure 4.42: The power dissipation of the *PMOS* tail-biased *VCO*.

The power level of the output signal oscillating at frequency *860MHz* is *-22.18dBm* as expressed in Figure 4.43.

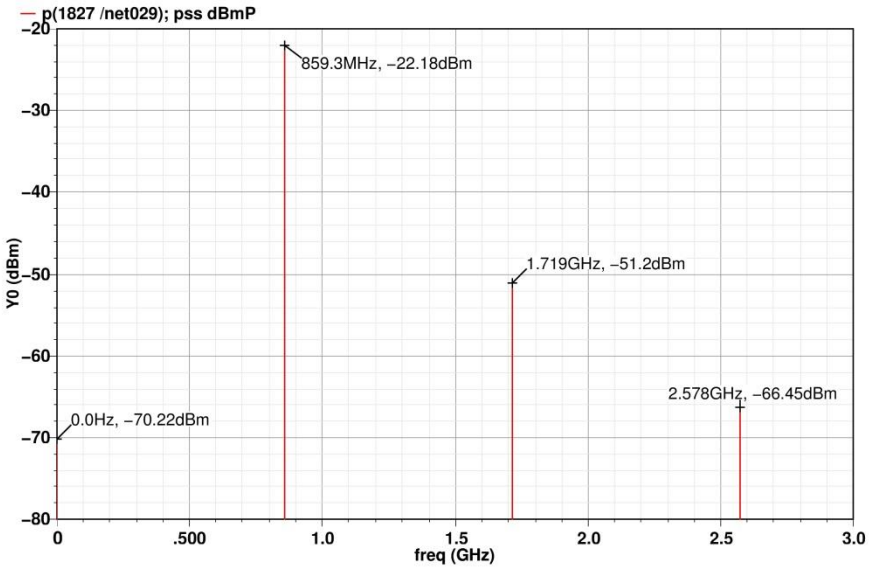


Figure 4.43: The signal power level at *860MHz* operating frequency.

Figure 4.44 shows the output phase noise at different offset frequencies from the carrier frequency (*860MHz*). The phase Noise was found to be *-123.8dBc/Hz* at *1MHz* offset frequency which is considered to be very low considering the fact that only quarter of the power dissipation of the *NMOS* topology is used to achieve a better performance against phase noise.

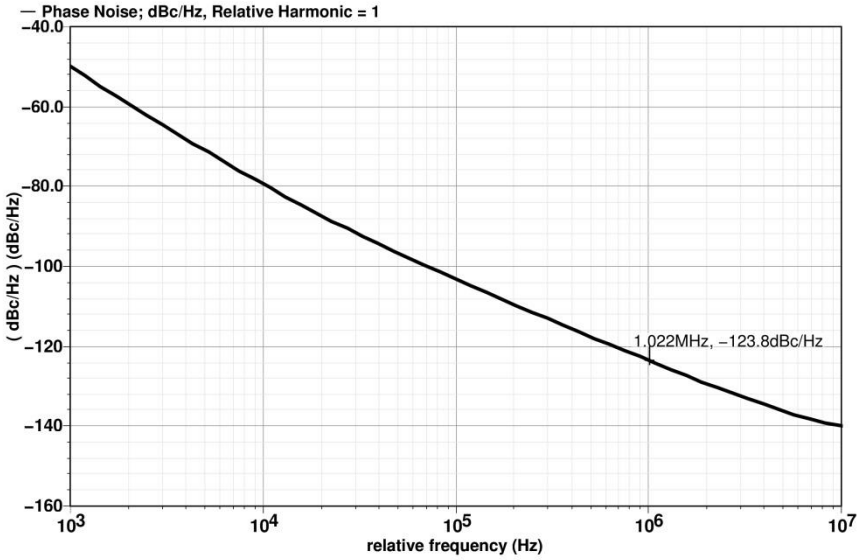


Figure 4.44: Output Phase Noise at different offset frequencies from a carrier frequency at *860MHz*.

It is evident from Figure 4.45 that the output phase noise does not exceed *-120dBc/Hz* for most of the frequencies of operation.

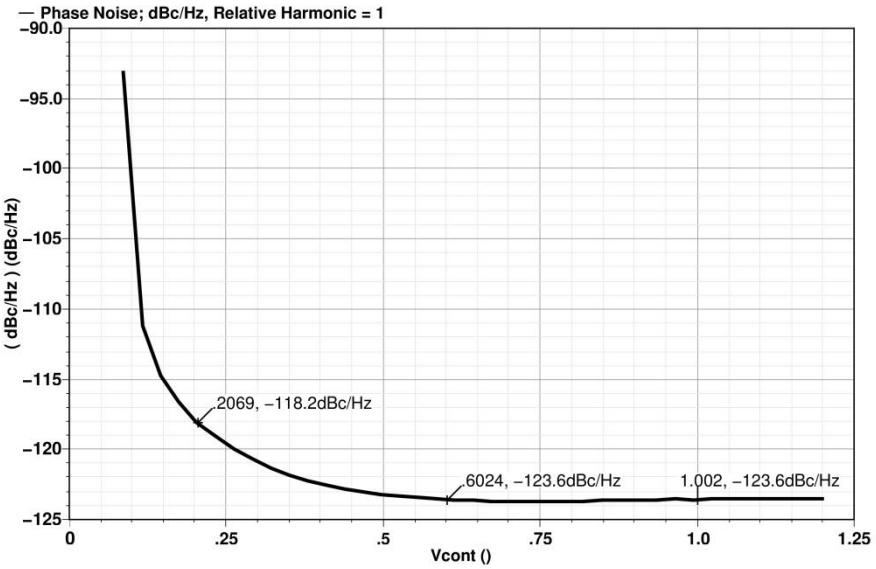


Figure 4.45: Phase Noise at *1MHz* offset frequency for all the frequencies of operation.

In this VCO, changing the supply voltage has almost no effect on the frequency of operation. For instance, at V_{DD} equals to *1.2V* the resulting output frequency is *860MHz*. As V_{DD} decreases the resulting output frequency decreases by only *3MHz* whereas the NMOS topology increased by *86MHz*. This low frequency pushing is caused by the fact that the output common-mode level is 0V (not V_{DD} or $V_{DD}/2$) which means that the supply voltage does not directly modulate the voltage across the varactors as in the previous topologies. As mentioned earlier, this phenomenon is called frequency pushing and is illustrated for the *PMOS* topology in Figure 4.46.

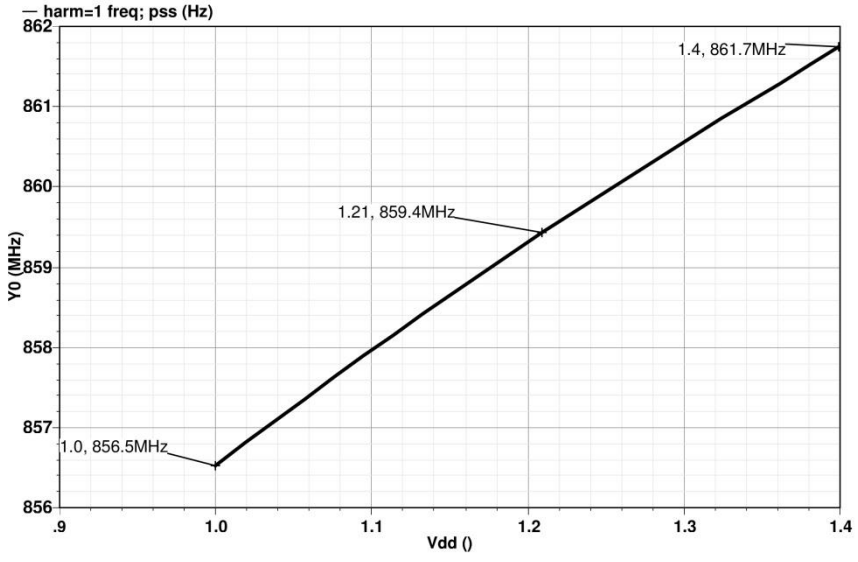


Figure 4.46: The effect of frequency pushing on a carrier frequency at *860MHz*.

It is important to highlight that the tuning range can be expanded by using a *PMOS* bottom-biased topology (illustrated in Figure 4.47). However, this topology suffers from an extremely high level of phase noise compared to the tail-biased one. This is caused by the

modulation of the output common-mode level (varactors' voltage) by the noise found in the current of transistors M_1 and M_2. In other words, this topology suffers from a much higher varactor voltage modulation which results in phase noise. Therefore the tail-biased topology is preferred over the bottom-biased one.

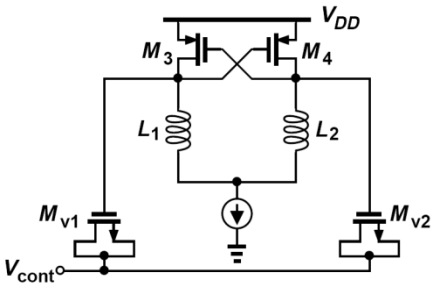


Figure 4.47: A typical Bottom-Biased Cross-Coupled *LC VCO*.

4.4.3.3 Findings:

It is clear from the ongoing discussion that the *PMOS* tail-biased cross-coupled *LC VCO* topology depicted in Figure 4.37 is more robust than the *NMOS* topology. However, it suffers from a principle drawback which is having a limited speed. This topology incorporates *PMOS* transistors that possess much lower mobility than *NMOS*. This issue is only critical at frequencies above *10GHz* which is well beyond our operating frequency. In the following section, we summarize the findings and results we have obtained from the simulations.

4.4.4 Summary of the LC VCO simulation results:

Table 4.1 provides a complete summary of the simulation results of the three presented *VCO* topologies. It is clear that the *PMOS* topology (depicted in Figure 4.37) is an optimization between the other two topologies with only quarter of their power dissipation.

Table 4.1: Comparison between the *NMOS* tail-biased, *NMOS* top-biased and *PMOS* tail-biased *LC VCOs*.

	***NMOS* Tail-Biased**	***NMOS* Top-Biased**	***PMOS* Tail-Biased**
Tuning Range	78MHz	356MHz	175MHz
Differential Peak Swing	790mV	866mV	300mV
Max Sensitivity	3.075X10^{8}Hz/V	6.25X10^{8}Hz/V	5.14X10^{8}Hz/V
Power Dissipation	1.2mW	1.2mW	0.3mW
Signal Level	-6.06dBm	-4.05dBm	-22.18dBm
Output Phase Noise *@1MHz* offset (worst case)	-120.9dBc/Hz	-108.7dBc/Hz	-118dBc/Hz
Frequency Pushing (Max to Min frequency)	132MHz	29MHz	5.2MHz
Figure of Merit (*FOM*)	187.95dB	188.94dB	198.09dB

where Figure of Merit is calculated as

$$FOM = \frac{(Tuning\ Range)^2}{Power\ Dissipation\ \times Phase\ Noise\ \times (Offset\ Frequency)^2} \qquad \text{Equation 4.12}$$

To cover the entire frequency range of operation (*50MHz* to *850MHz*), there are basically two approaches. We could incorporate multiple *LC VCOs* and allow them to work in parallel or we could propose other highly tunable *VCO* topologies that could cover the whole range. In the next section, we analyze the feasibility of both approaches and demonstrate the need for Ring Oscillators.

4.4.5 Discrete Tuning LC VCOs using Switched Capacitors:

The first approach to be able to cover the entire range of frequency operation is to design multiple *VCOs* according to the design procedure expressed in section 4.4.3 where each of them covers a small part of the whole range of operation and allow them to work in parallel. These multiple *VCOs* working in parallel dissipate a high level of power. In our case, if we optimistically consider that each *VCO* has a tunability of *1:1.5*, then it is required to have at least eight *VCOs* working in parallel (to provide sufficient overlapping between the different ranges). This means that if each of them had a power dissipation of *0.3mW*, the combination will require a minimum of *3mW* (including the additional switching circuit). Doing so not only draws more power but also increases the chip area.

Also the design of the switching circuit is very complex because the *VCO* is required to be used in a phase-locked-loop. This means that complex switching algorithms have to be used so as not to disturb the operation of the *PLL*. Another crucial drawback of this approach is the extensive use of inductors which occupy a large size and may lead to an enormously large *VCO* design. All of these drawbacks suggest that we look for other highly tunable topologies that could cover the entire range of frequency operation with a single *VCO*.

One famous topology that has a high level of tunability is depicted in Figure 4.48. This topology can expand the capacitance range beyond C_{MAX} and C_{MIN} of the varactors by the use of switched capacitors. The idea is simply to place a bank of switched capacitors (with capacitance C_U) in parallel with the *LC* pair and switch them in and out as required to change the resonance frequency.

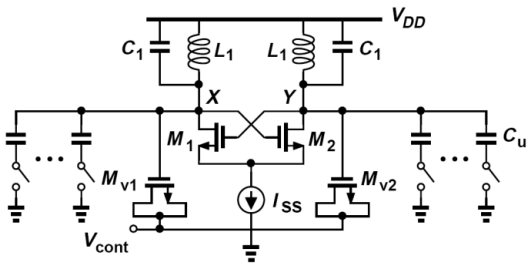


Figure 4.48: A typical discrete tuning *VCO* using switched capacitors.

This topology is considered to have to levels of control; a fine control achieved by the control voltage (V_{CONT}) and a coarse control achieved by the digital input to the capacitors bank (requiring an additional control circuit). Figure 4.49 demonstrates the two types of tuning available. It is obvious that the fine tuning covers a continuous but narrow tuning range whereas the coarse tuning shifts the entire characteristic upwards or downwards.

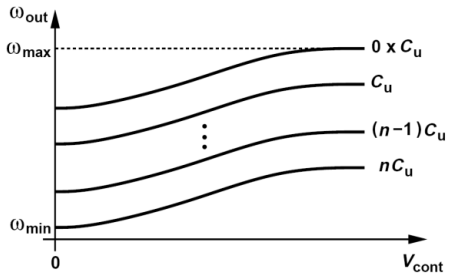


Figure 4.49: Fine and coarse tuning in a discrete tuning *VCO*.

Following Equation 4.11, the lowest possible frequency is reached when all the capacitors are switched in and the varactor has its maximum capacitance yielding an oscillation frequency of

$$\omega_{MIN} = \frac{1}{\sqrt{L_1(C_1 + C_{VAR,MAX} + nC_U)}}. \qquad \text{Equation 4.13}$$

On the other hand, the highest possible frequency is reached when the entire capacitance bank is switched out and the varactor has its minimum capacitance yielding a frequency of operation of

$$\omega_{MAX} = \frac{1}{\sqrt{L_1(C_1 + C_{VAR,MIN})}}. \qquad \text{Equation 4.14}$$

This discrete tuning *VCO* suffers from several issues such as the on-resistance of the switches (R_{ON}) degrades the quality factor of the *LC* pair which leads to a much higher phase noise. It is important to highlight that R_{ON} cannot be reduced by simply using wider switches because wider switches introduce a larger off-capacitance to the ground which limits the tuning range significantly.

Another very important issue in discrete tuning is the possibility of having "blind zones" i.e. a range of frequency that cannot be covered by the *VCO* for any combination of coarse and fine controls. Blind zones are avoided by having some overlap between successive tuning characteristics which requires a smaller value of C_U and larger bank of capacitors. This overlap also prevents the sensitivity of the *VCO* from varying significantly near the ends of the tuning curve.

In the following, we will present the discrete tuning *VCO* (that covers the entire range of the digital *TV* Tuner system) developed by Yu-Che Yang and Shey-Shi Lu[1].

[1] Yu-Che Yang and Shey-Shi Lu, "A Single-*VCO* Fractional-*N* Frequency Synthesizer for Digital *TV* Tuners", *IEEE Transactions on Industrial Electronic*, VOL. 57, No. 9, September 2010, pp. 3207-15.

4.4.5.1 Sample Design:

To achieve a better phase noise performance over the wide tuning range a discrete tunable *LC VCO* is designed to oscillate at several *GHz* (divided later by a frequency divider to generate the required frequency). This *VCO* has a tuning range from *2300MHz* to *3900MHz*. Thus, the topology of Figure 4.15 is preferred as it is faster than its *PMOS* counterpart and this is critical because the frequency of operation is relatively high.

Figure 4.50 demonstrates this *VCO* where coarse tuning is achieved by virtue of a *5-bit* binary switched capacitors array with an Auto-frequency calibration (*AFC*) loop that automatically chooses the coarse setting of the capacitors.

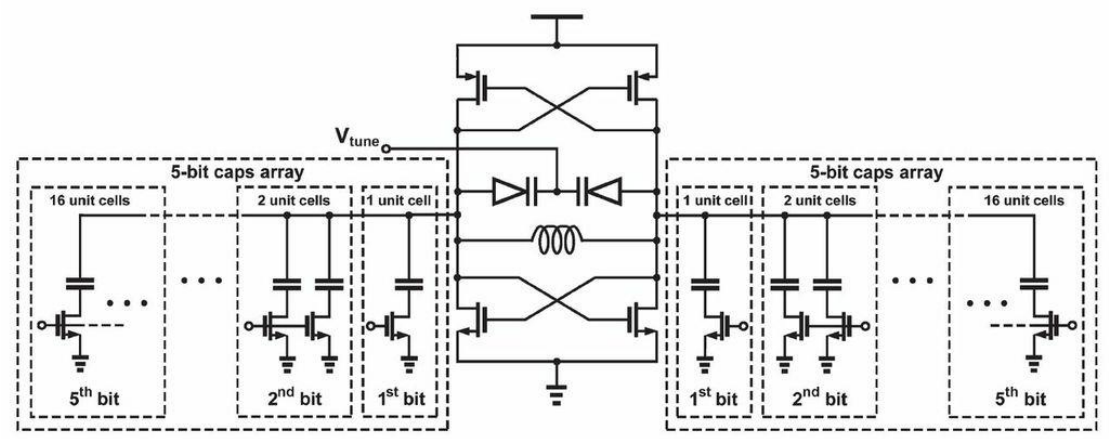


Figure 4.50: A *5-bit* switched capacitors array Discrete Tuning *LC VCO*.

The tail current source has been removed completely from this topology because it is the dominant source of flicker noise and to leave more headroom, thus, nearly rail-to-rail voltage swing is possible. However, removing the tail current source makes the *VCO* more sensitive to variations in the supply voltage which leads to a bad frequency pushing performance of the *VCO*. This problem can be solved by using a separate supply for the *VCO* but this solution is very expensive.

As for the capacitors array, it is made up of small capacitor unit cells (*114fF* each) and *NMOS* switches (*4µm/130nm*). Figure 4.51 presents both the coarse and fine tuning curves of this *VCO*. The *VCO* frequency has a tuning range from *2300MHz* to *4400MHz* and the *VCO* gain is smaller than *140MHz/V*.

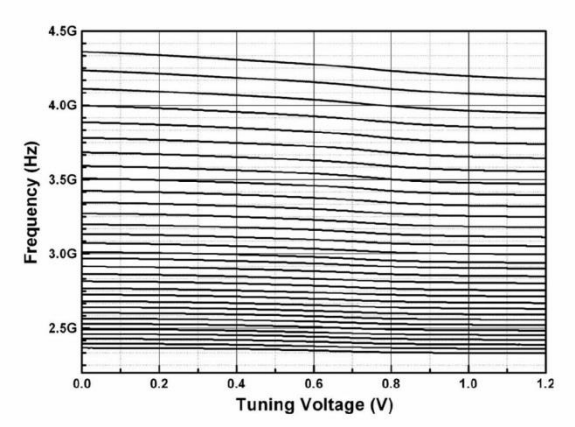


Figure 4.51: Coarse and fine tuning curves of the Discrete Tuning *VCO*.

As for the dynamic *AFC*, it automatically chooses the capacitors array setting according to the desired output frequency of the synthesizer and selects the appropriate tuning curve for the *PLL*. In addition, the *AFC* can calibrate the *VCO* frequency error caused by variations in process, temperature and supply voltage.

To conclude, this single-*VCO* for Digital *TV* Tuners can cover the entire range of frequency operation (*48MHz* to *862MHz*). Since the system has only one *VCO*, the power consumption and the chip area are reduced greatly compared to using multiple *VCOs* to cover the entire frequency range. The entire synthesizer draws *14mA* from a *1.2V* supply. As for the phase noise, it is lower than *-100dBc/Hz* at *1MHz* frequency offset.

It is clear that using discrete tuning, the entire range of frequency operation of the Digital *TV* Tuner system can be covered with a single *VCO*. However, there are several issues that limit the use of discrete tuning in *VCOs* such as the on-resistance of the switches, blind zones, complex capacitor arrays, complex *AFCs*, high power consumption, variable *VCO* gain and higher phase noise. Therefore, in the following section, we will discuss ring oscillators as viable topologies to be used in the *TV* Tuner Direct Conversion Receiver for their inherently wide tuning range.

4.5 Ring Oscillators:

Ring oscillators can typically provide a much wider tuning range than their *LC* counter parts. Consequently, in applications where the frequency of operation must be varied by orders of magnitude (such as the *TV* Tuner Receiver), Ring oscillators are the direct choice. In general, a ring oscillator consists of multiple amplifier stages connected together in a loop. Let's start by discussing the behavior of a single-stage common source amplifier in a feedback structure.

Figure 4.52 illustrates a single-stage common source amplifier with feedback. The open-loop transfer function of this structure has only one pole which can provide at most a frequency-dependent shift of 90^O (as frequency approaches infinity). The common-source structure provides an additional *DC* phase shift of 180^O resulting in a maximum phase shift of 270^O. Thus, the loop fails to satisfy Barkhausen's criteria (Equation 4.2 and Equation 4.3) at any frequency and no oscillation can occur.

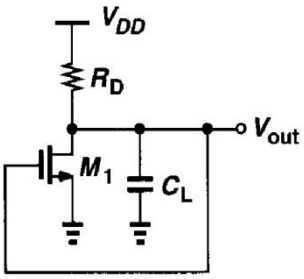


Figure 4.52: A single-stage common-source amplifier in a feedback loop.

The previous discussion suggests that oscillation can occur if we increase the number of amplifier stages in the loop, which increases the number of poles. Figure 4.53 demonstrates the feedback loop with two stages. This configuration introduces two significant poles allowing the frequency-dependent phase shift to approach 180^O (at infinite frequency).

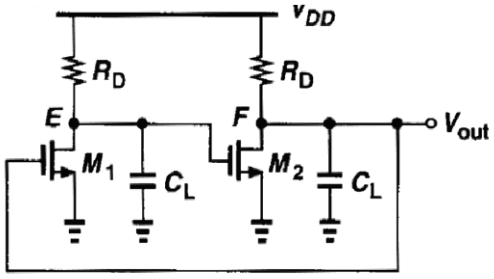


Figure 4.53: Two common-source amplifiers in a feedback loop.

However, this configuration does not oscillate because it experiences a positive feedback at zero frequency and the circuit simply "latches up". For example, if V_E is high, V_F becomes low forcing V_E to remain high and the circuit remains in this state forever. Figure 4.54 illustrates another reason that prevents this circuit from oscillating which is having a frequency-dependent phase shift of 180^O only at infinite frequency where the gain becomes significantly low. Therefore, both of Barkhausen criteria are not satisfied.

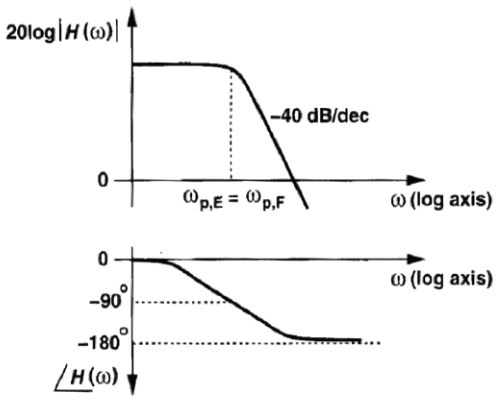


Figure 4.54: Bode Plots of the two-pole system.

It is clear that to allow and sustain oscillation, a third stage must be added to the feedback loop. This results in a greater phase shift around the loop caused by the additional pole.

Figure 4.55 presents a topology that has three identical amplifier stages. In this circuit, the frequency-dependent phase shift reaches 180^O at frequencies much lower than infinity where the gain is sufficiently high and both of Barkhausen's criteria are satisfied. This circuit is an example of Ring oscillators.

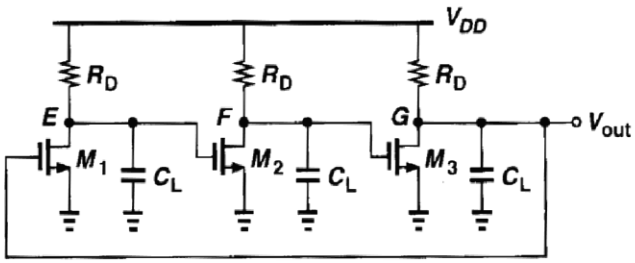


Figure 4.55: Three common-source amplifiers in a feedback loop.

To determine the oscillation frequency and the minimum required voltage gain per stage, we have to derive the open-loop transfer function first. For three identical amplifier stages each with a transfer function of $-A_O/(1+s/\omega_O)$, the open-loop gain is given by,

$$H(s) = -\frac{A_O^3}{\left(1+\frac{s}{\omega_O}\right)^3}. \qquad \text{Equation 4.15}$$

To achieve Barkhausen's phase criteria (Equation 4.3), each stage must contribute a frequency-dependent phase shift of 60^O. This occurs at the frequency of oscillation ω_{OSC} given by

$$\tan^{-1}\frac{\omega_{OSC}}{\omega_O} = 60^O\,. \qquad \text{Equation 4.16}$$

Thus,

$$\omega_{OSC} = \sqrt{3}\,\omega_O\,. \qquad \text{Equation 4.17}$$

As for the minimum voltage gain per stage required to satisfy Barkhausen's gain criteria (Equation 4.2), it is derived as follows

$$|H(s)| = \frac{A_O^3}{\left[\sqrt{1+\left(\frac{\omega_{OSC}}{\omega_O}\right)}\right]^3} = 1 \qquad \text{Equation 4.18}$$

resulting in

$$A_O = 2\,. \qquad \text{Equation 4.19}$$

To summarize, the three-stage ring oscillator (depicted in Figure 4.55) requires a low frequency gain of at least *2* and it oscillates at a frequency of $\sqrt{3}\omega_O$ where ω_O is the *3-dB* bandwidth (pole) of each stage. However if the low frequency gain is smaller than *2*, the circuit inevitably fails to oscillate. On the other hand, if the *DC* gain is greater than *2*, the oscillation amplitude increases at a very fast rate. After a while, a maximum amplitude is reached due to the nonlinearities and saturating effects present in the circuit.

As for the phase shift around the loop, each amplifier stage contributes a frequency-dependent phase shift of 60^O (Equation 4.16) in addition to a *DC* phase shift of 180^O, resulting in a total phase shift of 240^O per stage. The resulting waveforms at the three nodes are expressed in Figure 4.56. This ability to generate an oscillating output with multiple phases is one of the main advantages of ring oscillators.

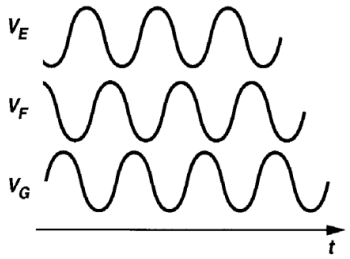


Figure 4.56: Waveforms at the three nodes of a three-stage ring oscillator.

The three-stage ring oscillator can also be implemented using the differential topology illustrated in Figure 4.57. If the low frequency gain per stage is above *2*, the amplitude of oscillation grows until each stage experiences complete current steering. Consequently, the output voltage swing at each node is $I_{SS}R_1$.

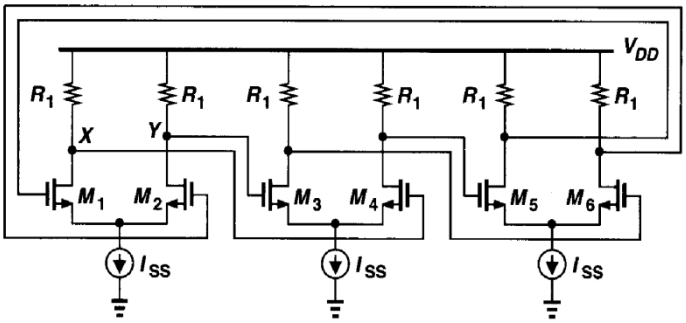


Figure 4.57: A three-stage differential ring oscillator.

Another approach to calculate the frequency of oscillation is to use a large-signal analysis which takes the nonlinear effects of the circuit into consideration. Figure 4.58 demonstrates a typical three-stage ring oscillator. In this case (as shown in Figure 4.59), V_X and V_Z start at V_{DD} whereas V_Y starts at *0V*. Since V_Z is high, V_X is forced to become low. This forces V_Y to become high after one inverter delay and V_Z to becomes low after two inverter delays and the circuit will oscillate forever. This generates an oscillating output that has a period of oscillation equals to $6T_D$.

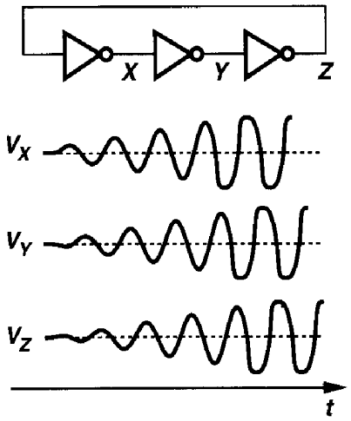


Figure 4.58: A three-stage ring oscillator with inverter stages.

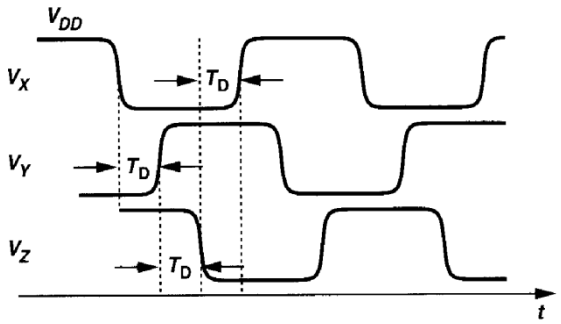


Figure 4.59: Waveforms of a three-stage ring oscillator with V_X and V_Y starting at V_{DD} and V_Y starting at zero.

It is important to clarify that the frequency of oscillation obtained from the small and large signal analyses are not equal. There is no contradiction there and both results are correct because the circuit oscillates at both frequencies but not at the same time. When the circuit is released, the oscillation occurs at the frequency estimated by the small-signal analysis. However, as the amplitude of oscillation grows, the circuit becomes nonlinear and the circuit oscillates at the lower frequency estimated by the large-signal analysis.

In the following, we will attempt to design a tunable three-stage Ring *VCO* for the receiver of our *TV* Tuner system. It achieves tunability by having a tunable *PMOS* load.

4.5.1 A PMOS-Load Three-Stage Ring VCO:

In the last section, we have established that a three-ring topology can sustain oscillations. In this section, we will design a tunable differential oscillator that covers the entire range of the *TV* Tuner frequency operation. Figure 4.60 demonstrates the topology proposed for each differential stage. In this case, the tunability is achieved by using a compound *PMOS* load made up of a *PMOS* transistor operating in the triode region and a diode-connected *PMOS*. The diode-connected *PMOS* transistors (M_3 and M_4) are basically a variable resistance that changes with the applied current (current-controlled resistance). Therefore, by varying I_{SS}, the value of the output resistance is varied. This changes the delay of each stage and hence the frequency of oscillation.

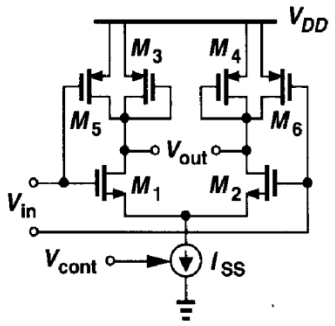


Figure 4.60: A differential stage that achieves a wide tuning range using a compound *PMOS* load.

It is important to highlight that the additional *PMOS* transistors M_5 and M_6 (operating in the triode region) are added to pull each output node to V_{DD}. This minimizes the variations in the output voltage swing throughout the tuning range with large variations in I_{SS}.

In the following, we will analyze the simulation results obtained using Cadence for our design. It is important to state that the startup condition expressed in Equation 4.19 is satisfied by our design of the amplifier depicted in Figure 4.60. This amplifier is used as a replica circuit to construct a three-stage ring *VCO*.

As for the Current (I_{SS}) used in each amplifier stage, it is varied by employing a differential current-folded circuit (depicted in Figure 4.61) which saves valuable headroom. It is simply a differential pair -with V_{CONT} as its input- that drives two current mirrors where one of them is used to generate a variable I_{SS} as required. In the following, we discuss the simulation results of our design of the VCO employing this topology.

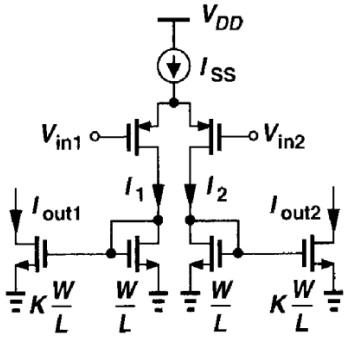


Figure 4.61: The Differential Current-Folding topology used to vary the current in the amplifier stage (I_{SS}).

4.5.1.1 Simulation Results:

Figure 4.62 shows the tuning of the current (I_{SS}) by varying the applied control voltage (V_{CONT}). The figure demonstrates that as V_{CONT} varies form *0.2V* to *1V* (the range of the Charge-Pump as designed in 1, I_{SS} is varied from *12µA* to *422µA*. This means that the power dissipation varies significantly throughout the tuning range. Consequently, we will obtain simulation results for both the maximum and minimum power dissipation as it is expected that the performance will vary significantly between them.

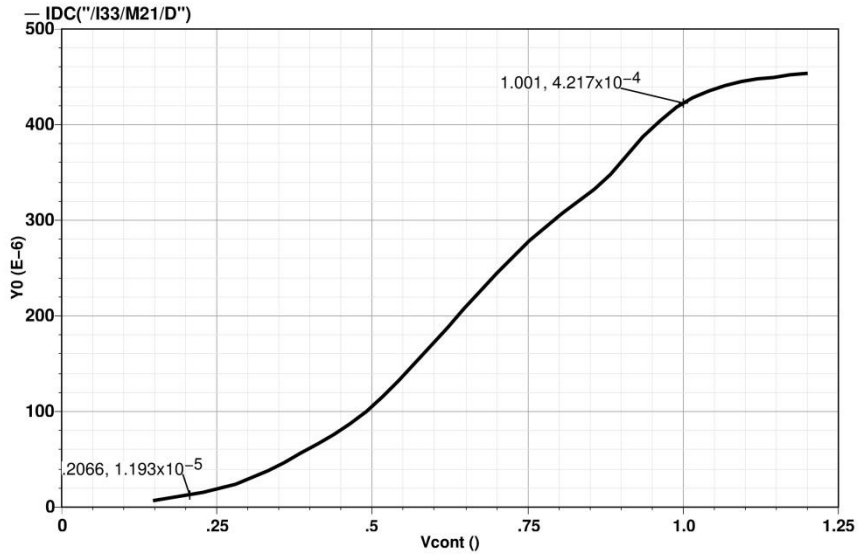


Figure 4.62: The variation of the current in each amplifier stage (I_{SS}) with the control voltage (V_{CONT}).

Figure 4.63 demonstrates the tuning curve achieved by this *VCO*. For the charge-pump designed in 1 with V_{CONT} from *0.2V* to *1V*, the tuning range available by this *VCO* is from *41.61MHz* to *980MHz*. This tuning range is considered to be extremely wide (*1:24*) and it covers the entire range of frequency operation with a single *VCO*.

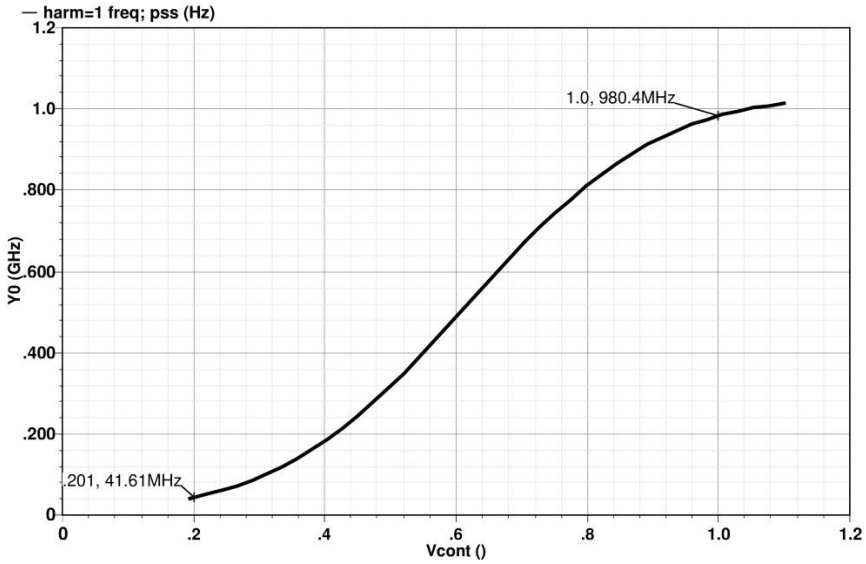


Figure 4.63: The tuning range of the *PMOS*-Load three-stage Ring *VCO*.

Figure 4.64 and Figure 4.65 illustrate that the obtained single ended peak-to-peak output swing is *0.4V* at *41MHz* oscillation frequency and *0.8V* at *980MHz* which demonstrates that the output voltage swing is not constant and is doubled across the tuning range. It is also clear that this *VCO* does not have a constant common-mode level across the tuning range which is considered a major disadvantage.

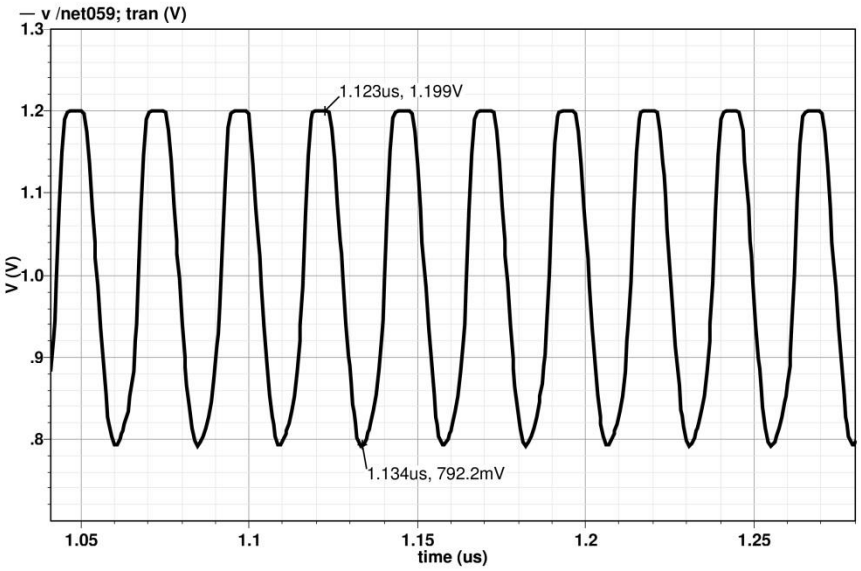


Figure 4.64: The single ended output swing obtained by the *PMOS*-Load three-stage Ring *VCO* at *41MHz* oscillation frequency.

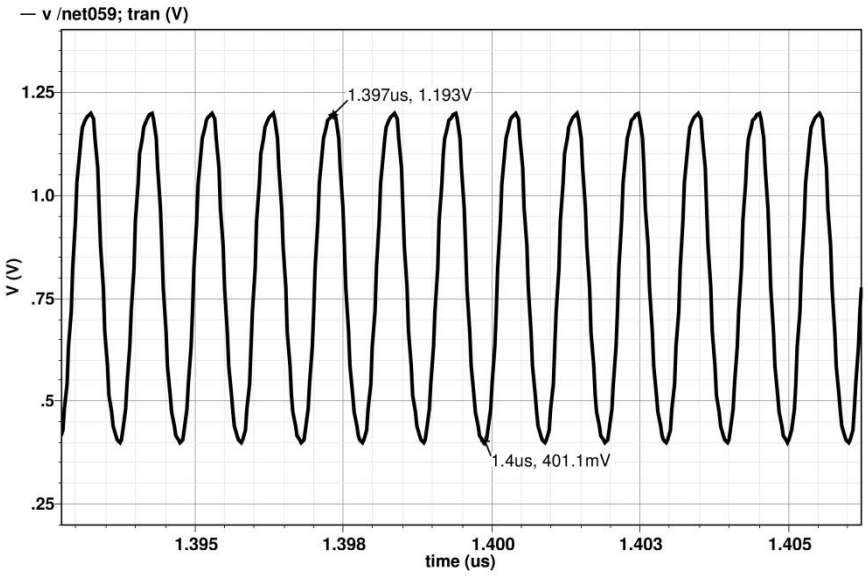


Figure 4.65: The single ended output swing obtained by the *PMOS*-Load three-stage Ring *VCO* at *980MHz* oscillation frequency.

On the other hand, Figure 4.66 and Figure 4.67 depict the differential output swing obtained at both the minimum and maximum power dissipation. It is clear that the generated signal has a clean sinusoidal waveform. This signal is readily applied to a buffer or an inverter stage to obtain a square-like waveform with *50%* duty cycle.

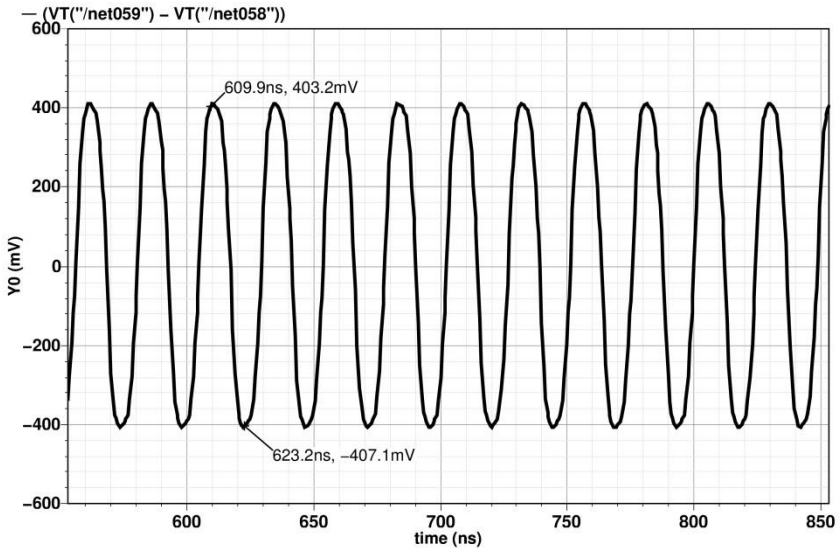


Figure 4.66: The differential output swing obtained by the *PMOS*-Load three-stage Ring *VCO* at *41MHz* oscillation frequency.

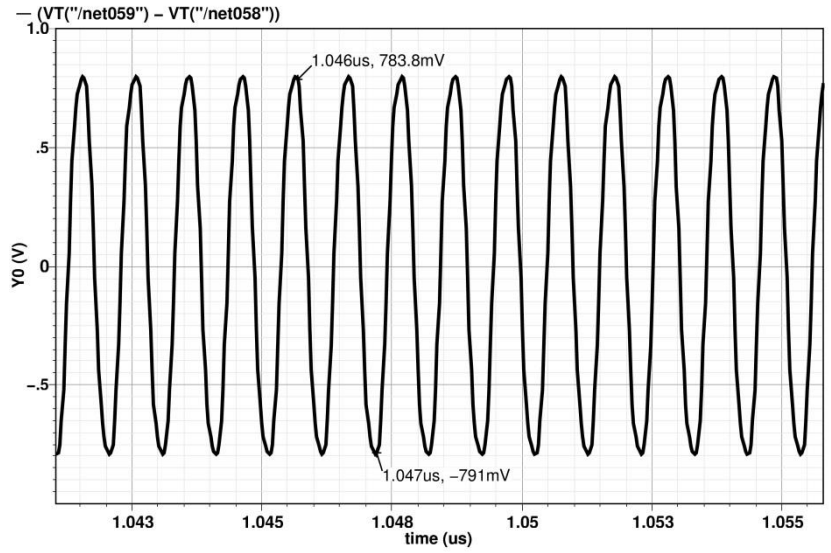


Figure 4.67: The differential output swing obtained by the *PMOS*-Load three-stage Ring *VCO* at *980MHz* oscillation frequency.

As for the sensitivity (K_{VCO}) of the designed *VCO*, as evident by Figure 4.68, it varies by orders of magnitude across the tuning range (increases by a factor of *10* then decreases again by a factor of *10*). This behavior is undesirable in *PLLs* as the sensitivity is required to be almost constant throughout the tuning range.

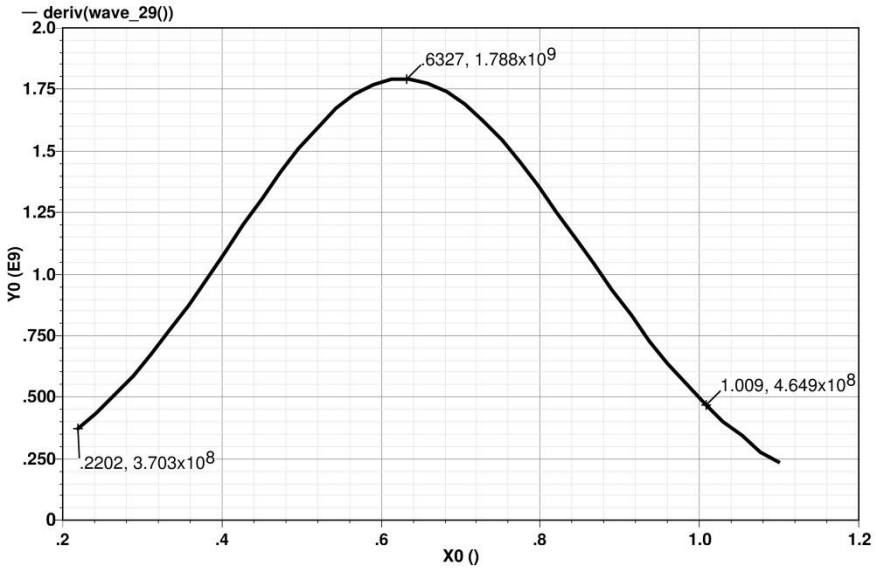


Figure 4.68: The sensitivity of the PMOS-Load three-stage Ring VCO.

Figure 4.69 presents the maximum power dissipation of this *VCO* (*-27.1dB ≈ 1.95mW*) at *980MHz* oscillation frequency. On the other hand, Figure 4.70 presents the minimum power dissipation of this *VCO* (*-41.11dB ≈ 77.45µW*) at *41MHz* oscillation frequency.

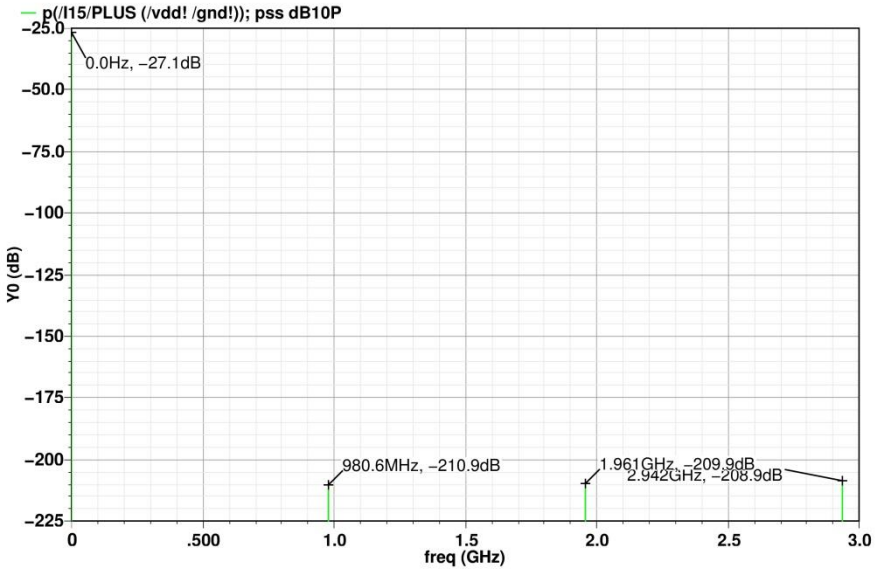


Figure 4.69: The max power dissipation of the *PMOS*-Load three-stage Ring *VCO* at *980MHz* oscillation frequency.

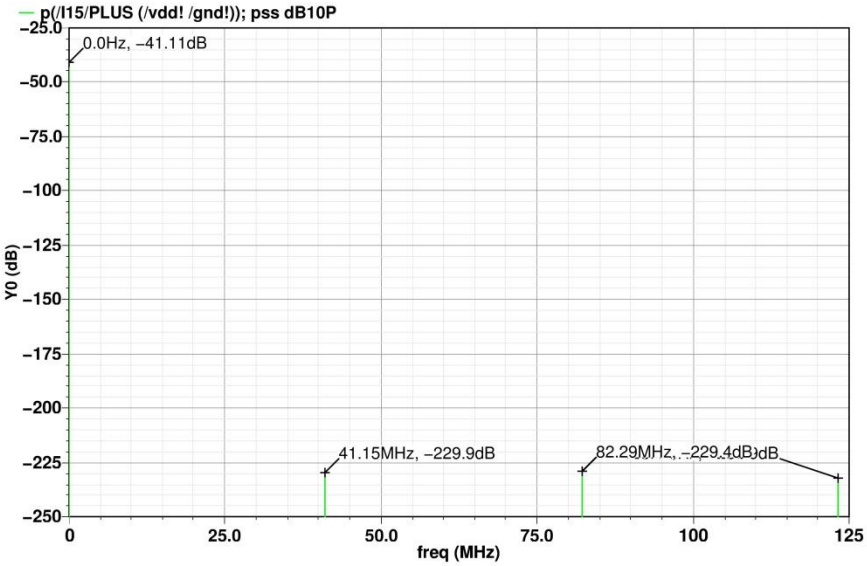


Figure 4.70: The min power dissipation of the *PMOS*-Load three-stage Ring *VCO* at *41MHz* oscillation frequency.

The power level of the output signal oscillating at frequency *41MHz* is *-23.89dBm* as expressed in Figure 4.71. Whereas at frequency *980MHz* the signal power level is *-7.332dBm* as shown in Figure 4.72. It is obvious that there is also a large variation in the signal power level throughout the tuning range.

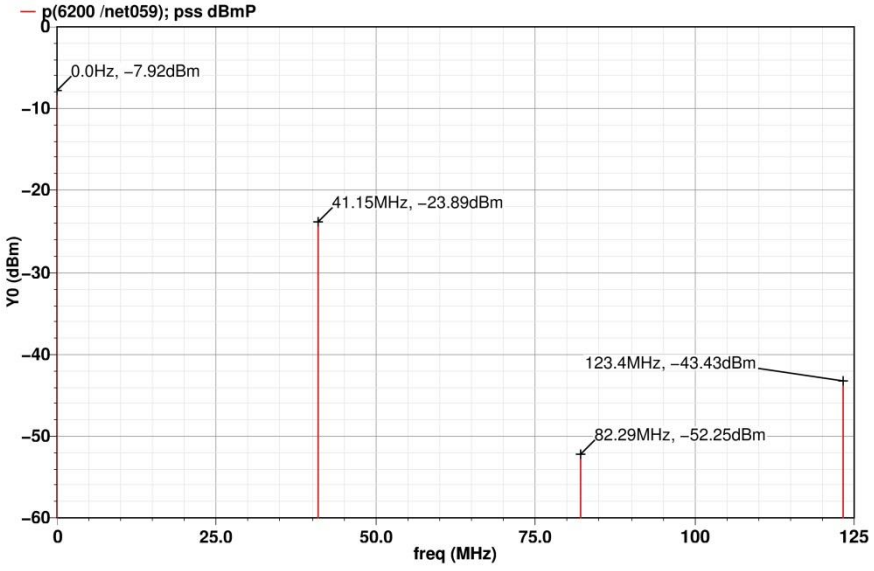


Figure 4.71: The signal power level at *41MHz* operating frequency.

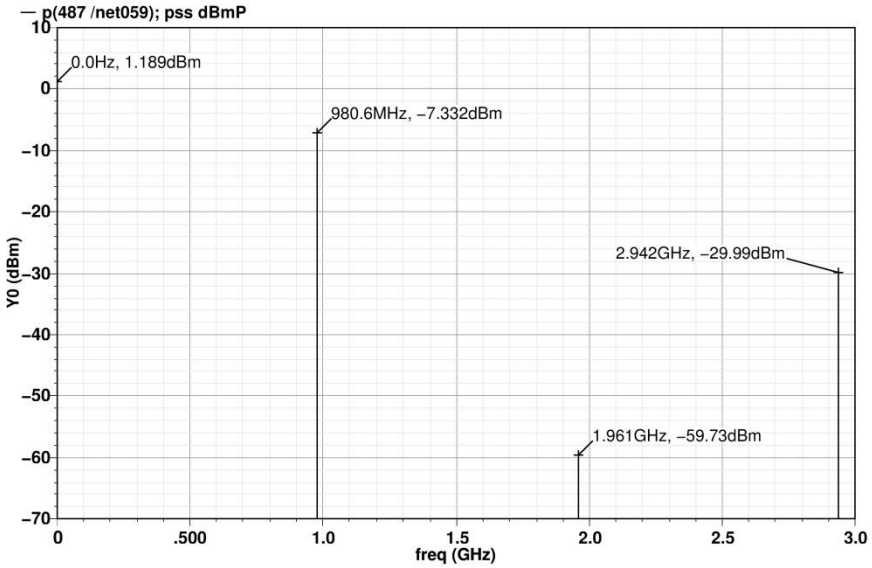


Figure 4.72: The signal power level at *980MHz* operating frequency.

As for frequency pushing, changing the supply voltage has a very weak effect on the *41MHz* frequency of operation whereas it has a stronger effect on the *980MHz* one. This is clearly displayed in Figure 4.73 and Figure 4.74 where the change in oscillation frequency is only *4MHz* and *54.2MHz* respectively.

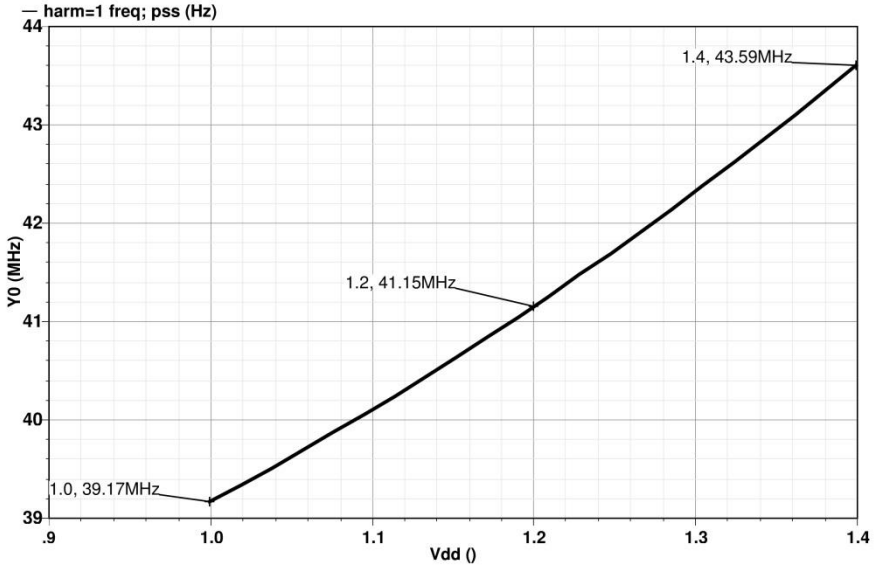


Figure 4.73: The effect of frequency pushing on a carrier frequency at *41MHz*.

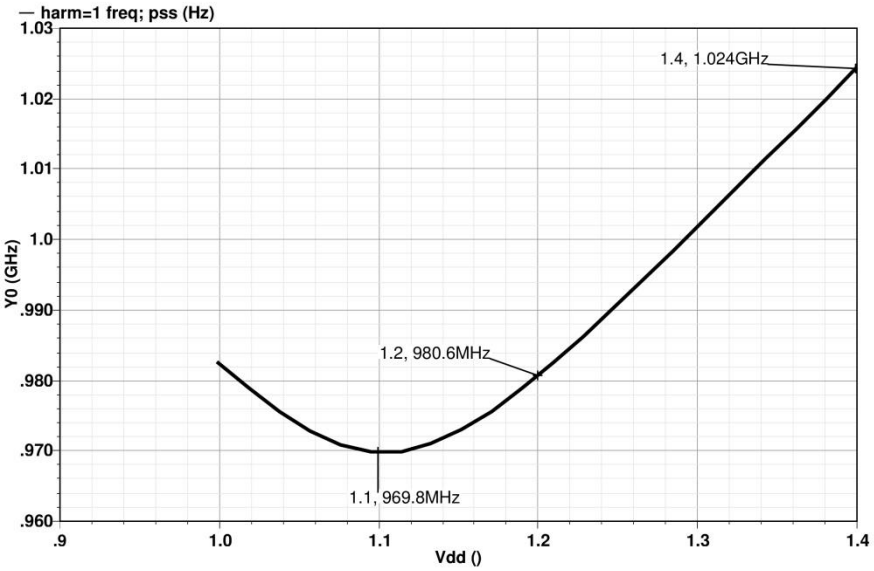


Figure 4.74: The effect of frequency pushing on a carrier frequency at *980MHz*.

Figure 4.75 and Figure 4.76 show the output phase noise at different offset frequencies from the carrier frequency (*41MHz* and *980MHz* respectively). The phase Noise was found to be -*101.5dBc/Hz* for the *41MHz* signal and *-84.55dBc/Hz* for the *980MHz* signal (at *1MHz* offset frequency). This level of phase noise is considered to be very high and does not even satisfy the requirements of our system design of the *TV* Tuner receiver which is *-100dBz/Hz*.

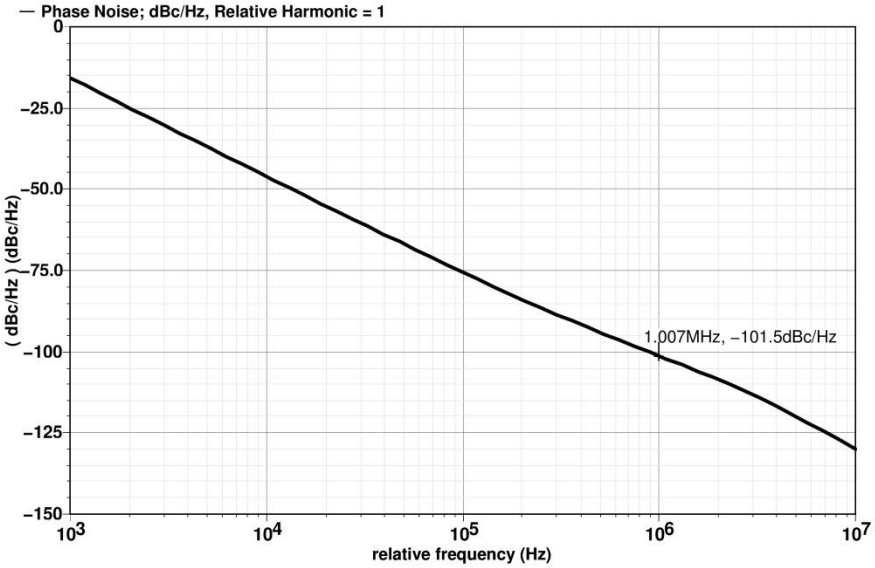


Figure 4.75: Output Phase Noise at different offset frequencies from a carrier frequency at *41MHz*.

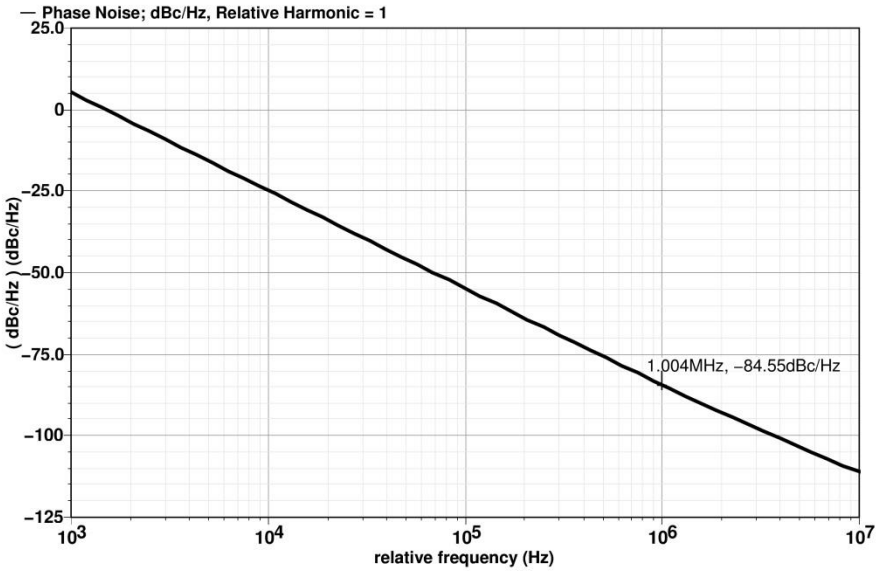


Figure 4.76: Output Phase Noise at different offset frequencies from a carrier frequency at *980MHz*.

It is evident from Figure 4.77 that the output phase noise exceeds *-100dBc/Hz* for most of the frequencies of operation which means that it cannot be used in our *TV* Tuner Receiver.

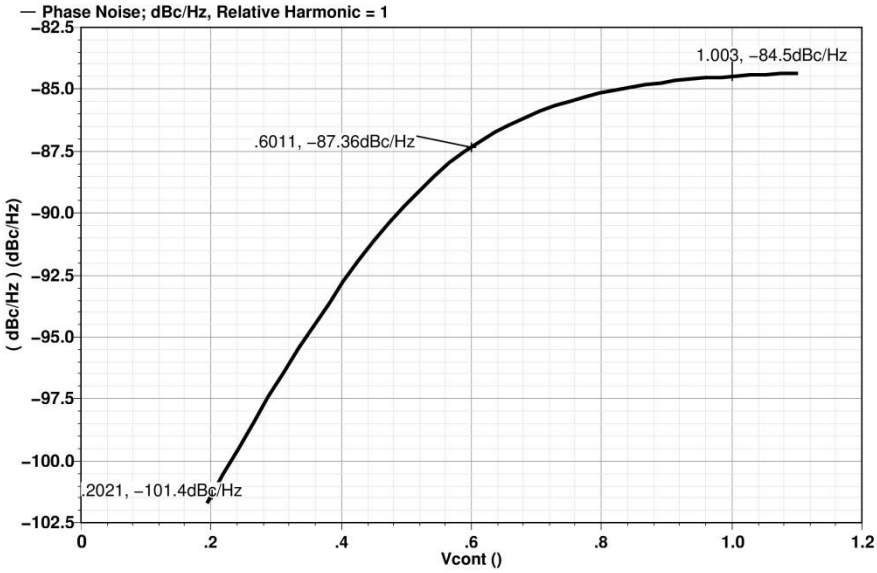


Figure 4.77: Phase Noise at *1MHz* offset frequency for all the frequencies of operation.

4.5.1.2 Findings:

As evident from the ongoing discussion, this topology depicted in Figure 4.60 achieves a very wide tuning range as the oscillation frequency can be varied by a ratio of *1:24* with only a two-fold variation in the amplitude. However, it suffers from two major drawbacks which are having a very high level of phase noise than allowed and the variation of power dissipation, output common-mode level, output swing and other performance parameters significantly across the tuning range. These two drawbacks are treated in the following section where another topology is proposed that achieves almost the same tuning range but

with a much higher level of performance that is nearly constant throughout the tuning range.

4.5.2 A Three-Stage Ring VCO with Interpolation and Positive Resistance:

As seen in the previous section, the three-stage ring *VCO* topology with the compound *PMOS*-load failed to achieve an acceptable level of phase noise and had a variable level of performance across the tuning range. In this section, we will propose a topology that overcomes these two problems by varying the delay without changing the current drawn from the supply. We start with a typical three-stage ring oscillator with a differential amplifier stage that has a resistive load as shown in Figure 4.78.

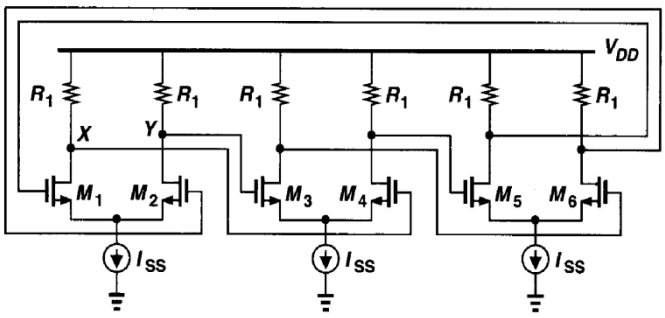


Figure 4.78: A typical three-stage differential Ring oscillator with a resistive load.

We add some tuning techniques to this circuit which achieve a variable delay in each amplifier stage as follows:

1. Positive Feedback through a Cross-Coupled transistor pair:

A Cross-Coupled transistor pair exhibits a negative input resistance ($-2/g_m$) which is directly controlled by the bias current. This negative resistance is placed in parallel with R_1 and R_2 as shown in Figure 4.79 which results in an equivalent load resistance of $R_1/(1 - g_m R_1)$. This total resistance is more positive if $1/g_m > R_1$ (the same applies to R_2). After applying this idea to each stage of the ring oscillator, as the bias current (I_1) increases, g_m increases and the total resistance of the load increases, therefore, the frequency of oscillation decreases and tunability is achieved.

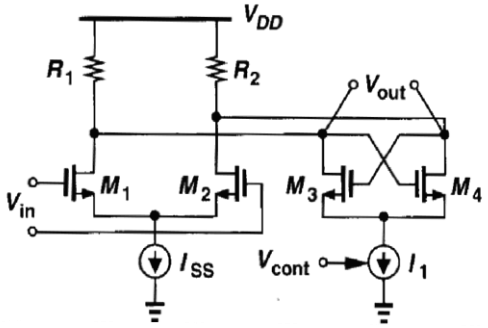


Figure 4.79: The differential amplifier stage with a variable negative resistance in parallel with the load.

It is important to highlight that as the bias current (I1) is changed, so do the currents flowing through R1 and R2. Consequently, the output voltage swing is not constant across the tuning range. This effect can be eliminated by varying ISS in the opposite direction such that the total current steered between R1 and R2 stays constant. I1 and ISS are varied differentially by using the differential Current-Folding topology illustrated in

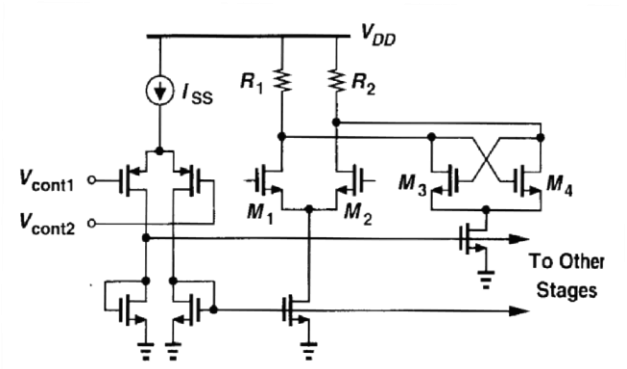


Figure 4.80 where the sum of I_1 and I_{SS} is always constant. This folding topology was chosen because it saves precious headroom. This circuit gives a differential output swing of

$$V_{SWING,DIFF} = 2R_1(I_{SS} + I_1). \qquad \text{Equation 4.20}$$

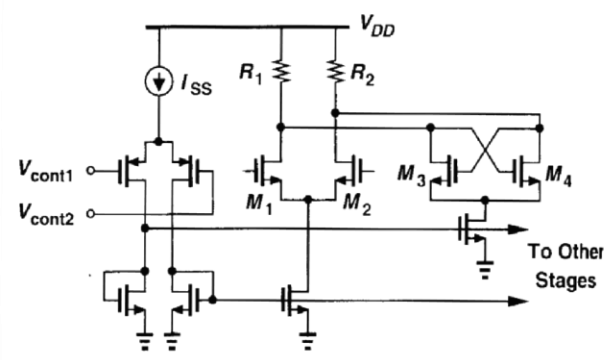


Figure 4.80: The Ring *VCO* stage with a differential Current-Folding amplifier used to vary I_1 and I_{SS} in opposite directions.

It is important to state that an additional small constant current source is used besides I_{SS} to prevent the current flowing through differential pair from dropping to zero which makes the gain of each stage zero prohibiting any oscillation. This ensures that M_1 and M_2 always remain on and the oscillation is maintained. This tuning technique provides a two-to-one frequency tuning with reasonable linearity,

2. Interpolation between a fast and a slow paths:

This approach is based on interpolation between two different paths namely a fast path and a slow path whose outputs are summed with variable gains as seen in Figure 4.81. The gains of each path are adjusted via the control voltage (V_{CONT}) in opposite directions.

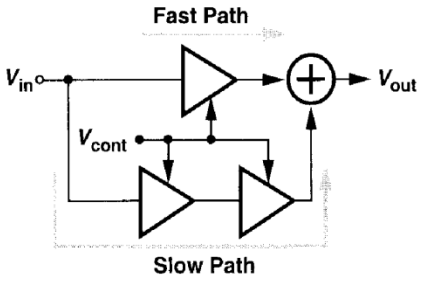


Figure 4.81: Interpolation between a fast path and a slow path to vary the delay of each stage of the ring VCO.

If the fast path is on and the slow one is completely disabled, the circuit oscillates with the maximum possible frequency. On the other hand, if the fast path is disabled and the slow one is on, the circuit oscillates with the minimum possible frequency. When V_{CONT} is in between the two extremes, each path partially shares in the output signal and the frequency is between the max and min values. By implementing this interpolation technique in each stage of the ring oscillator, the delay of each stage becomes variable which also varies the frequency of oscillation.

As expressed in Figure 4.82, each stage is implemented as two simple differential pairs with resistive loads whose gains are made variable by means of their tail current (the gains are varied in opposite directions). The output currents of the fast and slow paths are added together and fed to the resistances R_1 and R_2 to achieve interpolation.

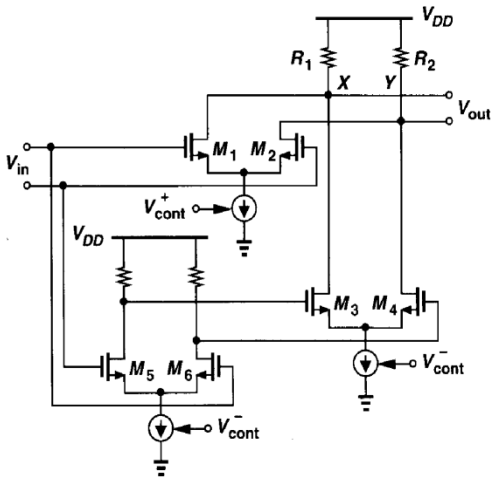


Figure 4.82: Interpolating stage implemented as differential pairs with resistive loads and different delays whose output currents are summed together.

It is important to state that the gain of the differential pair M_5-M_6 does not have to be varied because the slow path is already fully disabled when the gain of M_3-M_4 drops to zero. However, we are going to make it variable anyway because by doing so the tuning range is expanded as much as possible. Also, the sum of the tail currents of the fast (M_1-M_2) and slow (M_3-M_4) differential pairs is kept constant throughout the tuning range by varying them in opposite directions. This achieves interpolation between the two paths with a constant voltage swing across the tuning range. Since the slow path has an extra stage compared to the fast one, this idea also provides a two-to-one tuning range.

To conclude, adding the cross-coupled transistor pair achieves a tuning range of *1:2*. When this cross-coupled pair is added to the loads of the interpolating circuit in Figure 4.82, the tuning range is increased significantly. Also, by varying the gain of transistors (M_5 *and* M_6), the tuning range is additionally expanded. The proposed topology is expressed in Figure 4.83. In order to obtain a constant voltage swing across the tuning range, the total current through the load resistances is kept constant by employing a differential current folding topology that varies the currents in opposite directions while keep the sum always constant. In addition to always keeping the total current constant, the current-folding topology also saves valuable headroom.

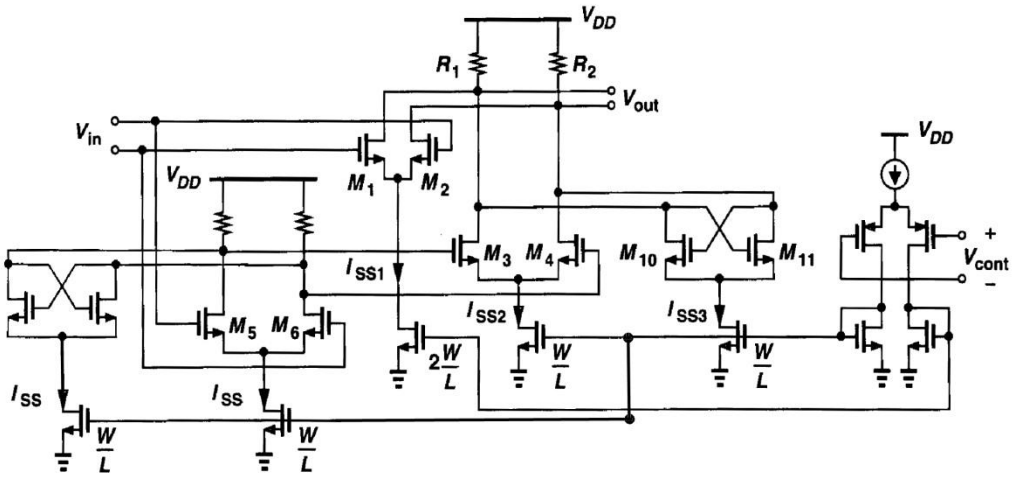


Figure 4.83: A single stage circuit of a three-stage ring oscillator that employs both interpolation and positive feedback.

As the current is steered to transistors (M_1 and M_2) only the fast path works and the frequency of oscillation is high. On the other hand, when the current is steered to transistors (M_3, M_4, M_{10} and M_{11}) the slow path works with positive feedback resulting in a very low frequency of oscillation. Throughout the tuning range, the total current flowing through the load resistances is kept constant by setting

$$I_{SS1} = I_{SS2} + I_{SS3} \qquad \text{Equation 4.21}$$

4.5.2.1 Simulation Results:

In the following, we will analyze the simulation results obtained using Cadence for our design. It is important to state that the startup condition expressed in Equation 4.19 is satisfied by our design of the amplifier depicted in Figure 4.83 throughout the tuning range. This amplifier is used as a replica circuit to construct a three-stage ring *VCO*. Also, we will obtain simulation results for both the maximum and minimum oscillation frequencies so that we could compare them with the *PMOS*-Load topology discussed in the previous section.

Figure 4.84 demonstrates the tuning curve achieved by this *VCO*. For the charge-pump designed in 1 with V_{CONT} from *0.2V* to *1V*, the tuning range available by this *VCO* is from *43MHz* to *1.295GHz*. This tuning range is considered to be extremely wide (*1:30*) and it covers more than the range of frequency operation with a single *VCO*.

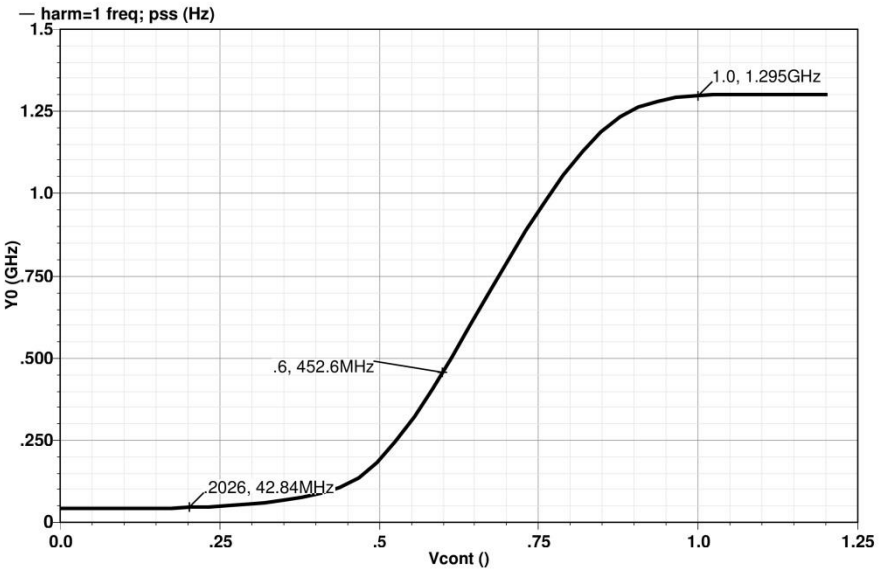


Figure 4.84: The tuning range of the three-stage Ring *VCO* that employs interpolation and positive feedback.

Figure 4.85 and Figure 4.86 illustrate that the obtained single ended peak-to-peak output swing is *0.43V* at *43Mhz* oscillation frequency and *0.1V* at *1.295GHz* which demonstrates that the output voltage swing is not constant and decrease by four times across the tuning range. It is also clear that this *VCO* does not have a constant common-mode level across the tuning range which is considered a disadvantage.

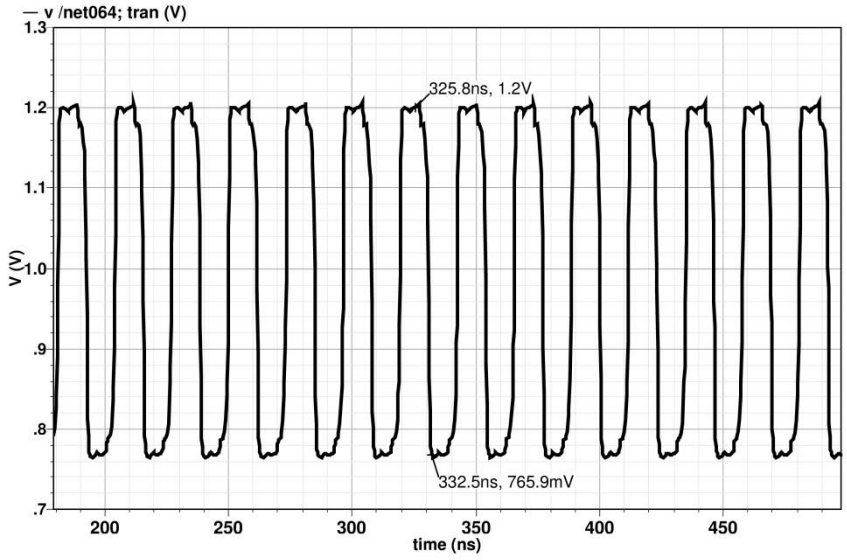


Figure 4.85: The single ended output swing obtained by the three-stage Ring *VCO* that employs interpolation and positive feedback at *43MHz* oscillation frequency.

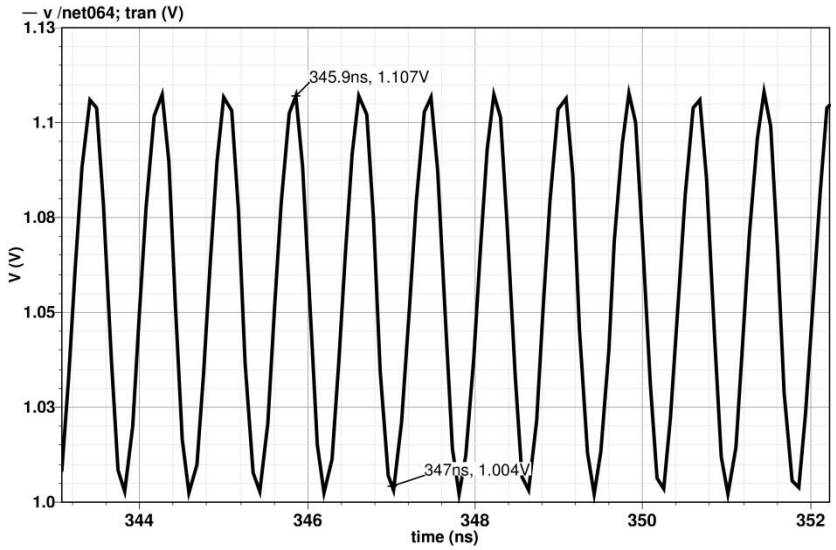


Figure 4.86: The single ended output swing obtained by the three-stage Ring *VCO* that employs interpolation and positive feedback at *1.295GHz* oscillation frequency.

On the other hand, Figure 4.87 and Figure 4.88 depict the differential output swing obtained at both the minimum and maximum oscillation frequencies. It is clear that the generated signal has sinusoidal-like waveform. This signal is readily applied to a buffer or an inverter stage to obtain a square-like waveform with *50%* duty cycle.

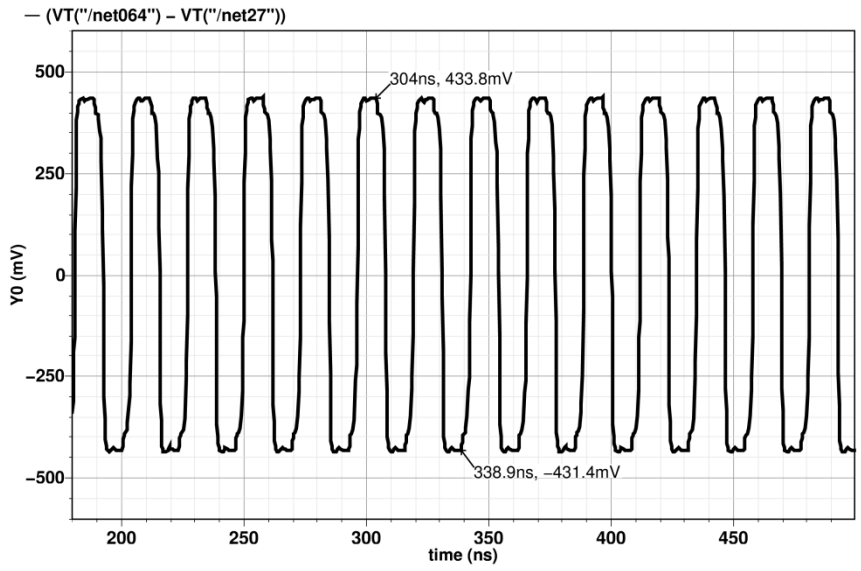


Figure 4.87: The differential output swing obtained by the three-stage Ring *VCO* that employs interpolation and positive feedback at *43MHz* oscillation frequency.

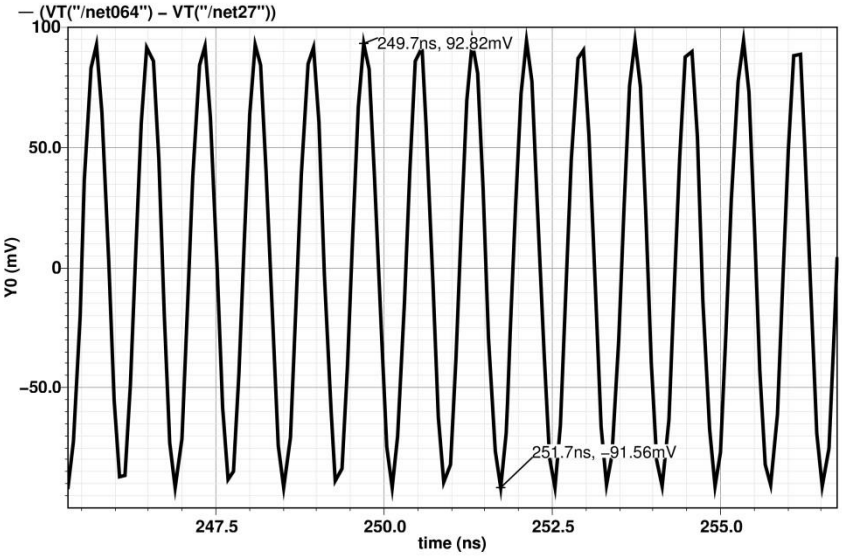


Figure 4.88: The differential output swing obtained by the three-stage Ring *VCO* that employs interpolation and positive feedback at *1.295GHz* oscillation frequency.

As for the sensitivity (K_{VCO}) of the designed *VCO*, as evident by Figure 4.89, it varies by orders of magnitude across the tuning range (increases by a factor of *100* then decreases again by a factor of *10*). This behavior is undesirable in *PLLs* as the sensitivity is required to be almost constant throughout the tuning range.

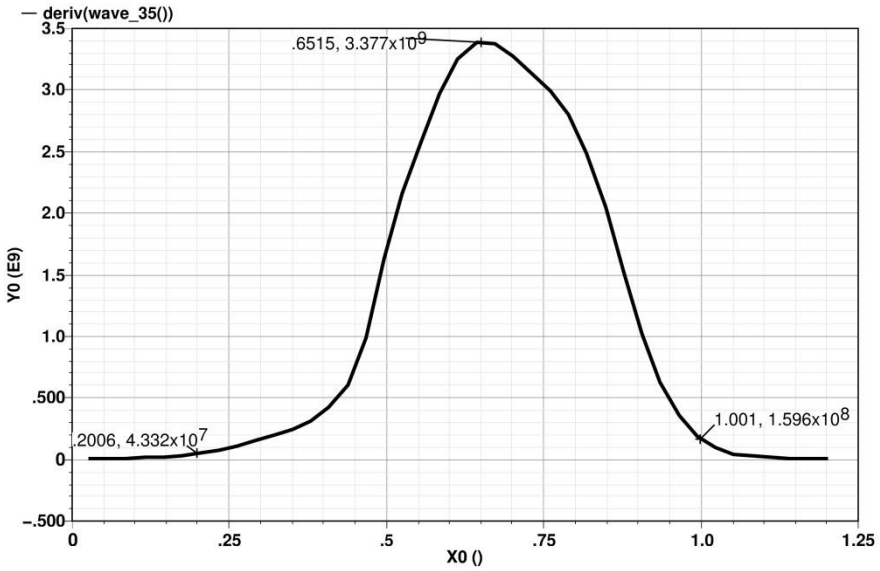


Figure 4.89: The sensitivity of the three-stage Ring *VCO* that employs interpolation and positive feedback.

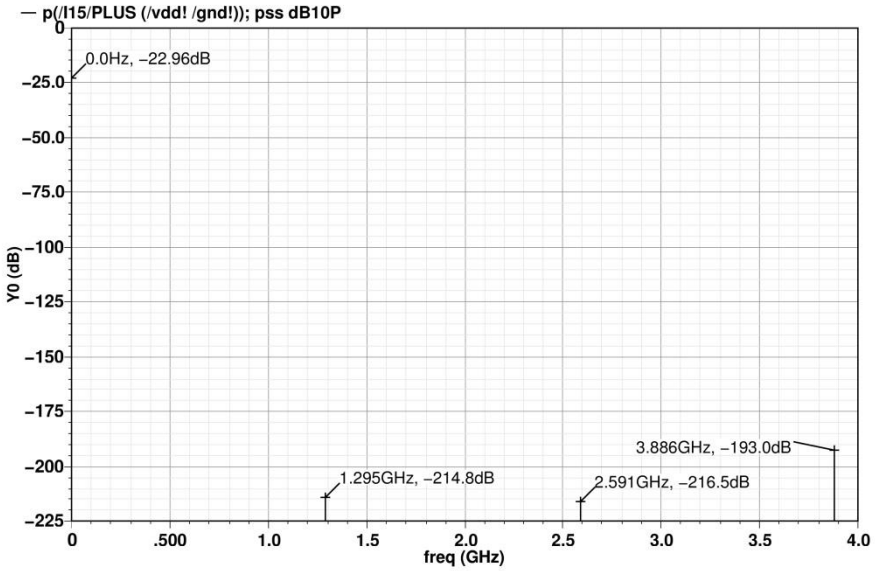


Figure 4.90 presents the power dissipation of this *VCO* (*-22.96dB ≈ 5mW*) at *1.295GHz* oscillation frequency. On the other hand, Figure 4.91 presents the power dissipation of the *VCO* (*-20.95dB ≈ 8mW*) at *43MHz* oscillation frequency. It is obvious that there is a very small variation in the power dissipation throughout the tuning range unlike the topology with the *PMOS*-Load.

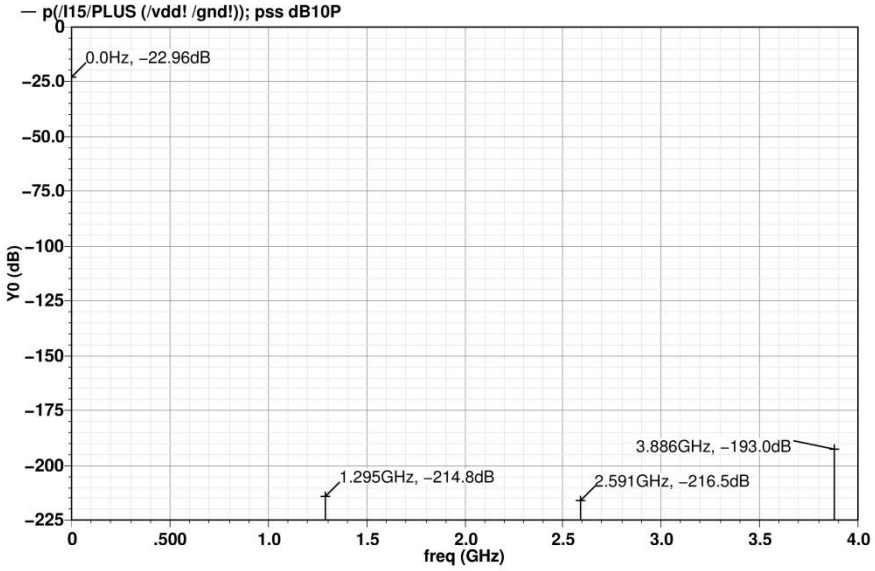


Figure 4.90: The power dissipation of the three-stage Ring *VCO* that employs interpolation and positive feedback at *1.295GHz* oscillation frequency.

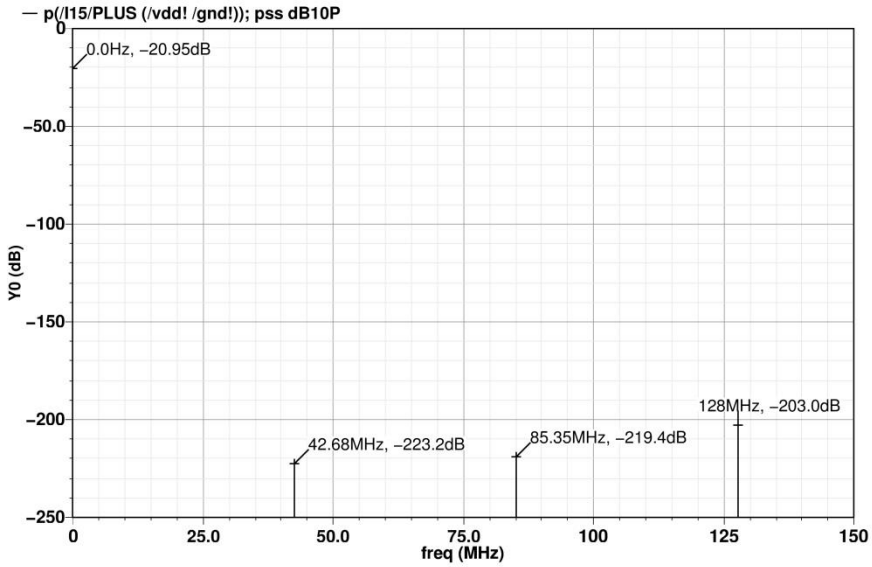


Figure 4.91: The power dissipation of the three-stage Ring *VCO* that employs interpolation and positive feedback at *43MHz* oscillation frequency.

The power level of the output signal oscillating at frequency *43MHz* is *-10.81dBm* as expressed in Figure 4.92 whereas at frequency *1.295GHz* the signal power level is *-24.98dBm* as shown in Figure 4.93. It is also clear that there is a very small variation in the signal power level across the tuning range unlike the topology with the *PMOS*-Load.

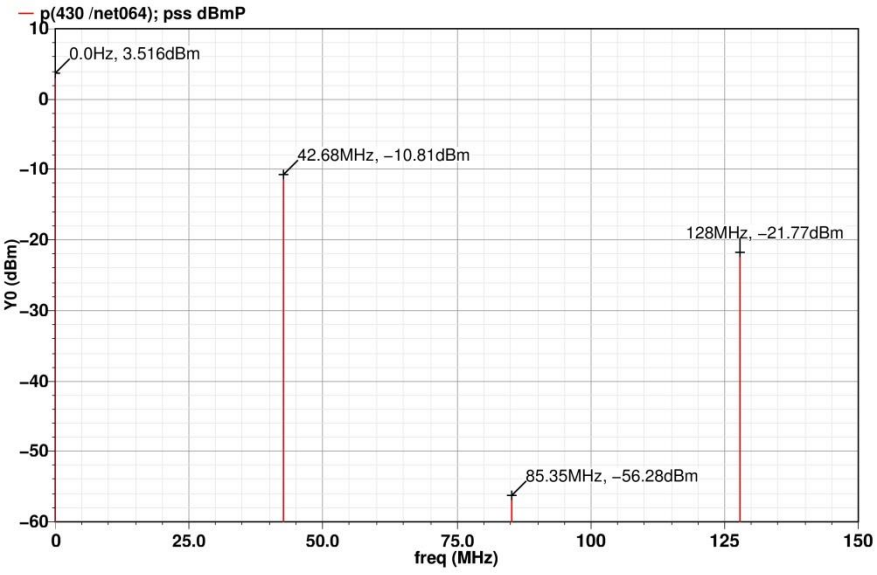


Figure 4.92: The signal power level at *43MHz* operating frequency.

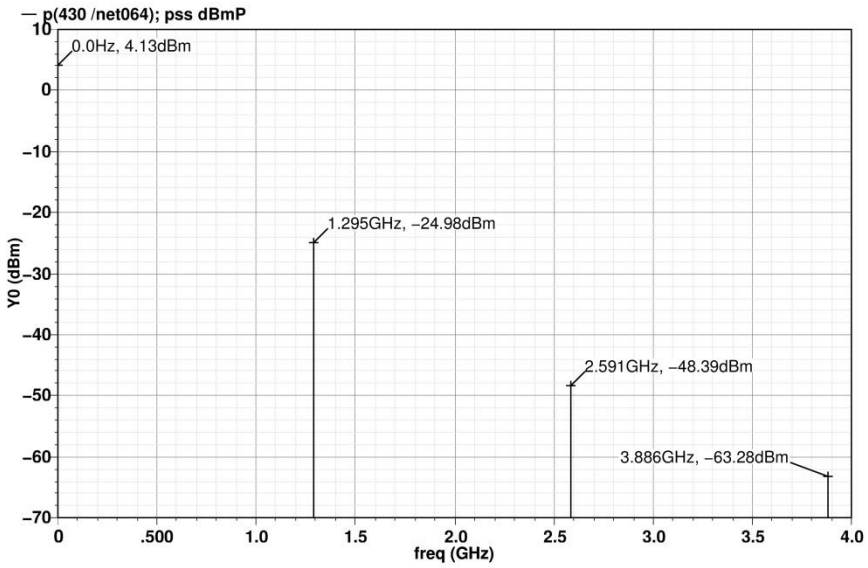


Figure 4.93: The signal power level at *1.295GHz* operating frequency.

As for frequency pushing, changing the supply voltage has a very weak effect on the *43MHz* frequency of operation whereas it has a very strong effect on the *1.295GHz* one. This is clearly displayed in Figure 4.94 and Figure 4.95 where the change in oscillation frequency is only *5.5MHz* and *219MHz* respectively. Also, as evident in Figure 4.95, the circuit does not oscillate at high frequencies for supply values below *1.15V*.

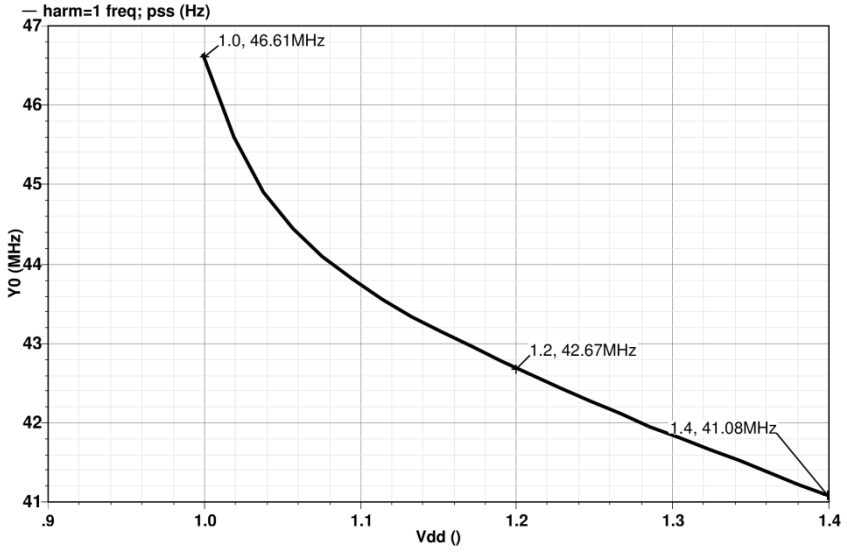


Figure 4.94: The effect of frequency pushing on a carrier frequency at *43MHz*.

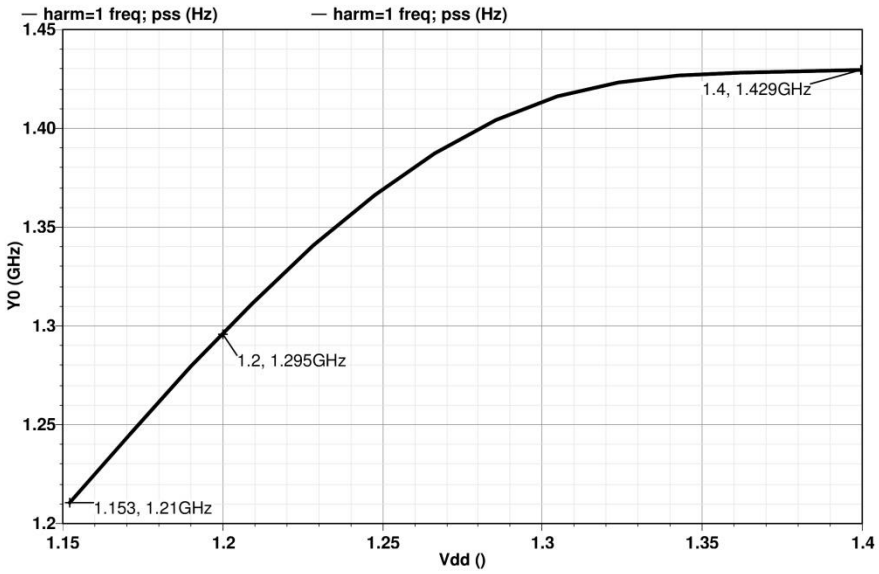


Figure 4.95: The effect of frequency pushing on a carrier frequency at *1.29GHz*.

Figure 4.96 and Figure 4.97 show the output phase noise at different offset frequencies from the carrier frequency (carrier frequencies at *43MHz* and *1.295GHz* respectively). The phase Noise was found to be *-110.4dBc/Hz* for the *43MHz* signal and *-91.87dBc/Hz* for the *1.295GHz* signal (at *1MHz* offset frequency). This level of phase noise is considered to be much lower than that obtained by the *PMOS*-Load topology. However, it still does not satisfy the requirements of our system design of the *TV* Tuner receiver which is *-100dBz/Hz*.

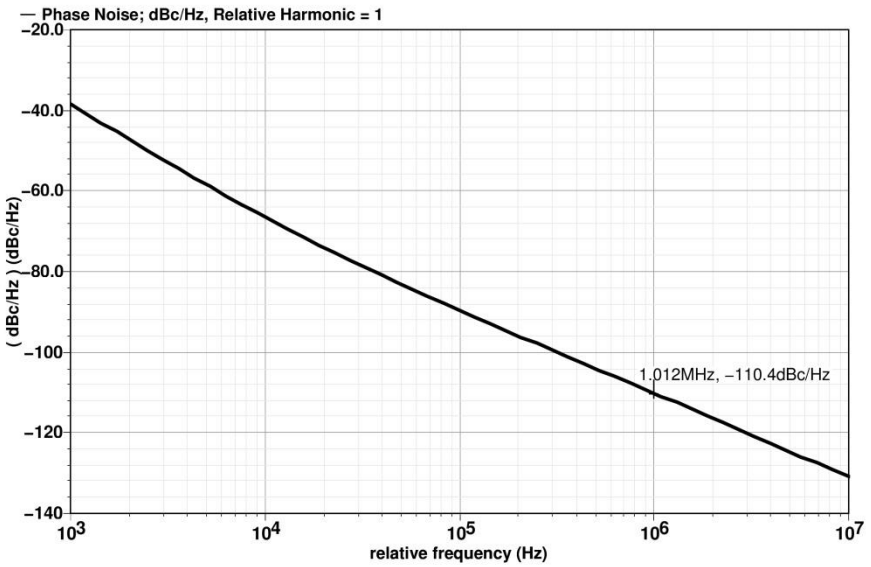


Figure 4.96: Output Phase Noise at different offset frequencies from a carrier frequency at *43MHz*.

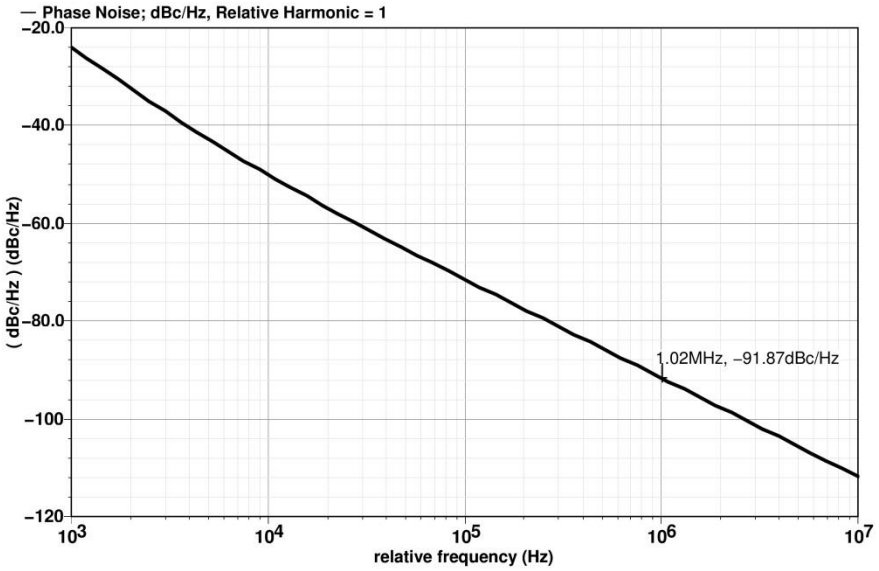


Figure 4.97: Output Phase Noise at different offset frequencies from a carrier frequency at *1.295GHz*.

It is evident from Figure 4.98 that the output phase noise exceeds *-100dBc/Hz* for most of the frequencies of operation which means that it cannot be used in our *TV* Tuner Receiver.

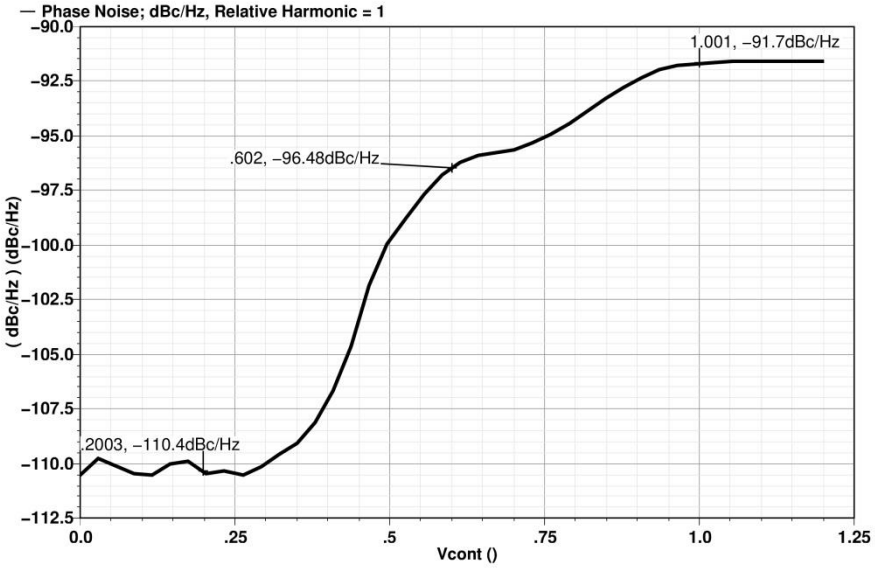


Figure 4.98: Phase Noise at *1MHz* offset frequency for all the frequencies of operation.

4.5.2.2 Findings:

It is obvious that this topology (depicted in Figure 4.83) achieves a very wide tuning range as the oscillation frequency can be varied by a ratio of *1:30* but with a four-fold variation in the amplitude. Unlike the PMOS-Load topology, this *VCO* has a relatively constant performance across the tuning range of frequency operation. However, it suffers from a major drawback which is having a very high level of phase noise than allowed. It also suffers from variations in the output common-mode level and output swing across the tuning range. These two drawbacks are treated in the following section where the same topology is employed but in

a four-stage ring *VCO* with some minor modifications to decrease the phase noise. The proposed four-stage circuit will also cover the entire tuning range required but with a much higher level of performance that is nearly constant throughout the tuning range.

4.5.3 Four-Stage Ring VCO with Interpolation, Positive Resistance and Additional Tail Capacitance:

As discussed in the previous section, the three-stage ring *VCO* topology that employs cross-coupled transistors pair and interpolation failed to achieve an acceptable level of phase noise. In this section, we will propose a ring oscillator that employs the same topology but in a four-stage ring *VCO*. It is expected that by increasing the number of stages, the noise immunity will increase significantly. There are also some minor modifications that aim at further reducing the level of phase noise.

In any Ring *VCO* circuit, there must be an odd number of inversions in the loop to prevent the circuit from latching up as explained earlier. For example, we can have a five-stage ring *VCO* made up of five inverting circuits, providing a frequency of oscillation equals to $1/(10T_D)$ where T_D is the large signal delay of each stage. On the other hand, to implement a four-stage ring *VCO*, we must employ differential stages with three inverting stages and one stage acting as a buffer. Such a configuration is illustrated in Figure 4.99 which provides a frequency of oscillation equals to $1/(8T_D)$. The ability to construct a Ring *VCO* with even number of stages is one of the advantages that differential topologies have over single ended ones.

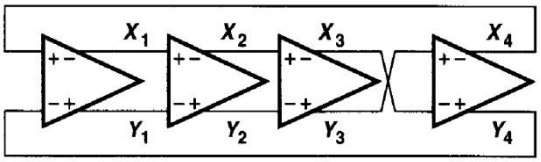


Figure 4.99: Four-stage Ring oscillator with one of the stages acting as a buffer.

The open-loop transfer function of the four-stage ring oscillator is given by

$$H(s) = -\frac{A_O^4}{\left(1+\frac{s}{\omega_O}\right)^4}. \qquad \text{Equation 4.22}$$

As for the startup condition, the minimum low frequency voltage gain per stage can be calculated by first finding the frequency that satisfies Barkhausen's criteria (Equation 4.2 and Equation 4.3). To satisfy the phase criteria, each stage contributes a frequency-dependent phase shift of 45^O. The frequency at which this is achieved can be obtained from

$$\tan^{-1}\frac{\omega_{OSC}}{\omega_O} = 45^O \qquad \text{Equation 4.23}$$

Hence

$$\omega_{OSC} = \omega_O\,. \qquad \text{Equation 4.24}$$

Consequently, the minimum voltage gain per stage is derived by equating the magnitude of the gain to *1* as follows

$$\frac{A_O}{\sqrt{1+\left(\frac{\omega_{OSC}}{\omega_O}\right)^2}} = 1 \qquad \text{Equation 4.25}$$

Therefore

$$A_O = \sqrt{2}\,. \qquad \text{Equation 4.26}$$

This obtained value for the minimum low frequency gain of each stage is lower than that required in the three-stage circuit as expected. The *VCO* provides four different phases and their complements as illustrated in Figure 4.100.

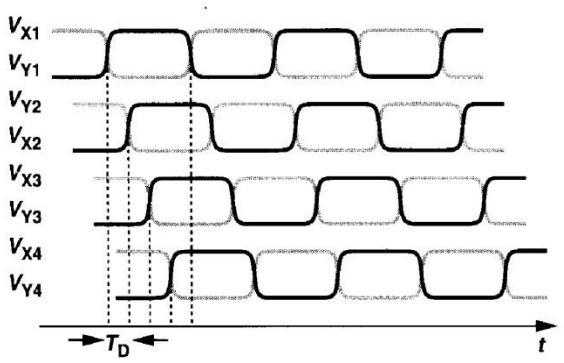


Figure 4.100: The output waveforms generated by the four-stage Ring *VCO* each with a different phase.

In addition to increasing the number of stages of the Ring *VCO*, there is a simple modification that can further reduce the phase noise. This modification is simply to add an additional capacitance in parallel with the tail-current source of each differential pair in each stage of the Ring *VCO*. This capacitor absorbs most of the current noise from the current source. Consequently, the total phase noise of the *VCO* is reduced significantly.

4.5.3.1 Simulation Results:

In the following, we will analyze the simulation results obtained using Cadence for our design. It is important to state that the startup condition expressed in Equation 4.19 is satisfied by our design of the amplifier depicted in Figure 4.83 throughout the tuning range. This amplifier is used as a replica circuit to construct a four-stage ring *VCO*. Also, we will obtain simulation results for both the maximum and minimum oscillation frequencies so that we could compare them with the *PMOS*-Load topology discussed in the previous section.

Figure 4.101 demonstrates the tuning curve achieved by this *VCO*. For the charge-pump designed in 1 with V_{CONT} from *0.2V* to *1V*, the tuning range available by this *VCO* is from *43MHz* to *867MHz*. This tuning range is considered to be wide enough (*1:20*) and it exactly covers the entire range of frequency operation with a single *VCO*.

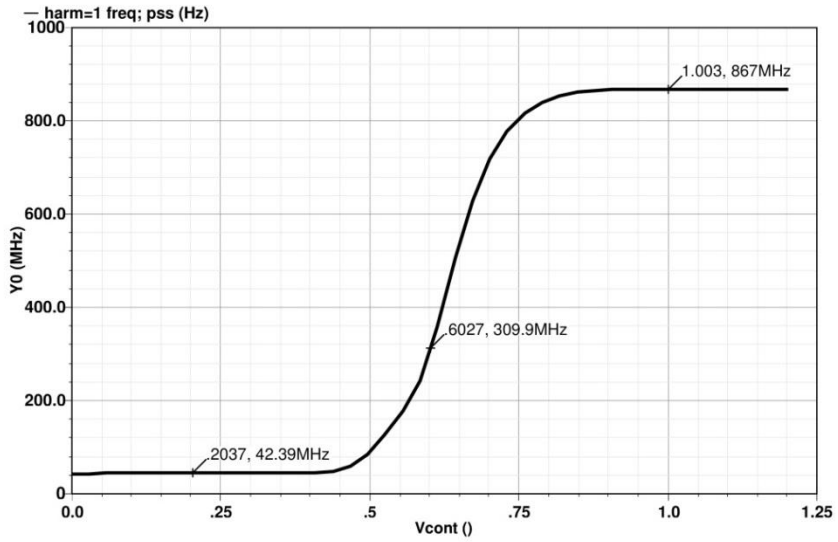


Figure 4.101: The tuning range of the Four-stage Ring VCO that employs interpolation, additional tail capacitance and positive feedback.

Figure 4.102 and Figure 4.103 illustrate that the obtained single ended peak-to-peak output swing is *0.52V* at *42Mhz* oscillation frequency and *0.37V* at *867MHz* which demonstrates that the output voltage varies slightly across the tuning range. It is also clear that this *VCO* has an almost constant common-mode level across the tuning range which is considered a major advantage.

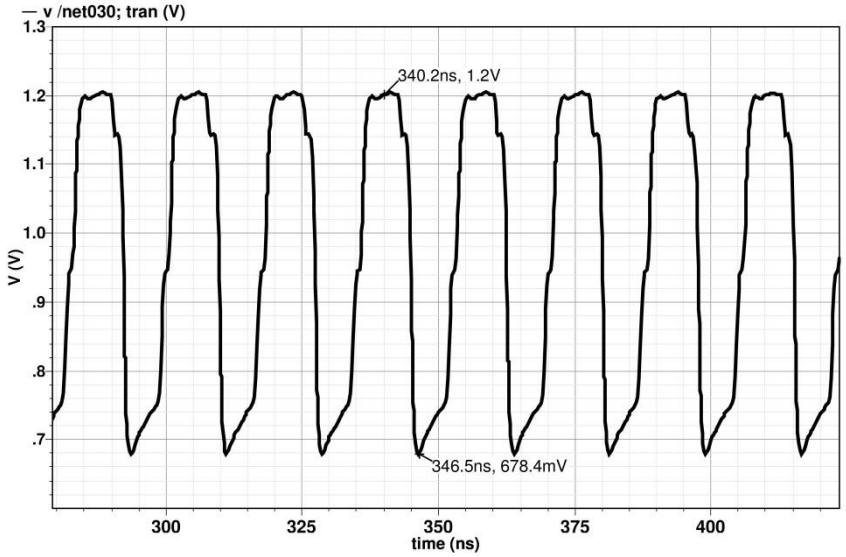


Figure 4.102: The single ended output swing obtained by the Four-stage Ring *VCO* that employs interpolation, additional tail capacitance and positive feedback at *42MHz* oscillation frequency.

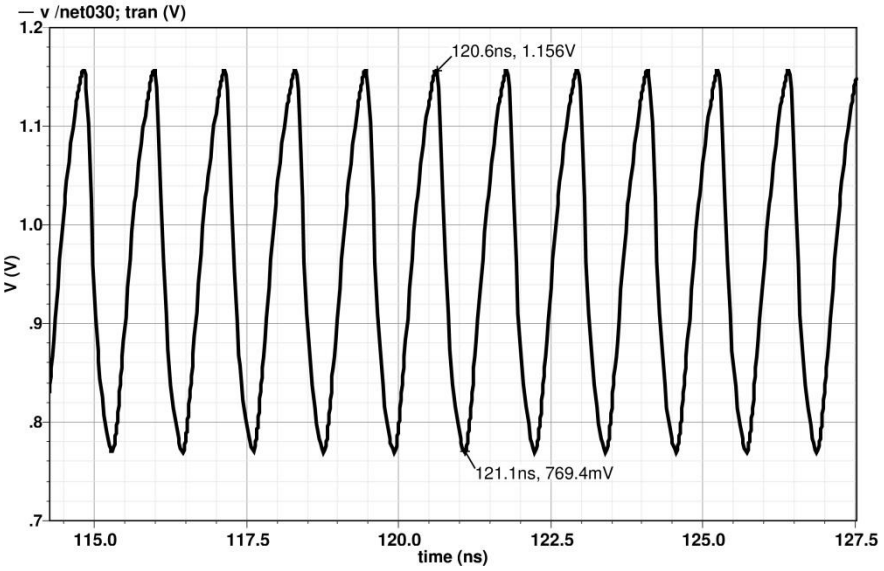


Figure 4.103: The single ended output swing obtained by the Four-stage Ring *VCO* that employs interpolation, additional tail capacitance and positive feedback at *867MHz* oscillation frequency.

On the other hand, Figure 4.104 and Figure 4.105 depict the differential output swing obtained at both the minimum and maximum oscillation frequencies. It is clear that the generated signal has sinusoidal-like waveform. This signal is readily applied to a buffer or an inverter stage to obtain a square-like waveform with *50%* duty cycle.

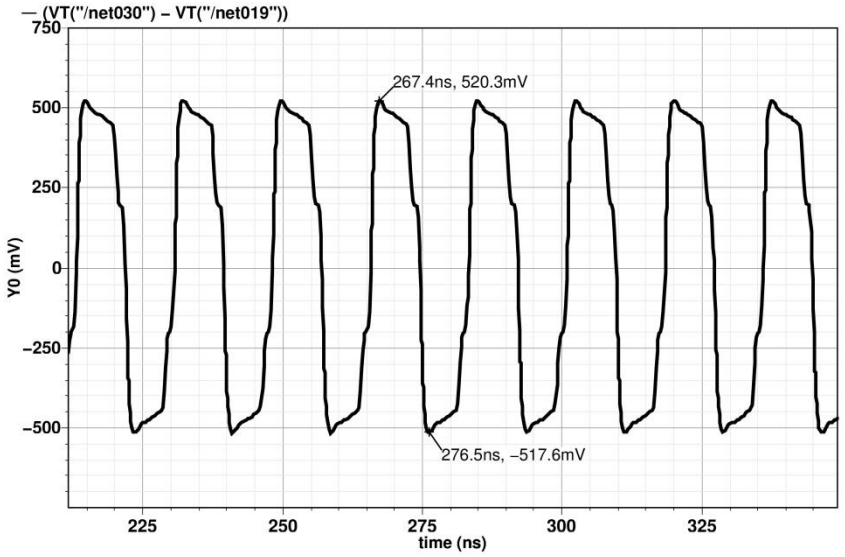


Figure 4.104: The differential output swing obtained by the three-stage Ring *VCO* that employs interpolation and positive feedback at *43MHz* oscillation frequency.

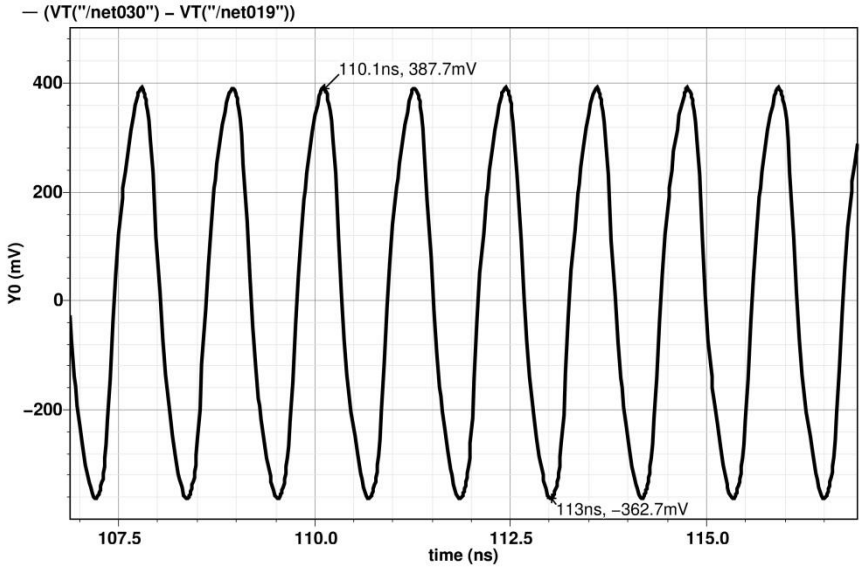


Figure 4.105: The differential output swing obtained by the three-stage Ring *VCO* that employs interpolation and positive feedback at *1.295GHz* oscillation frequency.

As for the sensitivity (K_{VCO}) of the designed *VCO*, as evident by Figure 4.106, it varies by orders of magnitude across the tuning range (increases by a factor of *5000* then decreases again by a factor of *1000*). This behavior is undesirable in PLLs as the sensitivity is required to be almost constant throughout the tuning range.

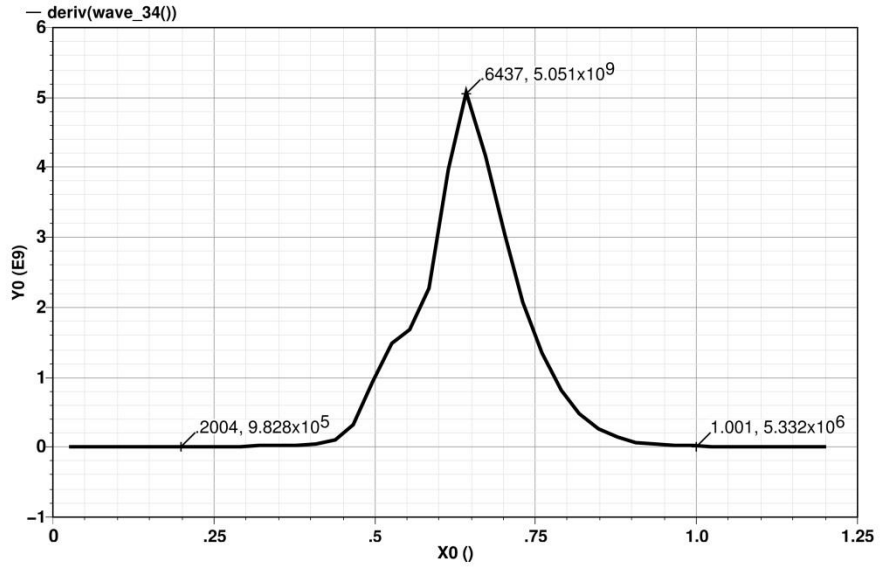


Figure 4.106: The sensitivity of the three-stage Ring *VCO* that employs interpolation, additional tail capacitance and positive feedback.

Figure 4.107 presents the power dissipation of this *VCO* (*-20.97dB ≈ 8mW*) at *42MHz* oscillation frequency. On the other hand, Figure 4.108 presents the power dissipation of the *VCO* (*-20.84dB ≈ 8.2mW*) at *867MHz* oscillation frequency. It is obvious that variation in the power dissipation is extremely small throughout the tuning range unlike the topology with the *PMOS*-Load.

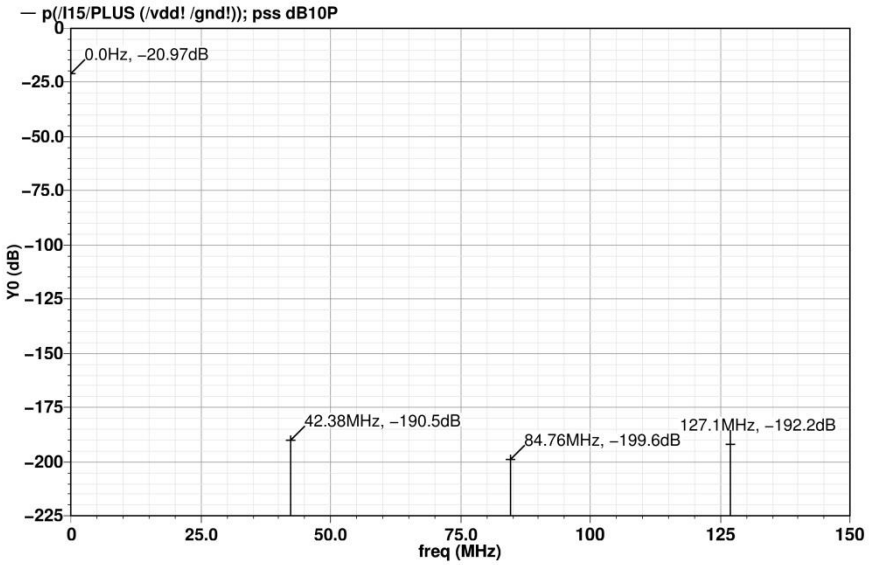


Figure 4.107: The power dissipation of the four-stage Ring *VCO* that employs interpolation, additional tail capacitance and positive feedback at *42MHz* oscillation frequency.

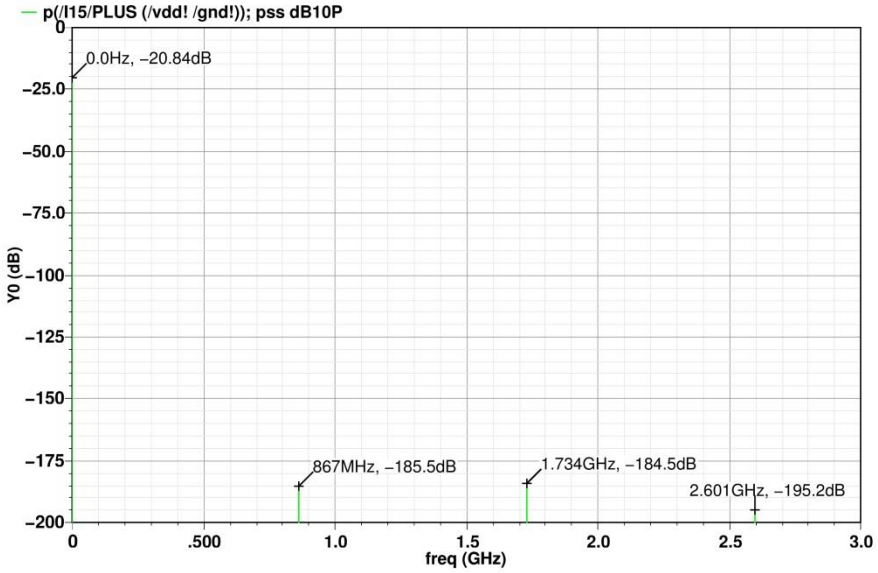


Figure 4.108: The power dissipation of four-stage Ring *VCO* that employs interpolation, additional tail capacitance and positive feedback at *867MHz* oscillation frequency.

The power level of the output signal oscillating at frequency *42MHz* is *-9.916dBm* as expressed in Figure 4.109 whereas at frequency *867MHz* the signal power level is *-14.92dBm* as shown in Figure 4.110. It is also clear that there is a somehow large variation in the signal power level across the tuning range like the topology with the *PMOS*-Load.

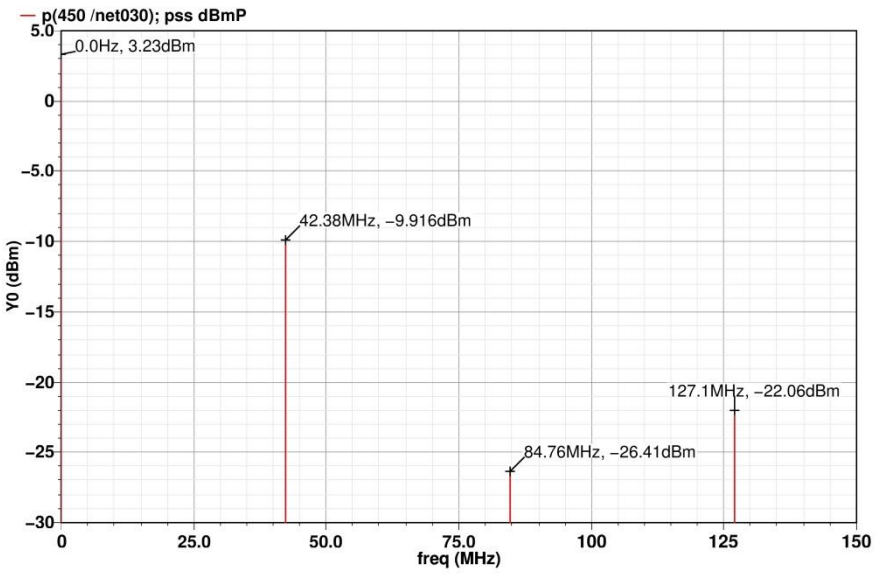


Figure 4.109: The signal power level at *42MHz* operating frequency.

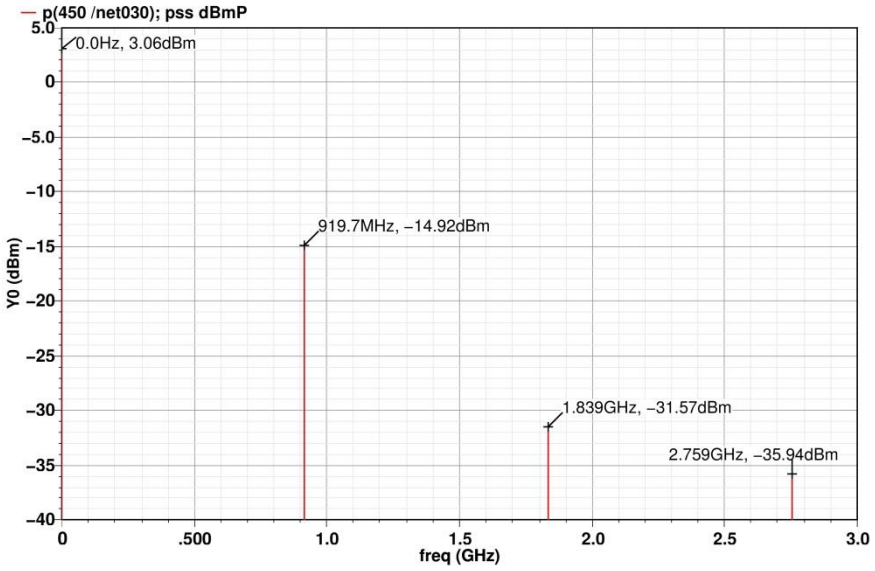


Figure 4.110: The signal power level at *867MHz* operating frequency.

As for frequency pushing, changing the supply voltage has a very weak effect on the *42MHz* frequency of operation whereas it has a very strong effect on the *867MHz* one. This is clearly displayed in Figure 4.111 and Figure 4.112 where the change in oscillation frequency is only *3.24MHz* and *131MHz* respectively. Also, as evident in Figure 4.112, the circuit maintains oscillation at high frequencies for supply values below *1.2V*.

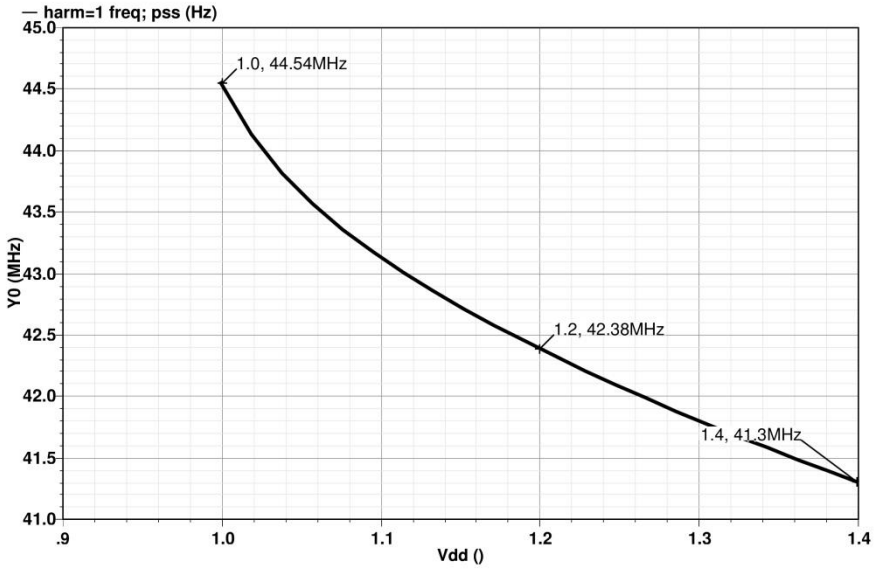


Figure 4.111: The effect of frequency pushing on a carrier frequency at *43MHz*.

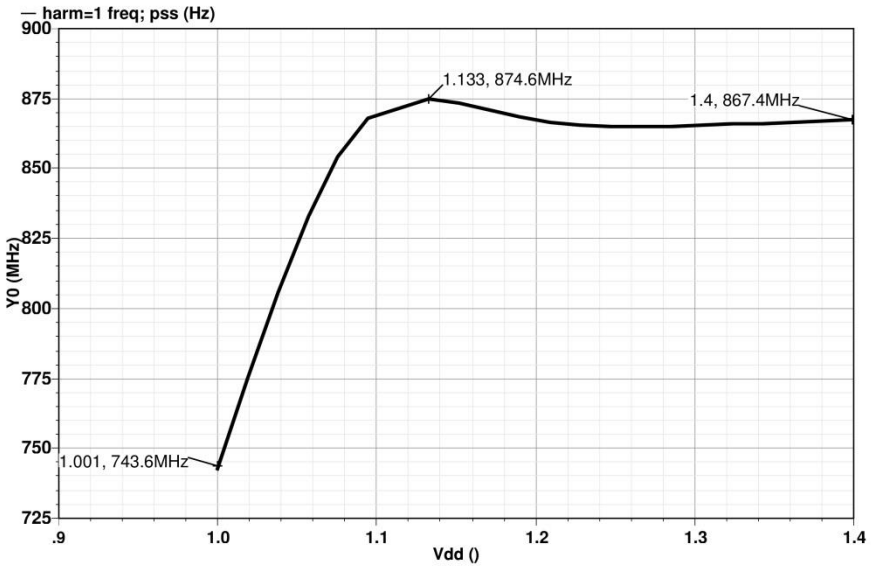


Figure 4.112: The effect of frequency pushing on a carrier frequency at *1.29GHz*.

Figure 4.113 and Figure 4.114 show the output phase noise at different offset frequencies from the carrier frequency (carrier frequencies at *42MHz* and *867MHz* respectively). The phase Noise was found to be *-121.1dBc/Hz* for the *42MHz* signal and *-105dBc/Hz* for the *1.295GHz* signal (at *1MHz* offset frequency). This level of phase noise is considered highly acceptable and it is much lower than that obtained by the *PMOS*-Load topology or the *3*-stage topology with interpolation. It also satisfies the requirements of our system design of the *TV* Tuner receiver which is *-100dBz/Hz*.

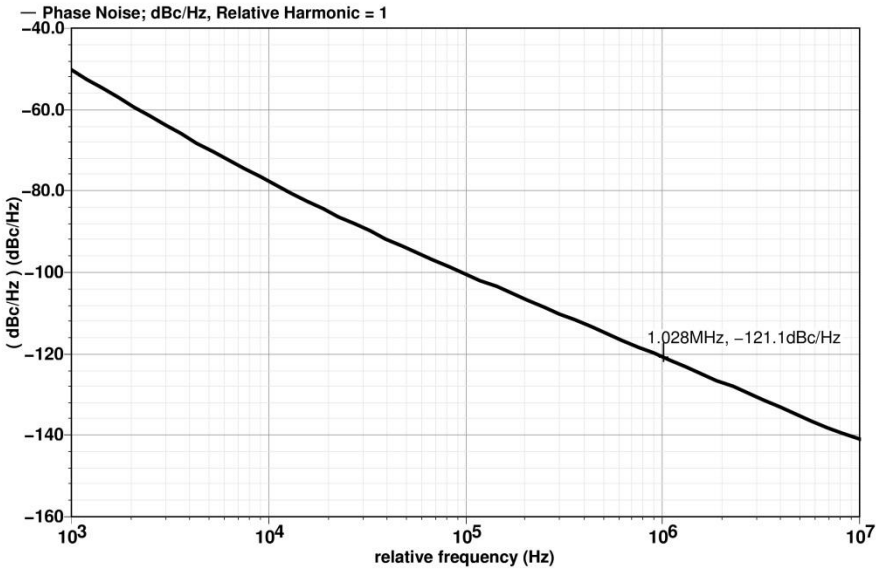


Figure 4.113: Output Phase Noise at different offset frequencies from a carrier frequency at *42MHz*.

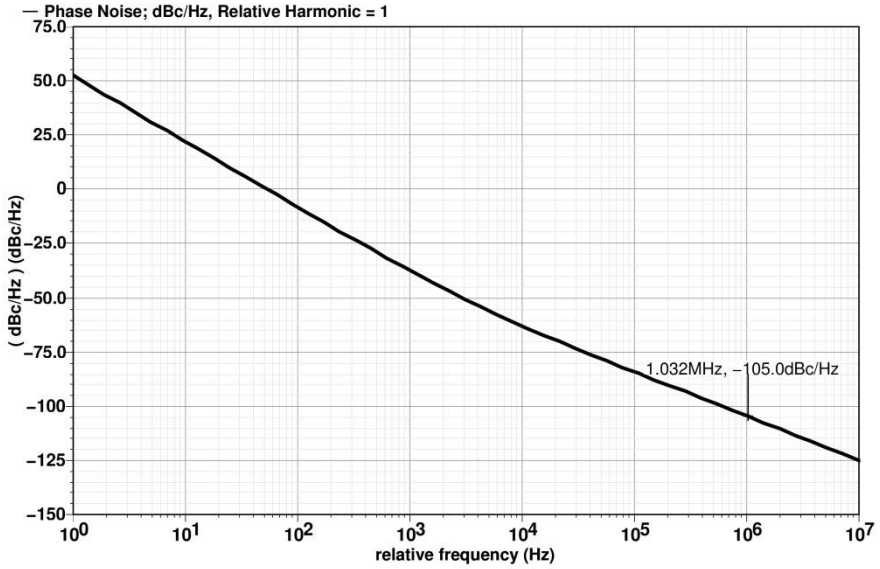


Figure 4.114: Output Phase Noise at different offset frequencies from a carrier frequency at *867MHz*.

It is evident from Figure 4.115 that the output phase noise does not exceed *-100dBc/Hz* for all the frequencies of operation which means that it can indeed be used in our *TV* Tuner Receiver.

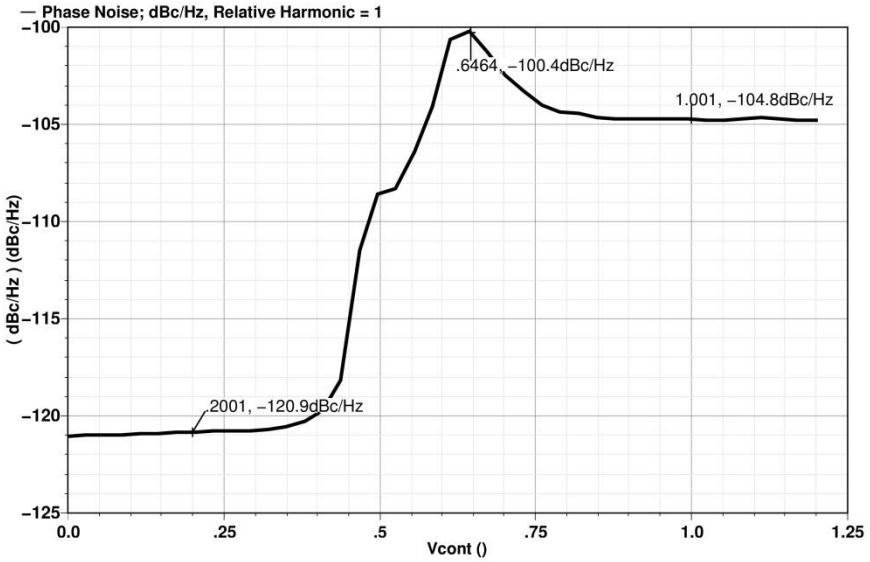


Figure 4.115: Phase Noise at *1MHz* offset frequency for all the frequencies of operation.

4.5.3.2 Findings:

It is clear from the ongoing discussion that this topology depicted in Figure 4.83 achieves a very wide tuning range as the oscillation frequency can be varied by a ratio of *1:20* and with only a slight variation in the amplitude. Unlike the *PMOS*-Load topology, this *VCO* has a constant performance across the tuning range of frequency operation.

4.5.4 Summary of the Ring VCOs Simulation Results:

Table 4.2 provides a complete summary of the simulation results of the three presented *VCO* topologies

Table 4.2: Comparison between the *PMOS*-Load *3*-stage Ring *VCO*, the *3*-stage Ring *VCO* with interpolation and positive feedback and the *4*-stage *VCO* with interpolation, positive feedback and additional Tail capacitances.

	***PMOS*-Load *3*-stage**		***3*-stage with Interpolation and Positive Feedback**		***4*-stage with Interpolation and Positive Feedback**	
	41MHz	980MHz	43MHz	1.295GHz	42MHz	867MHz
Tuning Range	983MHz (1:24)		1.252GHz (1:30)		825MHz (1:20)	
Differential Peak Swing	0.4V	0.8V	0.43	0.1	0.52	0.37
Max Sensitivity	$1.788X10^9$		$3.377X10^9$		$5.051X10^9$	
Power Dissipation	-41.11dB (77µW)	-27.1dB (1.95mW)	-20.95dB (8mW)	-22.96dB (5mW)	-20.97dB (8mW)	-20.84dB (8.2mW)
Signal Level	-23.9dBm	-7.33dBm	3.516dBm	4.13dBm	-9.916dBm	-14.92dBm
Output Phase Noise *@1MHz* offset	-101.5dBc/Hz	-84.55dBc/Hz	-110.4dBc/Hz	-91.87dBc/Hz	-121.1dBc/Hz	-105dBc/Hz
Frequency Pushing (max-min)	4MHz	54.2MHz	5.5MHz	219MHz	3.24MHz	131MHz
Figure of Merit (*FOM*)	171.5dB		174.79dB		179.59dB	

It is clear that the design of the four-stage ring oscillator that employs interpolation, additional tail capacitance and positive feedback has a robust performance that is constant throughout the tuning range. It also satisfies the phase noise requirements of our system design of the TV Tuner Receiver with appropriate power dissipation, output voltage swing and without using any inductors which means that the area of the chip is quite small. Without a doubt the design of the four-stage Ring *VCO* is by far the best of them as it optimizes between power dissipation, phase noise, tuning range and output swing. That is why we have chosen this topology to be employed in our *TV* Tuner Receiver.

4.6 Conclusion:

In this chapter, we have explained how a circuit can maintain oscillation using a feedback view of oscillators. Then, we discussed how this oscillating output can have a tunable frequency that is a function in the control voltage applied. After that, we presented three different *LC VCO* topologies with their design procedures and made a brief comparison between them. It was proven that for our *TV* Tuner Receiver the *PMOS* cross-coupled *LC* oscillator was the best one but we had to use multiple *VCOs* in parallel to cover the entire range of frequency operation (*48MHz* to *862MHz*). As a result, we switched our attention to Ring *VCOs* and implemented three distinct topologies where all of them were able to cover the entire tuning range with a single *VCO* but with much higher power dissipation than narrow range *LC VCOs*. It was also proven that only the four-stage Ring *VCO* satisfied the requirements of phase noise. It also has appropriate power dissipation, output voltage swing and does not use any inductors making the area of the chip quite small. Consequently, we have chosen this topology to be employed in our *TV* Tuner Receiver.

5 Frequency Dividers:

Frequency divider is a basic block in *PLL*. It is used in the feedback path to divide the output frequency by a certain ratio, called *modulus*, to be compared with the input reference frequency. This allows frequency multiplication which is a main role for the *PLL*, especially when it is used as a frequency synthesizer.

Frequency divider design in frequency synthesizers encounters some challenges arising from the fact that it is the responsible block for frequency determination. These parameters are imposed by frequency planning in order to satisfy system requirements. That's why frequency planning must be performed along with frequency divider design in order to guarantee an implementation of a frequency divider that is capable of achieving the assigned tasks by the frequency plan. The challenges of this kind are; maximum operating frequency, maximum divide ratio or modulus, modulus range of variation and modulus variation steps. Also some tradeoffs appear between these parameters and traditional analog circuits' parameters, such as power consumption, area, number of transistors, response time…etc.

In this Chapter we discuss the different aspects of frequency dividers design. We start in Section 5.1 by introducing the possible approaches for the basic building block of the divider, which is the divide-by-2 circuit, and discuss the pros and cons of each of them. In Section 5.2, we explain some techniques that enable modifying circuits in order to reach modulus other than *2* and to change it if needed. Then, in Section 5.3 we present the most popular and commonly used frequency divider which is called "*Pulse Swallow Divider*". We will study its theory of operation and show how it can be modified to reach the most beneficial results. Finally, in Section 5.4, we use all the preceding information to design a frequency divider for the *PLL* of a TV tuner according to *ASTC* standards.

5.1 Divide-by-2 Frequency Divider Design Strategies

The divider blocks in the feedback of the *PLL* can be implemented either by a digital logic approach or by an analog approach. Each approach of these has some abilities and some limitations. Each one of them includes different categories. The basic advantage of the analog approach is its extremely high frequency, in the range of tens of Giga Hertz. However, its circuits are very expensive and power consuming. Since in this work we are more interested in the low frequencies wideband challenges, we won't discuss this approach. Instead, we will focus our work on the digital approach and its different categories, and analyze their operating properties, in order to end up with a suitable decision about which type(s) we are going to use in our design. The main parameters of the different divider topologies that we are going to make our decision based on are; input and output swings, input capacitance, maximum and minimum speeds, power dissipation in addition to some other parameters.

Although the frequency synthesizer is part of the RF circuit, which is usually analyzed by the analog circuit theory, it is most widely analyzed as a digital logic circuit. That's because it is much easier to be analyzed and simpler to be designed in digital domain. The digital logic approaches can be further divided into two categories; static logic (*CML*) and dynamic logic (*TSPC*).

The main idea of frequency division in digital logic approaches, both static and dynamic, is based on the negative feedback edge triggered *master-slave D flip-flop* (*D-FF*) divide-by-2 circuit, shown in Figure 5.1. At any instant, either the master latch (the first) or the slave latch (the second) is activated. When clock is low, the master latch is activated and the slave latch is disabled, so the input *D* propagates through the master latch while the slave latch latches its old value, and when clock is high, vice versa.

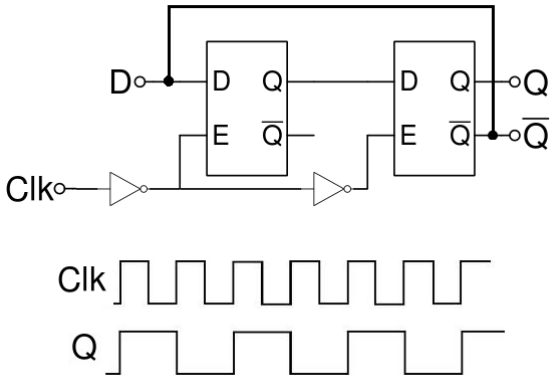


Figure 5.1: *D-FF* as Divide-by-2

When connecting the inverted output of the slave latch to the input of the master latch, as shown in the figure, the input *D*, and hence the output *Q*, is inverted once every clock rising edge, i.e., once every one clock cycle. This means that every two clock cycles, the output *Q* makes one cycle and hence the input clock frequency is divided by two.

The question now is how to implement this *D-FF* on circuit level. Implementing it by drawing its gate level circuit, shown in Figure 5.2, then compensating for each gate by its equivalent traditional *CMOS* digital circuit, results in a large circuit with many gate levels. For example, implementing the shown *D-FF* circuit in Figure 5.2 which consists of *8 NAND* gates; each one consists of 4 transistors, this gives total number of transistors equals *8×4 = 32* transistors per one *D-FF*. Since this *D-FF* is the basic building block for the divider, which means that it may be used profusely, therefore this large number of transistors and gate levels is very critical. Therefore, we tend to use other methods for implementing this *D-FF*.

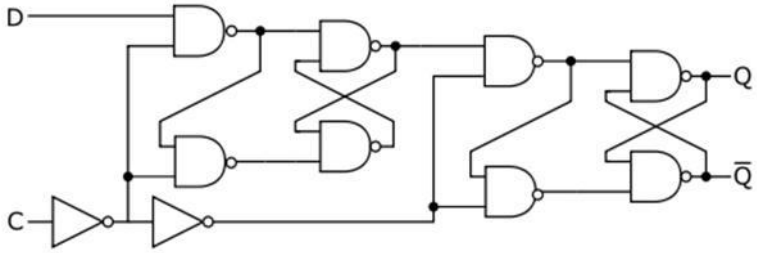


Figure 5.2: *D-FF* traditonal gate level circuit

5.1.1 Current-Mode Logic (CML)

Also known as source couple logic (SCL), the static logic *CML* is the most widely used topology in implementing the *D-FF* of the frequency divider. This is due to its combination between relatively high speed of analog circuits and simplicity of digital logic circuits. *CML* is a general approach that can be used for implementing any digital logic circuit, where it is capable of implementing the universal logic gates, *NAND* and *NOR*, as shown in Figure 5.3.

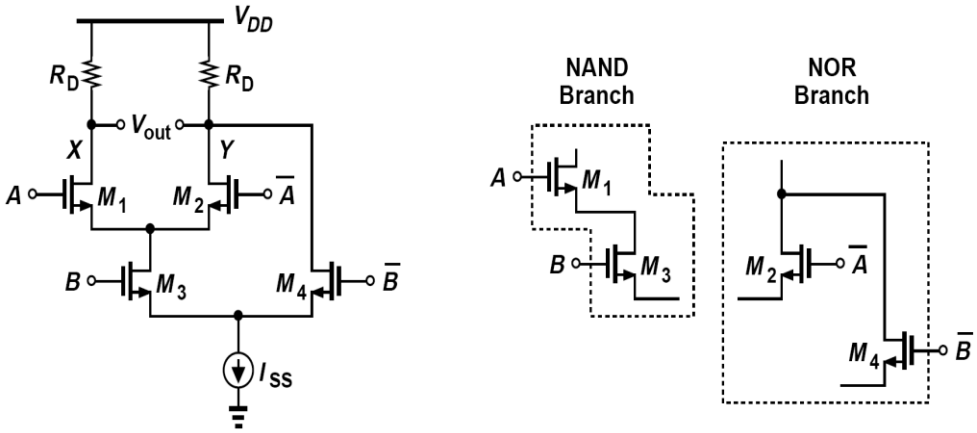


Figure 5.3: *NAND* gate implementation in *CML*

As shown in the figure, *CML* circuits provide differential outputs, which allows natural inversion, i.e., a single stage of the above circuit can serve as both *NAND* gate and *AND* gate at the same time by reversing the output terminals, thus there is no need for *NOT* gates in *CML*. The other main advantage of *CML* circuits is their high speed, which results from the property that a differential pair can rapidly enabled and disabled through its tail current source.

One of the limitations of *CML* circuits, that may cause problems in many designs, is the moderate input and output swings. The output swing can be easily calculated from the *NAND* gate circuit in Figure 5.3. In case that *A* and *B* are both high, node *X* is pulled down to voltage $V_{DD} - I_{SS} R_D$, while node *Y* is pulled up to voltage V_{DD}. In case that any of *A* or *B*, or both, is low, node *X* is pulled up to V_{DD}, while node *Y* is pulled down to voltage $V_{DD} - I_{SS} R_D$. Thus, we can conclude that the output swing is in the range of $I_{SS}R_D$.

Another severe problem in *CML* circuits is Stacking. Stacking problem arises from the condition that all ON transistors must operate in sat region, like any other differential circuit. Satisfying this condition requires applying different common mode voltages for different inputs. For example, in the *NAND* gate shown in Figure 5.3, in order to keep the transistor M_3 in sat, the differential input *A* should have a common mode voltage larger than that of the input *B* by at least one overdrive value. One way to generate these different common mode values is shown in Figure 5.4. The input *A* swings between V_{DD} and $V_{DD} - I_{SS1} R_1$, while the input *B* swings between $V_{DD} - I_{SS2} R_T$ and $V_{DD} - I_{SS2} (R_T + R_2)$. The value of R_T can be chosen such that $I_{SS2}R_T$ is greater than or equal to one overdrive value. This solution seems to be very simple, but actually it imposed a limitation on the circuit generating *F*. The input F can't exceed the value $V_{DD} - I_{SS2} (R_T + R_2) + V_{th}$, to avoid M_5 and M_6 operating in triode.

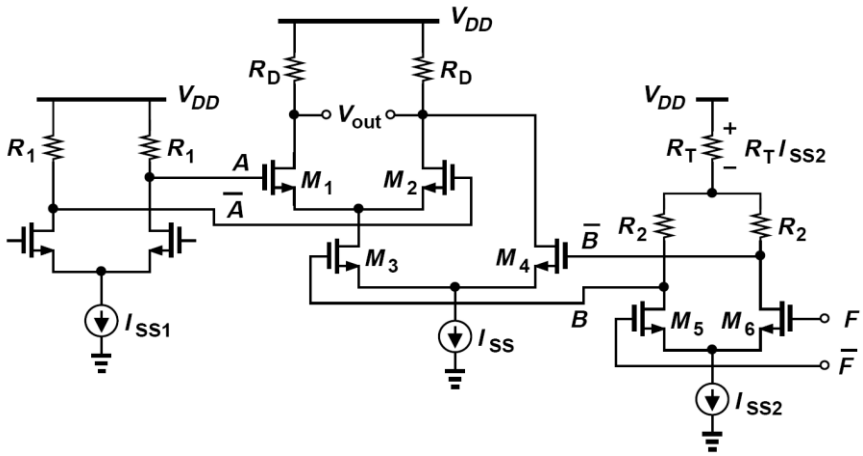


Figure 5.4: Compatible input common-mod generation for *CML NAND* gate

Having a logic circuit with multi stages of such gates, results in high common mode voltage requirements, which may exceed the available supply voltage itself. To be fair, this problem is not so critical in the implementation of the *D-FF* itself, as we are going to see in this section, but when considering the whole divider circuit which includes some switching and logic circuits, stacking problem will be very critical. One solution is to implement the *D-FF* only using *CML,* while the other parts of the digital circuit are implemented by *CMOS* logic. But in this case, *CML-to-CMOS* circuits are introduced at every interface between the *D-FF* and the *CMOS* logic circuits. Of course this is a very costly and unpractical solution due to much power and area wasted on these converter circuits.

Another solution is to implement the logic circuits based on the *CML NOR* gate instead of the *NAND* gate, in order to avoid stacking problem. *CML NOR* gate can be implemented without suffering from stacking, as shown in Figure 5.5. If the widths of M_1 and M_2 are greater than

that of M_3, then when either of the two inputs, or both, is high the current steers through the left branch pulling node X down to $V_{DD} - I_{SS1} R_D$, and node Y up to V_{DD}.

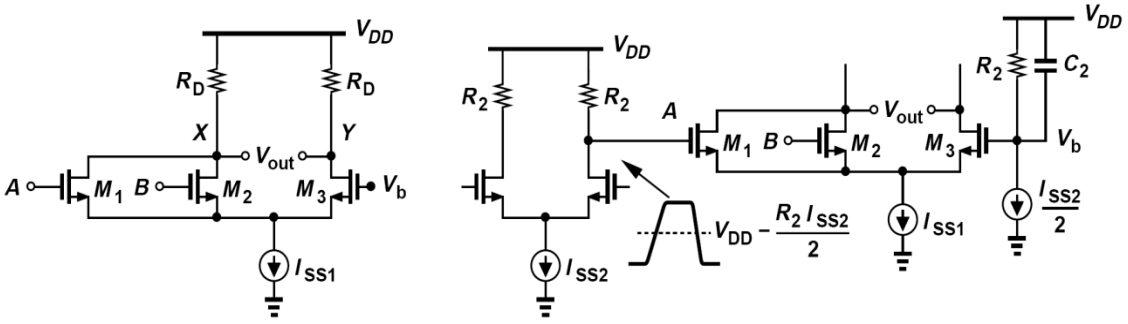


Figure 5.5: *CML NOR* gate (left) and its bias voltage generation (right)

Unfortunately this gate operates only with single ended inputs, which requires accurate generation for the input common mode voltage in addition to the bias voltage V_b. For example, let the input A is biased as shown in Figure 5.5 on the right, then its common mode voltage is $V_{DD} - I_{SS2} R_2/2$. Then, V_b also should have the same common mode value as shown in the figure. The capacitor C_2 is added to prevent supply voltage noises from reaching the gate of M_3, for accurate and stable biasing.

After this brief analysis for *CML* circuits and their properties in general, let's turn now to our main goal of this section, which is the implementation of the *CML D-FF*. First, let's start with the *CML* latch circuit shown in Figure 5.6 on the left. This circuit consists of an input differential pair $M_1 - M_2$, and a latch differential pair $M_3 - M_4$. When clock is high, M_5 is on (M_6 is off) steering the current in the input differential pair and enabling the sense mode, at which M_1 and M_2 sense the input difference and amplify it (inversely) to the output nodes X and Y, so that the output follows the input . When clock goes low, M_5, and hence M_1 and M_2, turns off disabling the sense mode, and M_6 turns on, steering the current through the latch differential pair and enabling the latch mode. In this mode, the circuit reduces as shown in Figure 5.6 at the middle, where the positive feedback amplifies the voltage difference between the two nodes X and Y. If the loop gain of this circuit is more than unity, it continues amplifying the difference between X and Y until they reach V_{DD} and $V_{DD} - I_{SS} R_D$.

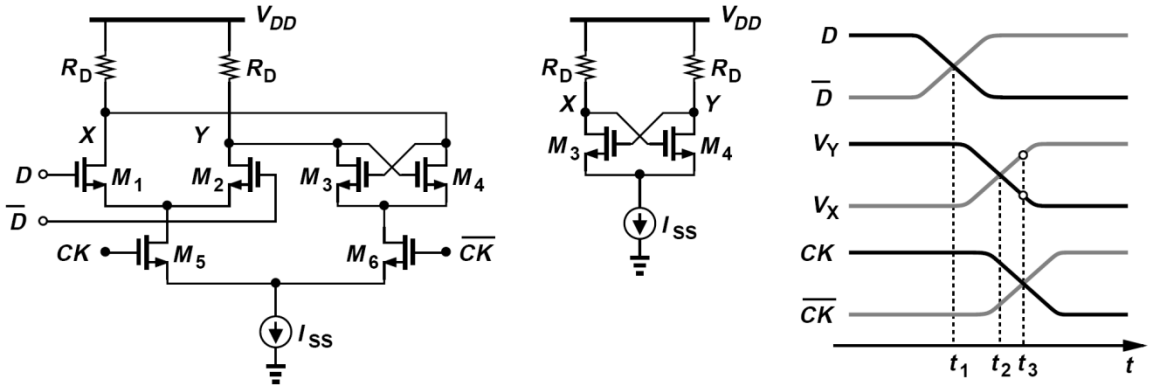


Figure 5.6: *CML* latch (left), *CML* latch at latch mode (middle), circuit's waveforms (right)

The high speed of this latch results from the self amplification of the circuit in the latch mode. For example, assume the waveforms shown if Figure 5.6 on the right, where the clock is initially high, i.e., the circuit is initially in the sense mode. At t_1, the input *D* is inverted, and then V_X and V_Y are inverted at t_2. After a very small time the clock turns low, at t_3 while V_X and V_Y didn't reach their full swing yet. Although they didn't reach their full swing at t_3, the positive feedback loop continues the amplification until reaching their full swing, if the clock period is long enough.

We can summarize the proper operation for the *CML* latch in one statement; during the latch mode, voltages must be amplified until reach their maximum swing. This imposes two conditions on two parameters; loop gain must be greater than unity, and amplification (or regeneration) time constant must be less than half the clock period (assuming *50%* duty cycle). In order to map these two conditions into equations and circuit parameters, consider Figure 5.7 showing the small signal equivalent circuit for a *CML* latch in latch mode.

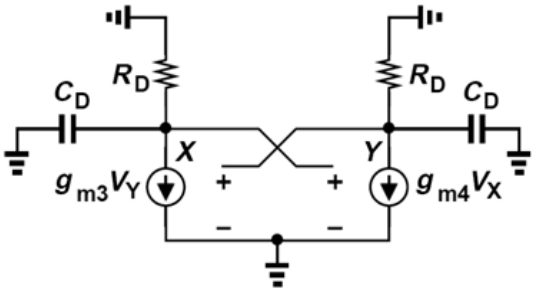


Figure 5.7: Small-signal equivalent circuit for *CML* latch in latch mode

Assume initially the difference between V_X and V_Y, V_{XY0}, was small enough, so that the small signal assumption is valid. C_D represents the total capacitance seen at nodes *X* and *Y*. Writing *KCL* at these two nodes gives:

$$\frac{V_X}{R_D} + C_D\frac{dV_X}{dt} + g_{m3,4}V_Y = 0 \qquad \text{Equation 5.1}$$

Similarly,

$$\frac{V_Y}{R_D} + C_D\frac{dV_Y}{dt} + g_{m3,4}V_X = 0 \qquad \text{Equation 5.2}$$

Multiplying both equations by $R_D\,dt$, and subtracting Equation 5.2 from Equation 5.1 we get:

$$-R_D C_D\, dV_{XY} = \left(g_{m3,4}R_D - 1\right)V_{XY}dt \qquad \text{Equation 5.3}$$

where

$$V_{XY} = V_X - V_Y \qquad \text{Equation 5.4}$$

then dividing both sides by V_{XY}, and integrating with the initial condition $V_{XY} = V_{XY0}$ at $t = 0$, gives:

$$V_{XY} = V_{XY0}\exp\left(\frac{(g_{m3,4}R_D - 1)t}{R_D C_D}\right) \qquad \text{Equation 5.5}$$

From Equation 5.5 we conclude that the condition for amplification of the difference between V_X and V_Y over the time is that:

$$g_{m3,4}R_D > 1 \qquad \text{Equation 5.6}$$

$$\therefore \quad loop\ gain = g_{m3,4}R_D \qquad \text{Equation 5.7}$$

Also we observe that V_{XY} increases exponentially with time with regeneration time constant:

$$\tau_{reg} = \frac{R_D C_D}{g_{m3,4}R_D - 1} \qquad \text{Equation 5.8}$$

For latches with high loop gain, i.e., $g_{m3,4}\, R_D >> 1$, the time constant is approximated to:

$$\tau_{reg} \approx \frac{C_D}{g_{m3,4}} \qquad \text{Equation 5.9}$$

Now suppose the *D* latch runs with a clock with period T_{ck} and duty cycle *50%*. In the latch mode, for V_{XY} to reach at least *90%* of its full swing ($I_{SS}\, R_D$) during time *0.5* T_{ck}, i.e.,

$$V_{XY} = V_{XY0} \exp\left(\frac{0.5\ T_{ck}}{\tau_{reg}}\right) \geq 0.9 I_{ss} R_D \qquad \text{Equation 5.10}$$

$$\therefore \quad V_{XY0} \geq 0.9 I_{ss} R_D \exp\left(\frac{-0.5\ T_{ck}}{\tau_{reg}}\right) \qquad \text{Equation 5.11}$$

Since V_{XY0} is the initial value in the latch mode, then it was the final value in the sense mode before clock falling edge, as shown in Figure 5.8.

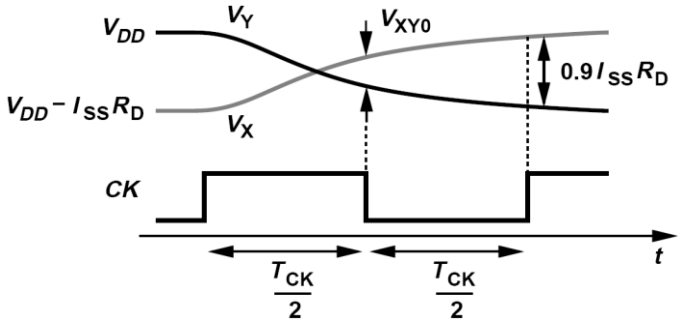


Figure 5.8: *CML D* latch waveforms

Thus, the condition is shifted to the sensing mode, such that at its end the worst case should give voltage difference V_{XY0} as specified in Equation 5.11. The worst case here is, as shown in Figure 5.8, when V_Y is maximum (V_{DD}) and V_X is minimum ($V_{DD} - I_{SS} R_D$) at the start of the sensing mode, and start changing according to:

$$V_Y = V_{DD} - I_{SS} R_D (1 - \exp(\frac{-t}{R_D C_D})) \qquad \text{Equation 5.12}$$

$$V_X = V_{DD} - I_{SS} R_D \exp(\frac{-t}{R_D C_D}) \qquad \text{Equation 5.13}$$

In order to satisfy the condition in Equation 5.11, we substitute for V_{XY0} by the difference between the two expressions in Equation 5.12 and Equation 5.13 at $t = 0.5\ T_{ck}$. After some mathematical work we reach this relation:

$$0.9 \exp\left(\frac{-T_{ck}}{2\tau_{reg}}\right) + 2 \exp\left(\frac{-T_{ck}}{2R_D C_D}\right) \leq 1 \qquad \text{Equation 5.14}$$

Now, we turn to the design phase of the divide-by-2 circuit using this *CML D* latch. First we connect two cascaded *CML D*-latches with the positive output of the first latch connected to the positive input of the second one. Then, connect the inverted output of the second latch to the input of the first latch as shown in Figure 5.9.

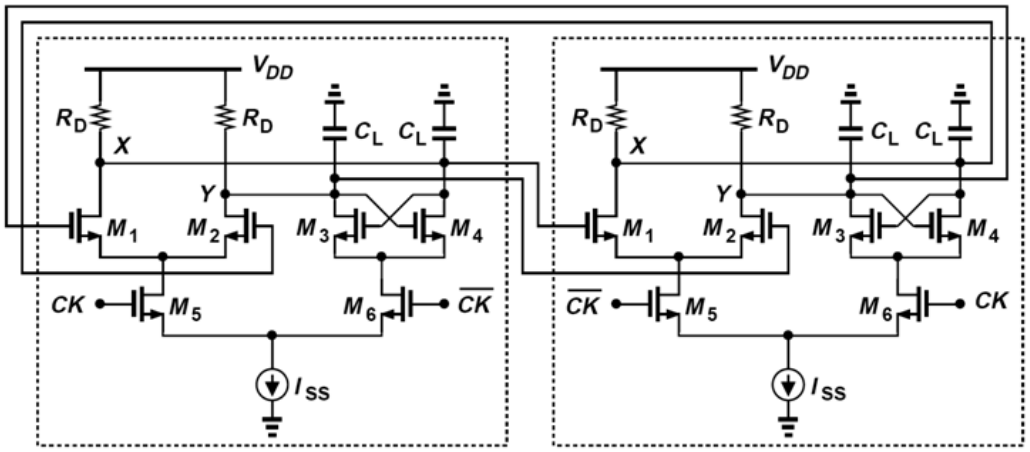


Figure 5.9: *CML* divide-by-2 circuit

C_L is the total output capacitance seen at nodes *X* and *Y*. Note that the clock is connected inversely to the two latches, i.e., in one half clock cycle the sensing circuit of one latch and the latching circuit of the other are enabled, and alternate in the second half clock cycle.

Before discussing the design notes of this circuit in details, let's sum up what our main goal is. Taking a look on the previous equations and conditions, we find out that we are looking forward to a circuit design such that; $R_D C_D$ value is small as much as possible and $g_{m3,4}$ is large as much as possible in order to allow higher frequency division.

When designing the components of this circuit, first we should satisfy the condition in Equation 5.6, so we tend to increase the resistance R_D and $g_{m3,4}$ by increasing the either the widths of the transistors M_3 and M_4 or the current source value. Fortunately, increasing the resistance R_D increases the output swing, but it reduces the headroom for lower transistors, which may cause one of them to operate in triode. Also the condition in Equation 5.14 sets a maximum limit for $R_D C_D$. In other words, as the value of $R_D C_D$ decreases, an input clock with lower period (higher frequency) can be used. So, when increasing R_D, we should be very careful for neither validating this condition nor getting one the transistors out of sat.

Concerning $g_{m3,4}$, increasing the widths of the transistors in the latching circuit in order to increase their transconductance, increases the value of the total capacitance at the nodes *X*

and *Y*. This increases the value of the regeneration time constant, which slows down the circuit and hence decreases the maximum allowable frequency. Also, the increase of C_D is limited by the condition in Equation 5.14. So, the optimum choice for the values of the widths of the transistors in the latching circuit as well as the resistance R_D, are those that satisfy the greater than unity loop gain condition.

On the other hand, increasing the value of $g_{m3,4}$ by increasing I_{SS} is a very efficient method, but just up to a certain value at which the limited widths of the transistors require more time for steering the full current, which decreases the maximum available frequency. Solving this problem we may widen the transistors slightly in order to steer the high current faster, but at the expense of the advantage of the small C_D value.

One method to increase the current steering speed between latching and sensing circuits, in order to increase the speed of the circuit, is to increase the input swing. However, this solution is very rigid, because, as we will see in section 5.2, frequency divider circuits consist of cascaded *D-FFs*, which means that the input of each *D-FF* is obtained from the output of the previous one. Thus, modifying the input swing of certain stage requires modifying the output swing of the previous one, which requires a unique design for every single stage.

One of the main problems encountered in the design of a *CML* frequency divider is the high power dissipation. The power dissipated in the circuit in Figure 5.9 is:

$$P_{diss} = 2\,V_{DD} I_{SS} \qquad \text{Equation 5.15}$$

A tradeoff arises here between the power dissipation and the maximum allowable frequency, because as we mentioned above increasing the current value helps operating at higher frequencies, but of course at the expense of the power dissipated.

Also another property in *CML D-FF* circuit is its relatively moderate size. Earlier in this section we talked about how severe is the large number of transistor in the traditional *D-FF* logic circuit. From Figure 5.9, we see that a single divide-by-2 unit contains 12 transistors only in addition to two current sources. Although *CML D-FF* saved in number of transistors, it didn't save much area, due to the presence of the resistance R_D in addition to the large widths of the transistors compared to traditional digital transistors' widths.

The last important note about *CML* divide-by-2 circuit is the dependency of its minimum required input swing with the frequency. To discuss this point, assume a divide-by-2 circuit with zero input swing. The current then is distributed equally over the two circuit branches and the transistors M_5 and M_6 are both on, consequently the sensing circuit and the latching circuit are both on at the same time. At this case the circuit reduces to the one shown in Figure 5.10.

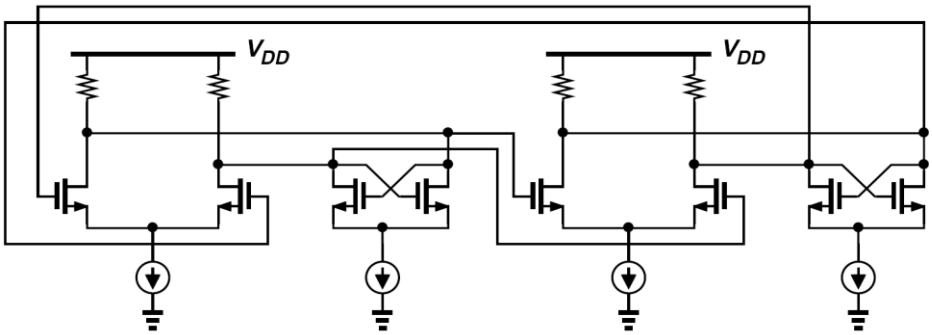


Figure 5.10: *CML* divide-by-2 circuit at zero input swing

This circuit is similar to a two stage ring oscillator circuit (Figure 4.53). This means that at zero input swing, the *CML* divide-by-2 circuit acts like a self oscillator with certain frequency $f_1/2$. If the input swing increased slightly, so that the transistors M_5 and M_6 are both still on at the same time, then the circuit operation would depend on the frequency of the input clock; if the clock frequency is close to f_1, then the circuit is injection-pulled by the clock and oscillates at half the clock frequency. While if the clock frequency is very different from f_1, then the circuit fails to divide by 2 and oscillates at its self-oscillating frequency $f_1/2$. As the input swing clock increases it becomes more able to injection-pull the circuit to oscillate at further frequencies than its self oscillating frequency (Figure 5.11).

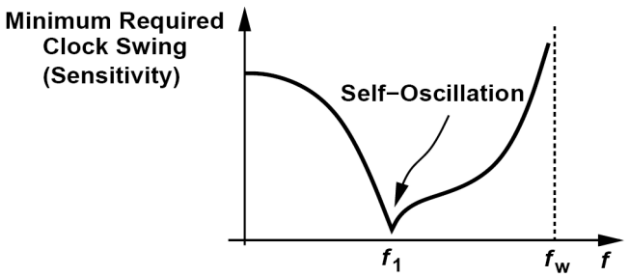


Figure 5.11: Relation between minimum required input swing and input frequency

A double edge-triggered (*DET*) *D-FF* can be implemented using *CML* technique by adding a *MUX* to the circuit which chooses one of the outputs of the two latches according to the input clock level as shown in Figure 5.12.

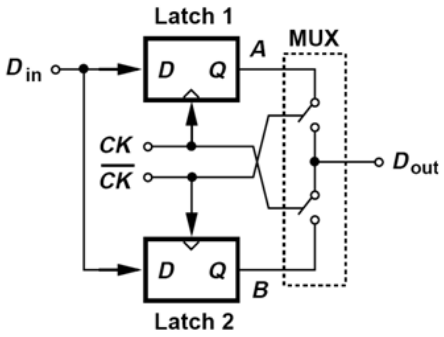


Figure 5.12: *CML* double edge-triggered *D-FF* block diagram (left) and transistor implementation (right)

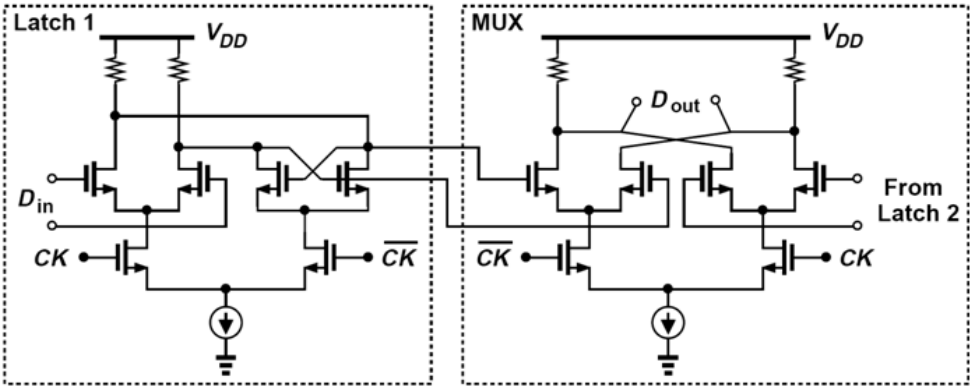


Figure 5.13: CML double edge-triggered *D-FF* transistor level

When connecting the feedback between D_{out} and D_{in} the circuit acts as a divide-by-1 circuit! i.e., it just reflects the input frequency to the output as it is.

5.1.2 True Single-Phase Clock (TSPC)

TSPC logic, a type of the *Clocked CMOS* (*C²MOS*) family, is considered the most efficient dynamic topology used in implementing high speed latches and flip-flops. It is mainly developed and used for latching in microprocessors due its compatibility with traditional digital *CMOS* logic. It is characterized by its high speed and small number of transistors, in addition to low power dissipation like all other *CMOS* circuits. Being a pure digital logic, *TSPC* doesn't suffer from such analog stuff in *CML* like current steering and loop gain, so its analysis is much easier and simpler.

The idea behind *TSPC* logic is actually explained in the name *Clocked CMOS*, where a *TSPC* logic circuit is a normal *CMOS* circuit with output controlled by a transistor whose gate is connected to a clock input. If the clocked control transistor is an *NMOS*, the circuit passes its logic result to the output directly when clock is high, and latches its final result when clock is low. This type is called *n-C²MOS*. If the clocked control transistor is a *PMOS*, the circuit latches at the high clock level and passes the result at the low level, and it is called *p-C²MOS*. Both types are shown in Figure 5.14.

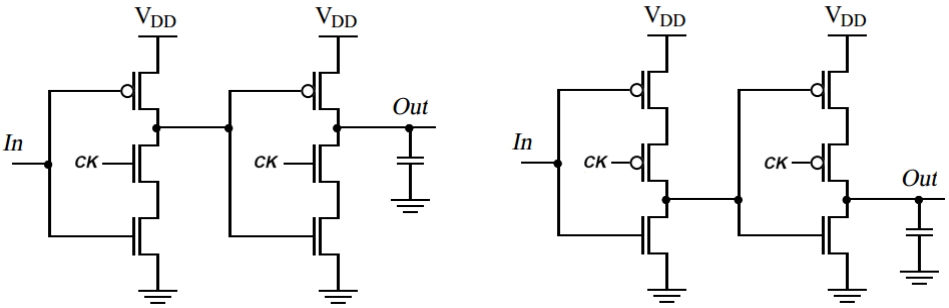


Figure 5.14: *n-C²MOS* buffer (left) and *p-C²MOS* buffer (right)

The name *True Single-Phase Clock* arises to refer to its advantage over the very similar *NORA* logic, or *No Race Logic*. This advantage is the usage of a single clock input signal, while in

NORA logic the clock and its logic inverse are needed at the same time, which requires an additional *NOT* stage that results in a small delay between the clock and its inverse.

In general, the logic functions in circuits incorporating *TSPC* latches can be implemented in one of two ways; either by inserting the logic circuits into the latches, as shown in Figure 5.15, or by inserting them between the latches.

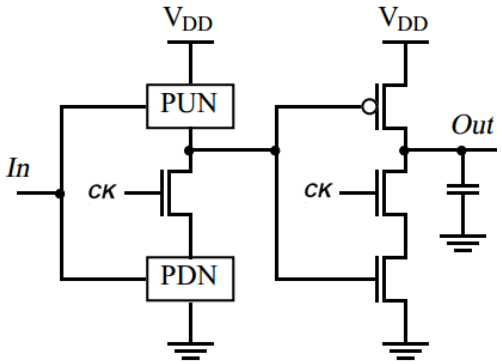


Figure 5.15: Logic circuit inserted into *TSPC* latch

A *TSPC* latch is typically one of those circuits in Figure 5.14, and it is made of two stages in order to prevent the input from propagation to the output. For example, consider the *p-C²MOS* latch that is made of single stage. Now suppose the following scenario. When clock is high and the output is high, the circuit is supposed to hold the output, but if the input changed from low to high the output would change from high to low. In other words, if the input is high the output is low (and vice versa for *n-C²MOS*) regardless of the value of the clock. When adding the second stage, the leaked output from the first stage would stop by the second stage.

Another improved circuit for *TSPC* latch is the so called "split output latch" shown in Figure 5.16. This circuit reduces the number of the transistors by one. In addition, its input capacitance, with respect to the clock input, is halved which helps in sharpening the clock edges. These improvements are at the expense of the voltage swing of the internal nodes, which causes the latch failing at relatively high frequency.

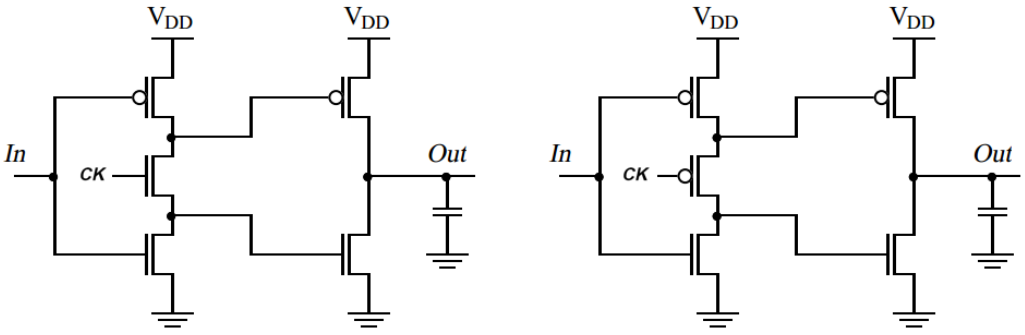


Figure 5.16: TSPC split output latch; *n-C²MOS* (left) and *p-C²MOS* (right)

Let's turn our concern now to how to implement the required *D-FF* using this topology. Traditional thinking may lead us to use two successive *TSPC* latch circuits with and connect them as in Figure 5.1, but of course this is not the optimum case. The first improvement we

can do here is to get rid of the clock inverter by using first latch of one of the two types; *n-C²MOS* or *p-C²MOS,* and the second latch of the other type. This ensures latching and sampling on opposite clock levels when having the same clock input. Secondly, remember that we added the second stage in the latch implementation in order to avoid the change in the output due to the change of the input to certain value (*1* in case of *p-C²MOS* and *0* in case of *n-C²MOS*) during the latching half clock cycle. On the other hand, in *D-FF* we do not have to keep passing the input for whole the time the clock is constant at certain level, so a stage in the middle between the two latches can be added to force the input of the second stage to be at the value that would not leak to its output *(force 0 if second latch is p-C²MOS and force 1 in case that second stage is n-C²MOS)* during the latching half clock cycle *(when clock level is low if second latch is p-C²MOS and when clock level is high if second latch is n-C²MOS)* and it passes the data from the first latch to the second latch only at the edge that turns the second stage from latching mode to sampling mode *(falling edge if second latch is p-C²MOS and rising edge if it is n-C²MOS)*, then it prevents passing the data from first latch to second latch during the sampling half clock cycle and holds on its current value. This stage can compensate for the function of the second stage of the two latches, leading to the *D-FF* circuits shown in Figure 5.17.

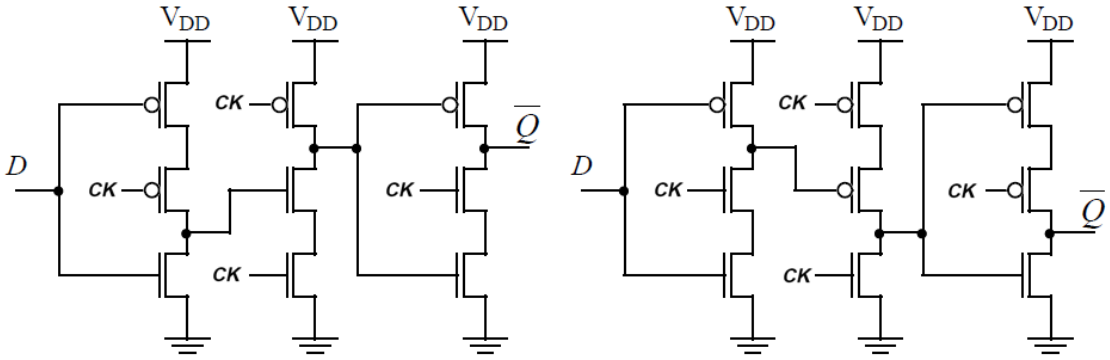


Figure 5.17: *TSPC* positive edge-triggered *D-FF* (left) and negative edge-triggered *D-FF* (right)

Another implementation for the *TSPC D-FF* circuit, shown in Figure 5.18, is dependent on the split output latches mentioned above. This circuit has the same advantages of the split output latches and suffers from the same problem.

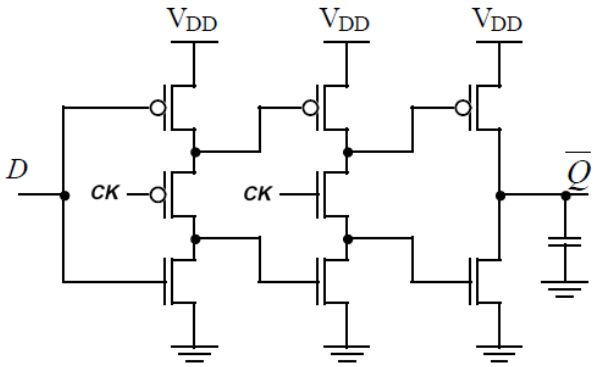


Figure 5.18: *TSPC* split output positive edge-triggered *D-FF*

The output of all these *D-FF* circuits follows the inverse of the input, so typically a normal *CMOS NOT* gate is added to the output to make it matches the input *D*. In order to convert a *TSPC D-FF* to a divide-by-2, all what we have to do is to connect the inverted output $\bar{Q}$ to the input *D*.

TSPC FF, like any other dynamic circuit, depends in latching on the transistors' parasitic capacitances, which are exposed to leakage after while, so this type of circuits is valid for relatively high frequency applications.

The main challenge in a *TSPC* divide-by-2 frequency dividers are the delay of the edge transitions. The delay of edge transitions at a certain node in a *CMOS* circuit is given by

$$\tau = RC \qquad \text{Equation 5.16}$$

where *R* and *C* are the load resistance and load capacitance seen at this node respectively. In order to decrease the delay we have to decrease the resistance *R* of the transistors of the same branch. This can be achieved by increasing their widths according to the relation

$$R_{triode} = \frac{1}{\mu_{n,p} C_{ox} {}^{W}/_{L} (|V_{GS}| - |V_{threshold}|)} \qquad \text{Equation 5.17}$$

The problem is that increasing *W* of the transistors of the current stage increases the input capacitance for this stage, according to

$$C \approx 0.5\, WLC_{ox} \qquad \text{Equation 5.18}$$

which increases the load capacitance of its preceding stage, resulting in increasing the delay of the later. The designer has to deal with this trade off wisely in order to reach the optimum case. For example, in the *TSPC* divide-by-2 circuit, we do not care much about its internal nodes' speed, all what we are looking forward to is a fast output transitions with the input clock rising or falling edges. So, we may tend to decrease the load capacitance of the output node $\bar{Q}$ by decreasing the widths of the first stage transistors that the output drives through the feedback connection. Decreasing the widths of the transistors of the first stage, and hence increasing their resistances and delay, is not a very critical issue, because all what we need from this stage is to reach its correct full swing within half the clock period, which is usually safe enough for frequencies less than *1GHz*.

Like all *CMOS* circuits, *TSPC D-FF* does not allow a full current path between V_{DD} and ground except at transition moments, which highly saves power compared to *CML D-FF* or any analog technique. Since power consumption in *CMOS* circuits depends on frequency, therefore we can calculate the power consumed by a single divide-by-2 circuit from

$$P_{diss} = CV_{DD}^2 f \qquad \text{Equation 5.19}$$

where *C* is the parasitic or load capacitance saving the voltage.

It is obvious from the presented *TSPC D-FF* circuits that their number of transistors (only *9*) and area is pretty small compared to the corresponding *CML* circuits, which is a big advantage over the *CML* circuits.

TSPC logic can also be used to implement a double edge-triggered *D-FF* by connecting shortening the outputs of two positive and negative edge-triggered *D-FFs*, as depicted in Figure 5.19. Like *CML DET* circuit, when the output $\bar{Q}$ of this circuit is connected to its input *D* the output frequency is the same as the input frequency.

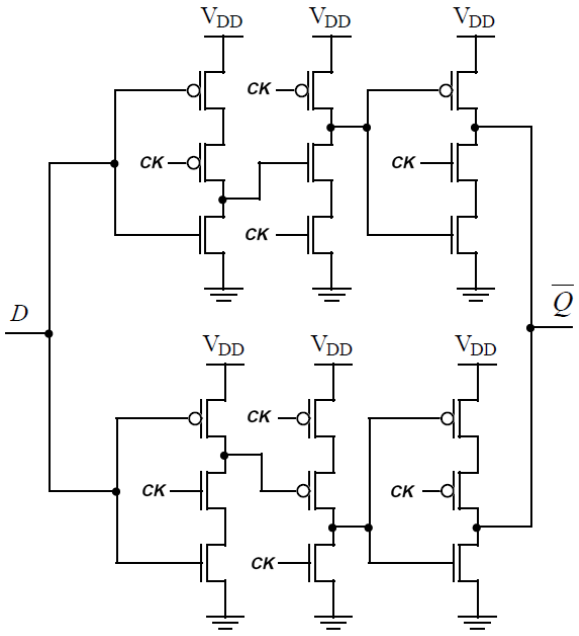


Figure 5.19: *TSPC DET D-FF*

5.2 Divide-by-M and Multi-Modulus Frequency Dividers

After implementing a divide-by-2 circuit, we want to use it to obtain a divider with any other integer dividing ration. Actually this part of the design is very creative and flexible. However, there are so many well known efficient techniques to achieve this goal, especially for the commonly used dividing values such as divide-by-3 and divide-by-5 circuits. Some other dividing values are not so common and it is rarely to be found in literature, but of course they can be obtained. The challenge here is not just about obtaining the dividing ratio (also called the *modulus*) itself, but it is also about obtaining it with the minimum hardware and the maximum speed.

Another important thing in frequency dividers design is how to obtain a divider that can divide by more than one divide ratio, called *multi-modulus divider*. This is very useful for frequency synthesizers in transceivers, where changing an input switch, called *modulus control (MC)*, changes the divide ratio and hence the output frequency of the *PLL*.

In this section we first discuss some well known divide-by-M circuits, in Subsection 5.2.1, and try to extract general techniques for dividing by any required integer. Then, in Subsection 5.2.2, we explain how to change these circuits into multi-modulus dividers by

adding some switching circuits and logic control. In this section we are going to use gate-level circuits, not transistor-level. The implementation of any *D-FF* or logic gate on the transistor level can be achieved using one of the techniques described in Section 5.1.

5.2.1 Divide-by-2^N, 2N & N Frequency Dividers

Let's start by the divide-by-3 circuit. As depicted in Figure 5.20, the divide-by-3 frequency divider consists of two *D-FFs* and an *AND* gate.

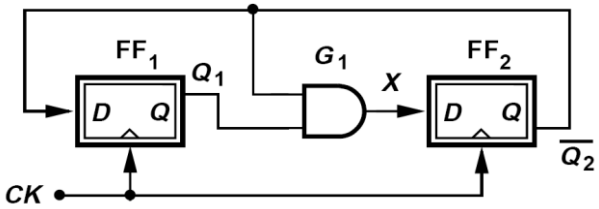


Figure 5.20: Divide-by-3 frequency divider

This circuit can be interpreted as a normal divide-by-2 circuit whose feedback is controlled by Q_1, so that when Q_1 is high the circuit divides by 2 and when it is low the *D* input of FF_2 and hence its output Q_2 are low ($\bar{Q}$ is high). Noting that Q_1 itself is the inverse of the previous Q_2, this means that each time the output Q_2 is low causes the node *X* to be low after one clock cycle, and hence, Q_2 to be low after two clock cycles. In other words, if output Q_2 is low at certain clock cycle, it forces itself to be low after two clock cycles. Also noting that the node *X* is the *AND* result of the inverse of Q_2 and its delayed version, then two successive zeros at the output Q_2 cause *X* to be one. In other words, two successive zeros on Q_2 forces it to be one in the next clock cycle. From the previous two notes, we conclude that the output Q_2 remains low for two clock cycles then turns high for one clock cycle, and then goes back to low for two clock cycles and so on. This allows *Q2 Q1* to act like a binary counter from *0* to *2* and avoid the state *3*.

The divide-by-4 operation can be obtained by one of two common methods; two successive divide-by-2 circuits, Figure 5.21, or two successive D-FFs with positive feed forward and negative feedback, Figure 5.22. The first method is very primitive; we simply divide by two twice.

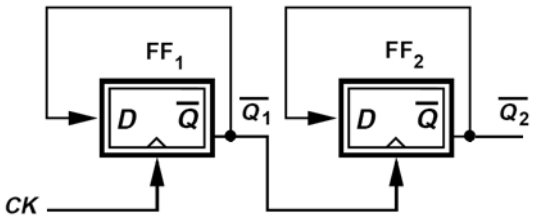


Figure 5.21: Divide-by-4 using 2 successive divide-by-2 circuits

The second method has same idea of a 2-bit feedback shift register, whose input is the inverse of its output. This results in Q_1 Q_2 sequence as shown in .

Table 5.1. Therefore the output Q_2 is inverted once every 2 clock cycles creating a divided by *4* output.

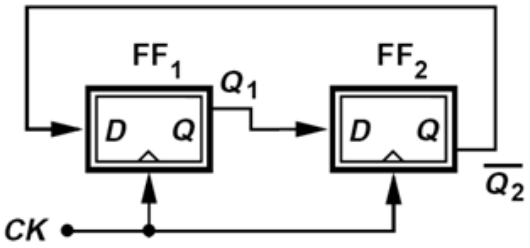


Figure 5.22: Divide-by-4 using 2 successive *D-FFs* with negative feedback.

Table 5.1: Output sequence of *D-FFs* of divide-by-4 using second method

Clock cycle	Q_1	Q_2
1	0	0
2	1	0
3	1	1
4	0	1
5	0	0

(Repeated)

In this part we try to categorize similar designs for frequency dividers and generalize some basic concepts and ideas in the design. So, let's turn to more generalized techniques that can be followed for a family of division ratios. First, we start by the most simple and primitive one, which is the **divide-by-2^N** family. This technique enables dividing by modulus 2^N using N cascaded divide-by-2 circuits. The divide-by-4 circuit shown in Figure 5.21 is the simplest member in this family. As the number of the cascaded divide-by-2 circuits increases by one, the modulus is doubled. This method is characterized by simplicity and hardware saving. Thus, it is mainly used in implementing binary counters, as we will see in section 5.3. The main problem here is that the *D-FFs* are not synchronized, which causes timing problems in digital circuits, especially when dividing by large numbers. Typically, for $N > 10$ the delay may exceed *1ns*. If the clock period is less than this value, then the outputs of the *D-FFs* of this circuit are not reliable and cannot be used in deriving a combinational logic circuit. That's because there is no instant at which they all are operating in same clock cycle. The severity of this problem arises clearly in the main design of our frequency divider in section 5.3, and possible solutions are mentioned there.

The second general family is the **divide-by-2N** which allows dividing by all even numbers. The divide-by-4 circuit using *D-FFs* with negative feedback is an already explained example for this technique. As the number of the *D-FFs* increases by *1*, the modulus increases by *2*. For example, a divide-by-6 circuit requires *3 D-FFs*; all forward connections are positive and only the backward connection is negative. Writing the output sequence of the *3 D-FFs* Q_1 Q_2 Q_3 in a divide-by-6 circuit, we get the sequence shown in

Table 5.2:

Table 5.2: Output sequence of divide-by-6 *D-FFs*

Clock cycle	Q_1	Q_2	Q_3
1	0	0	0
2	1	0	0

3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	0	0	0

(Repeated)

Here we see that the output Q_N remains zero for *N* clock cycles then turns high and remains high for another *N* clock cycles, having period *2N* times the clock cycle period. The absolute rule here is that, at any clock cycle the output of each *FF* is the output of the preceding *FF* in the previous clock cycle, except the first *FF*, whose output is the inverse of the output of the last *FF* in the train. This type has two advantages over the divide-by-2^N; first it is synchronized, second it is more general can divide by any even number not only powers of *2*. The disadvantage of this type compared to the divide-by-2^N is the relatively large number of *FFs*.

The previous two families of dividers have the advantage of *50%* duty cycle output. Unlike them, the third and the most general family, the **divide-by-N** frequency dividers, have output with duty cycle not equal to *50%*, usually *100/N%*. The idea of this type is to generate a pulse through an (N-1)-bit shift register, and let it propagate through its *FFs* until it disappears totally from the register, then generate another one, and so on. The divide-by-5 circuit shown in Figure 5.23 is an example for the divide-by-N family.

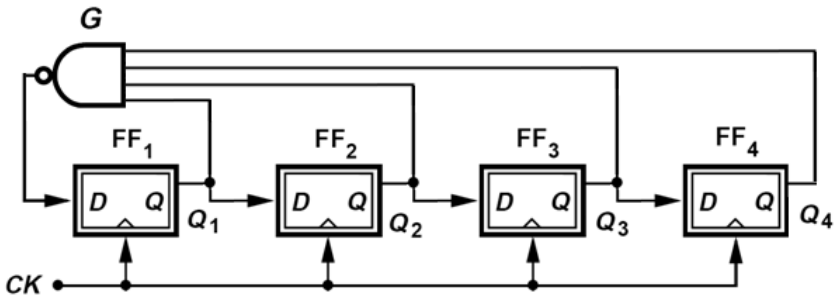


Figure 5.23: Divie-by-5 circuit with duty cycle *20%*

As the number of *FFs* increases by one, the modulus increases by one. The *NAND* gate is used here to generate the high pulse every five clock cycles. The output of first *FF* Q_1 is always *0* except if all the outputs, Q_1 through Q_4, were *0s* in the previous clock cycle, then the output Q_1 is *1*. This *1* propagates through the four *FFs* until it disappears after 4 clock cycles, then the *NAND* generates another *1*, and so on. The sequence of the outputs of the *D-FFs* is tabulated in

Table 5.3.

Table 5.3: Output sequence of divide-by-5 *D-FFs*

Clock cycle	Q_1	Q_2	Q_3	Q_4
1	0	0	0	0
2	1	0	0	0

3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	0	0	0	0

(Repeated)

It is obvious from the output sequence that the duty cycle is *20%*. In order to double this duty cycle to be *40%*, or *200/N%* in general, we remove the connection of Q_1 to the *NAND* gate. Thus, the *NAND* gate would generate two successive *1s* instead of only one *1*. Also we may use an *OR* gate instead of *NAND* to reverse the duty cycle, such that it becomes *80%* instead of *20%*.

Divide-by-N frequency dividers are characterized by flexibility. Their circuits are simple and modifiable. Also this technique allows dividing by any integer ratio, which is not available by the previous two techniques. Another advantage here is that all the *D-FFs* are synchronized. The only two drawbacks in this technique are the other than *50%* duty cycle and the large number of required *D-FFs*. One of the most important advantages of divide-by-N and divide-by-2-N family of dividers is that the output can be taken from the output of any *FFs* in the shift register, which makes sense because they all have same frequency but phases are different. This feature helps decreasing load capacitance and hence decreasing the delay in case of driving more than one circuit. For example, when the divider is required to drive other circuits that need its output frequency regardless of the phase, outputs can be distributed over the circuits from different *D-FFs*. Another usage for this feature is generating multi-phase output frequency. Actually, generating different phases using frequency dividers is a widely used process in many fields. One of them is generating in-phase and quad-phase components for IQ-based transceivers.

Using a circuit with double edge-triggered *D-FFs* allows dividing by half the value that the corresponding single edge-triggered circuit divides by. For example, if we replaced the single edge-triggered *D-FFs* in the divide-by-3 circuit in Figure 5.20 by double edge-triggered the circuit would divide by 3÷2 = 1.5. Figure 5.24 shows the divide-by-1.5 circuit and its output waveform.

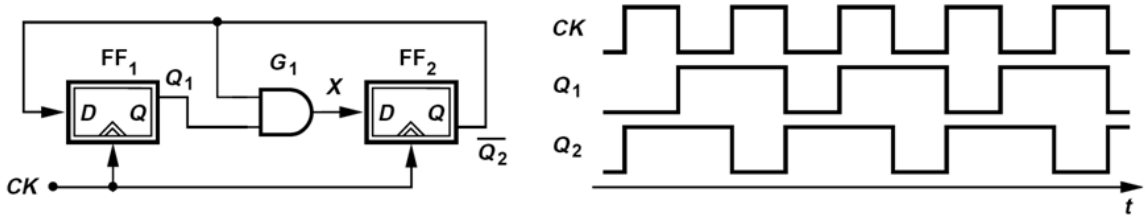


Figure 5.24: Divide-by-1.5 circuit (left) and its output waveform (right)

Before leaving this part we have to denote that all these dividing techniques should not make us forget that dividing by any number can be achieved by dividing by its divisors successively.

5.2.2 Multi-Modulus Frequency Divider

A multi-modulus frequency divider is a divider that can divide by more than single modulus. It should have input signals called modulus control that allow choosing the divide ratio. All frequency synthesizers require multi-modulus frequency dividers in order to allow selecting channels. The simplest form of these dividers is the *dual-modulus* frequency divider, which can divide by either one of two values according to a single input control. These dividers are usually named *divide-by-(N+1)/N*, which are used as *prescaler dividers* in *pulse swallow dividers* as we will see in Section 5.3. In fact, the pulse swallow divider itself is considered a multi-modulus divider, but due its complexity it will be covered in detail in next section. The main idea in the dual-modulus frequency divider is to merge the circuits of two dividers with two different dividing ratios, such that when the control signal is *1* the circuit reduces to one of the dividers' original circuit, and when the control signal is *0* it reduces to the other's circuit.

Let's start with the divide-by-2/3 circuit, shown in Figure 5.25. In this circuit when the modulus control *MC* is high, the output of the *OR* is *1*, so the inverted output of FF_2 is passed directly to node *X*, and thus the circuit reduces to a divide-by-2 circuit. When *MC* is low, the *OR* passes Q_1 to the *AND* gate as it is, reducing the circuit to the divide-by-3 circuit shown in Figure 5.20.

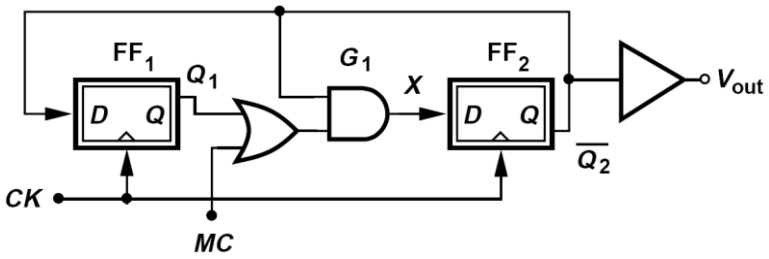


Figure 5.25: Divide-by-2/3 circuit

The buffer added to the output of the divider is to drive the following components in the circuit, in order to decrease the load capacitance at output of FF_2 in order to avoid delayed edges. That is because FF_2 is already driving the gate G_1 and the FF_1. Another circuit for the divide-by-2/3 that alleviates this problem is shown in Figure 5.26. The output in this modified circuit is taken from Q_1 which drives the NOR gate only. This circuit proved faster response than the former one by about *40%*. In this circuit when *MC* is high Q_2 is always *0*, thus the gate G_1 acts as a *NOT* gate for Q_1, so the circuit is again reduced to the normal divide-by-2 circuit. When *MC* is low the gate G_2 acts as a *NOT* gate for the inverse of Q_1. Actually, the result circuit is another form for the divide-by-3 divider. To prove this, assume the NOT gate G_2 is replaced by a bubble at the input of the gate G_1 and another bubble is added to other input of G_1 with replacing Q_1 by its inverse, then the *NOR* gate G_1 with two bubbles at its inputs can be replaced by an *AND* gate, which delivers us back to the original divide-by-3 circuit shown in Figure 5.20 with the *FFs* replaced.

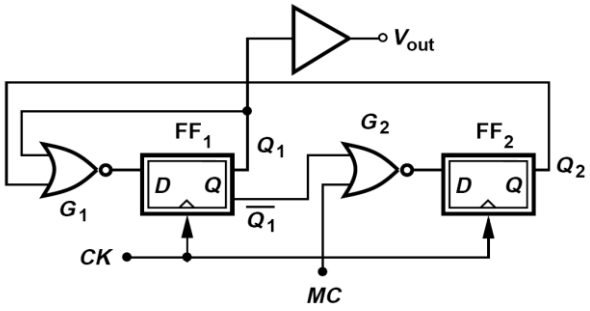


Figure 5.26: Fast divide-by-2/3 circuit

The second basic divide-by-N/(N+1) circuit is the divide-by-3/4 circuit, shown in Figure 5.27. The same concept of the divide-by-2/3 is applied here, with the difference that when *MC* is high the circuit reduces to the divide-by-4 circuit in Figure 5.22 instead of divide-by-2. While, when *MC* is low the circuit still reduces to the divide-by-3 circuit.

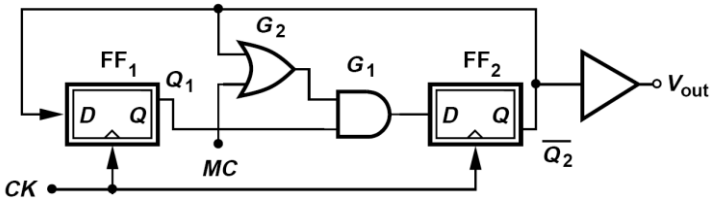


Figure 5.27: Divide-by-3/4 circuit

Here we introduce a new triple-modulus circuit that can divide by *2*, *3* or *4*. Combining the two circuits of divide-by-2/3 and divide-by-3/4, Figure 5.28 shows the obtained divide-by-2/3/4 circuit. When MC_1 is *0* the circuit reduces to the divide-by-2/3 circuit shown in Figure 5.25, i.e., divides by *2* if MC_0 is *1* and divides by *3* if MC_0 is *0*. On the other hand, when MC_0 is *0* the circuit reduces to the divide-by-3/4 circuit in Figure 5.27, i.e., divides by *4* if MC_1 is *1* and divides by *3* if MC_1 is *0*.

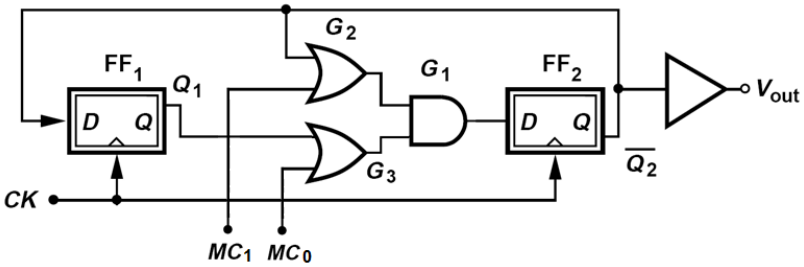


Figure 5.28: Divide-by-2/3/4 circuit

Note that when MC_1 MC_0 are *11* the output of the circuit will stuck, resulting in a constant output with zero frequency. The designer should be careful of this case. The following table shows the possible modulus control values and their corresponding divide ratios.

Table 5.4: Input control and corresponding dividing values for divide-by-2/3/4 circuit

MC_1	MC_0	Divide by
0	0	3
0	1	2
1	0	4
1	1	Invalid

This circuit can also be reduced to divide by *2* or *4* only. This is done by allowing only one modulus control input *MC* that is connected to one of MC_0 and MC_1 and its inverse is connected to the other. This will require an additional *NOT* gate to invert *MC*.

Another circuit we introduce here is the quad-modulus divide-by-2/3/4/5 circuit. Actually it can be generalized to be multi-modulus **divide-by-2/3/../N**. This concept exploits the divide-by-N family discussed in Section 5.2.1. We simply suggest adding an *OR* gate at the output of each of the *FFs* except the first one, as shown in Figure 5.29.

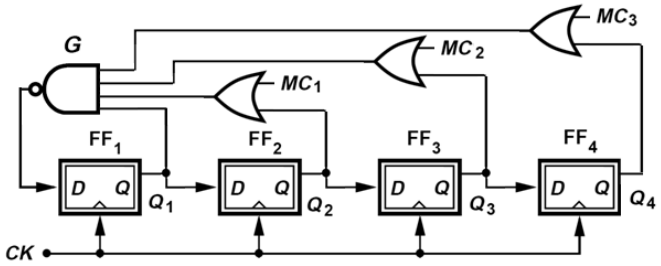


Figure 5.29: Divide-by-2/3/4/5 circuit

When $MC_3\,MC_2\,MC_1$ are *000* the circuit reduces to that in Figure 5.23 and divide by *5*. When MC_3 is *0*, FF_4 goes out of the scene, and the circuit divides by *4*. When MC_2 is *0* too, FF_3 also goes out of the scene and the circuit divides by *3*, and so on.

Note that if MC_3 is *0* neither MC_2 nor MC_1 can be *1*, because this will result in an output with varying frequency. For example, if $MC_3\,MC_2\,MC_1$ are *001*, the output sequence of the *FFs* will be as shown in Table 5.5.

Table 5.5: Variable output frequency of divide-by-2/3/4/5 circuit at $MC_3\,MC_2\,MC_1 = 001$

Clock cycle	Q_1	Q_2	Q_3	Q_4
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	1	0	1	0
5	0	1	0	1
6	0	0	1	0
7	0	0	0	1
8	0	0	0	0

(Repeated)

To avoid such logic errors in this type of dividers, we should first define the valid values for all *MC* inputs, and then design a switching circuit (decoder) that allows entering a binary

coded input that is decoded by this switching circuit to one of the valid values for the *MCs*. The following table shows the valid values for the *MCs* of a general multi-modulus divide-by-2/3/../N circuit.

Table 5.6: Valid modulus control input values for a divide-by-2/3/../N circuit

MC_1	MC_2	MC_3	…	MC_{N-4}	MC_{N-3}	MC_{N-2}	Divide by
0	0	0	…	0	0	0	N
0	0	0	…	0	0	1	N-1
0	0	0	…	0	1	1	N-2
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
0	0	1	…	1	1	1	4
0	1	1	…	1	1	1	3
1	1	1	…	1	1	1	2

Some cases other than those in this table may work, but they are special cases and they result in random dividing values which cannot be predicted easily, so it is not preferable to build a design based on these cases.

Although the most significant *D-FFs* may be not taking part in pulses' generation, the output can still be taken from them, because it propagates through all the *D-FFs* like a shift register. Despite being an advantage for the ability to drive more number of circuits without affecting the load capacitance or the speed, this causes much power consumption in vain. Switches may be added between the *FFs*, or at the clock or each *FF*, to prevent the signal from propagation through the nonessential *FFs*.

A similar concept can be applied for the divide-by-2N frequency dividers to obtain a general multi-modulus **divide-by-2/4/6/../2N** frequency divider. For a general divide-by-2N circuit, if the inverted outputs of all the *FFs*, including the first, are given to a *MUX* whose select signals are $log_2(N)$ modulus control input signals, and its output is fed back to the input of the first *FF*, then the divider can divide by values from *2* to *2N*, where *N* is number of the *D-FFs* in the circuit.

Also the divide-by-2^N family of frequency dividers we talked about in Section 5.2.1 can be modified in a similar way to obtain a multi-modulus **divide-by-2/4/8/../2^N** frequency divider. This can be achieved by applying the same method in the case of divide-by-2/4/6/../2N, which is to enter the outputs of all the *N FFs* to a *MUX* controlled by the input *MCs* and the output frequency is taken directly from the output of the *MUX*. This requires a very careful design for this *MUX* to maximize its speed without increasing its input capacitance so much, in order to avoid heavily loading the *FFs* themselves, especially because of their critical delay due to their asynchronous nature. The design of this *MUX* and its connections will be discussed in details in Section 5.3.

5.2.3 Divide-by-[m(2N-r)+kr]

The divide-by-[m(2^N-r)+kr] circuits are the most compact and flexible frequency dividers. Their idea is based on exploiting the multi-modulus frequency dividers discussed in the previous Subsection along with the binary counters or the divide-by-2^N circuits discussed in Subsection 5.2.1. The use of the divide-by-2^N circuits allows reaching large modulus values with the minimal hardware while the divide-by-m/k allows the fine tuning in order to reach any value between powers of *2*. The idea is to use a divide-by-m/k circuit, called *prescaler divider*, whose output is used to drive a binary counter of *N* bits, which is implemented by a divide-by-2^N circuit, and then the output bits of the counter are processed by a logic circuit to produce a single bit that is fed back to the modulus control of the prescaler divider. That is, if the feedback bit is *1* for $(2^N - r)$ counts, then the prescaler divider will divide by *m* for $(2^N - r)$ counts of the binary counter and it will divide by *k* for the rest *r* counts. This implies a total divide ratio *M* such that

$$M = m(2^N - r) + kr \qquad \text{Equation 5.20}$$

In order to illustrate the operation of this circuit clearly we are going to discuss two examples in detail in this Subsection, in addition to the pulse swallow divider discussed in Section 5.3, which is considered a multi-modulus divide-by-[m(2^N-r)+kr] frequency divider.

Let's start with the first example shown in Figure 5.30. In this circuit, the prescaler divider is the divide-by-2/3 circuit presented in Figure 5.20, the binary counter is FF_3 and FF_4 with output bits *A* and *B*, and the feedback logic circuit that generates the modulus control is the *NAND* gate G_2. That is, we chose *m = 2*, *k = 3*, *N = 2* and *r = 1*, and hence the modulus $M = 2(2^2 - 1) + 3 = 9$.

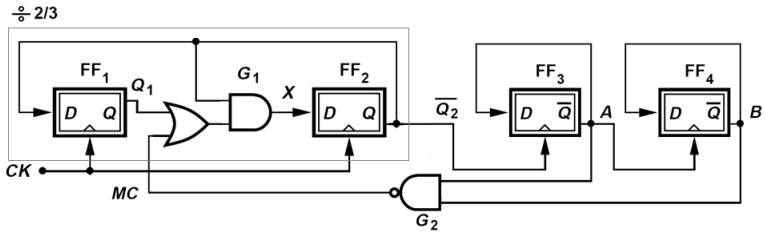


Figure 5.30: Divide-by-9 circuit

Assume the binary counter bits *A* and *B* are initially *0s*, and assume they are counting ascending. Then initially *MC* is *1* and hence the prescaler divider divides by *2*, resulting in *1* count for the binary counter every *2* input clock cycles. This case lasts for *3* counts, i.e., $3 \times 2 = 6$ input clock cycles, until the *AB* becomes *11*, then MC changes to 1 making the prescaler to divides by *3*. This case makes the counter counts *1* every *3* input clock cycles, and it remains for only *1* count before the counter bits *AB* revert to the initial case *00* returning *MC* to *1*. This means that the counter takes $3 \times 2 + 3 = 9$ input clock cycles until it reverts to the initial *00*, i.e., the period of the output bit *B* is *9* times the period of the input clock, therefore the output frequency equals the input clock frequency divided by *9*.

The following table shows the progress of the circuits' bits during *1* period of the output frequency. Note that the output of FF_3 changes with the rising edge of the inverse of Q_2, i.e., with the falling edge of Q_2 (at *1-to-0* transitions in the table). Also note that *A* is the least significant bit (*LSB*) of the counter and *B* is the most significant bit (*MSB*).

Table 5.7: Illustration of divide-by-9 circuit operation in one period of the output

Clock cycle	Q_2	A	B	Count	MC	Prescaler divides by
1	0	0	0	0	1	2
2	1	0	0		1	2
3	0	1	0	1	1	2
4	1	1	0		1	2
5	0	0	1	2	1	2
6	1	0	1		1	2
7	0	1	1	3	0	3
8	0	1	1		0	3
9	1	1	1		0	3
10	0	0	0	0	1	2

(Repeated)

The second example for the divide-by-[m(2^N-r)+kr] frequency dividers is the divide-by-15 circuit depicted in Figure 5.31. Here the prescaler divider is the divide-by-3/4 and the logic circuit generating *MC* is the *OR* gate G_3. The counter is the same 2-bit counter as the previous example. So, $m = 4$, $k = 3$, $N = 2$ and $r = 1$, thus the modulus $M = 4\,(2^2 - 1) + 3 = 15$.

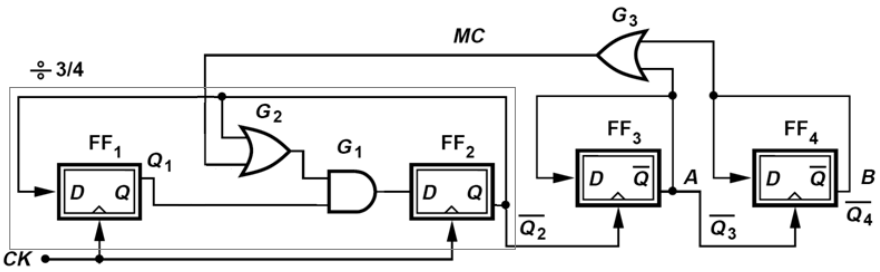


Figure 5.31: Divide-by-15 circuit

It is obvious that these circuits are very flexible and can be easily modified to obtain any required modulus. For example, replacing the divide-by-2/3 presecaler in Figure 5.30 by the divide-by-2/4 circuit discussed in Page139, would change the modulus into *10* instead of *9* ($M = 2\,(2^2 - 1) + 4 = 10$). Another example for the flexibility is to replace the *OR* gate in Figure 5.31 by a *NOR*, which results in swapping the values *m* and *k* so that the circuit divides by $M = 3\,(2^2 - 1) + 4 = 13$.

This type of dividers can be easily modified to be a multi-modulus divider itself by introducing an external control signal to the logic circuit generating the feedback bit. For example, in the divide-by-9 circuit, if the 2-input *NAND* gate G_2 is replaced by a 3-input *NAND* gate with the third input connected to an external modulus control MC_e. That is, when MC_e is *1*, the circuit operates normally as illustrated and divides by *9*. While when MC_e is *0*, it forces the original *MC* to be *1*, causing the prescaler to divide by *2* constantly, and hence the

circuit reduces to divide-by-8 divider. Thus, this small modification allows implementing a dual-modulus divide-by-8/9 circuit. Similarly, in the divide-by-15 circuit, if the 2-input *OR* gate G_2 is replaced by a 3-input *OR* gate with the third input connected to an external MC_e, the circuit will become a dual-modulus divide-by-15/16 frequency divider.

Being dual-modulus frequency dividers, the divide-by-8/9 and 15/16 can be used as prescaler dividers themselves in a larger frequency divider. This means that the concept can be implemented in a recursive way to reach much larger dividing ratios with maintaining the advantages of compaction and flexibility.

In order to perceive to what extent this method is very efficient, let's imagine combining the two circuits in Figure 5.30 and Figure 5.31 in a similar way to what we made with the divide-by-2/3/4 circuit in Figure 5.28. Then introduce two external control signals; MC_0 is connected to the *NAND* gate G_2 in Figure 5.30 and MC_1 to the *OR* gate G_3 in Figure 5.31, as mentioned above. For simplicity, replace this *OR* gate G_3 with a *NAND* gate, so that the two feedbacks resemble each other. Then the outputs of these two gates, G_2 and G_3, are controlled by two other *XOR* gates, to allow passing the outputs either directly or inverted. This requires other two external control signals MC_2 and MC_3 at the inputs of the additional *XOR* gates. The following table shows the possible sets of values for the four external control signals and the corresponding obtained modulus of the circuit. Note that there are *7* cases of the truth table of *MCs* are excluded due to being invalid.

This proposed circuit may encounter additional modification that allows the change the number of the bits of the counter *N* as mentioned in Subsection 5.2.2 and explained in Subsection 5.3.3. Allowing *N* to change from *0* to *2* will expand the range of the dividing values obtained by this circuit, which consists of only 4 *FFs* and logic gates, to include all integer values from *2* up to *16*!!

This technique may be generalized much more by replacing the dual-modulus prescaler by any general multi-modulus prescaler resulting in modifying the dividing ration in Equation 5.20 to be

$$M = \sum_{i=1}^{n} r_i m_i \qquad \text{Equation 5.21}$$

where

$$\sum_{i=1}^{n} r_i = 2^N \qquad \text{Equation 5.22}$$

Table 5.8: Valid sets of modulus control values and their corresponding moduli for divide-by-8/9/.../16 circuit

$M = m(2^2 - 1) + k$	m	k	MC_3	MC_2	MC_1	MC_0
8	2	2	1	0	0	0
9	2	3	1	0	0	1
10	2	4	1	0	1	1
11	3	2	1	1	0	1
12	3	3	1	1	0	0
13	3	4	1	1	1	0
14	4	2	0	1	1	1
15	4	3	0	1	1	0
16	4	4	0	1	0	0

The efficiency, flexibility and feasibility of using these higher order prescalers in these dividers are not studied deeply yet.

5.3 Pulse Swallow Divider (PS-Divider)

In the previous Section, we ended up with a general divide-by-[m(2^N-r)+kr] frequency divider, that is flexible and efficient. When we tried to modify it to be a multi-modulus divider by varying its two values *m* and *k*, we obtained great results that allowed for a modulus range of nine integers using only 4 *D-FFs*. In this section we intend to continue modifying this divider to be a multi-modulus by varying its other values; *r* and *N*.

5.3.1 Theory of PS-Divider and PS-Counter

A multi-modulus divide-by-[m(2^N-r)+kr] with $k = m+1$ and varying *r* is called a "pulse swallow divider". That is because a normal divide-by-m circuit followed by a 2^N counter is supposed to always count *1* every *m* input clock cycles, while the pulse swallow divider counts *1* every *m* clock cycles for only 2^N-*r* counts and during the other *r* counts it counts *1* every *m+1* clock cycles, so we say that it swallows *1* additional clock cycle for *r* counts. The commonly used notations to define a pulse swallow divider are *S*, *N* and *P*, where $S = r$, $N = m$ and $P = 2^N$.

A typical pulse swallow divider is shown in Figure 5.32. It consists of three main blocks; each block determines one of the three values of the divider. As we will see these three blocks can be reduced to two only; a prescaler and a program counter, which is the same form as a divide-by-[m(2^N-r)+kr], and that is why a pulse swallow divider can be considered a special case of the divide-by-[m(2^N-r)+kr] frequency divider.

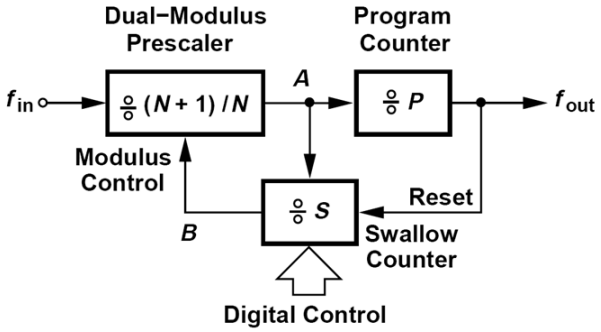


Figure 5.32: Pulse swallow divider block diagram

Although this circuit is characterized by having another counter as the feedback circuit that generates the modulus control of the prescaler, the theory of operation of this circuit is similar to those in Figures Figure 5.30 and Figure 5.31. The dual-modulus prescaler can divide by one of two values; *N+1* or *N*. Its output drives two counters; one of them counts from *0* to a constant value *P* and called "Program Counter", while the other counts from *0* to a certain value *S* that is determined by an external input and it is called "Swallow Counter". Initially, the modulus control signal is *0*, thus the prescaler divides the input frequency, f_{in}, by *N+1* for *S* times, and then the swallow counter inverts the modulus control signal causing the prescaler to divide by *N* for the *P-S* times before the program counter reaches it maximum count and resets its value and the value of the swallow counter as well, which reverts the modulus control to *0* and makes the prescaler divides by *N+1* again. This operation implies a modulus *M* such that

$$M = (N+1)S + N(P-S) \qquad \text{Equation 5.23}$$

$\therefore$

$$M = S + NP \qquad \text{Equation 5.24}$$

It is obvious that this Equation and Equation 5.20 are the same with substituting for *m* by *N*, *k* by *N+1*, 2^N by *P* and *r* by *S*.

The implementation of each of these three main blocks has been already discussed in the previous section. The prescaler can be any dual-modulus frequency divider whose two moduli are *1* integer different from each other; for example the divide-by-2/3 or the divide-by2-4. The swallow counter can be one of the multi-modulus frequency dividers whose moduli are also *1* integer form each other. The condition of *1* integer difference between the possible dividing ratios of both the prescaler and the swallow counter ensures a unity step dividing ratios for the pulse swallow divider, which means a covering all integers within certain range. The program counter block is typically a divide-by-2^N circuit that discussed in Section 5.2.1. Another condition arises here that is that all possible dividing ratios of the swallow counter should be less than the value *P* of the program counter in order to operate as specified above.

Here we introduce a standard efficient implementation for the PS-divider by merging both the program counter and the swallow counter into one block called "Pulse Swallow Counter", shown in Figure 5.33.

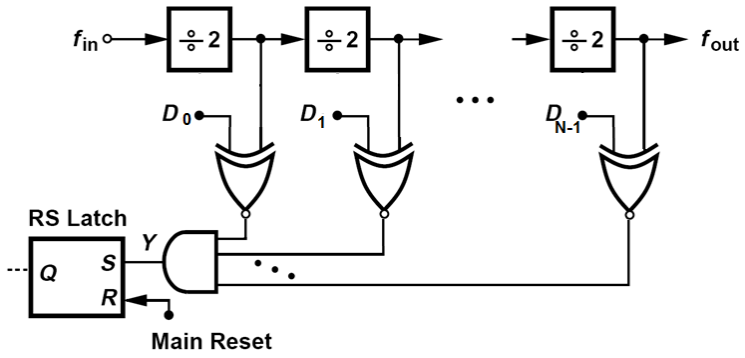


Figure 5.33: Pulse swallow counter circuit

The divide-by-2 blocks are those of the program counter itself. The input of this counter f_{in} is the output of the prescaler *(N+1)/N* divider, and its output f_{out} is the output of the PS-divider. The output of the RS-latch is fed back to the modulus control of the prescaler. D_0 through D_{N-1} are the frequency select inputs so that

$$S = D_0 + D_1 2 + D_2 2^2 + \cdots + D_{N-1} 2^{N-1} \qquad \text{Equation 5.25}$$

This input is compared to the output bits of the counter using the *XNOR* gates. The theory of operation is as follows. Initially the output of the RS-latch is *0* causing the prescaler *(N+1)/N* to divide by *(N+1)*. The counter starts with all bits reset and it counts up until the count equals the frequency select inputs, and then all the *XNOR* gates outputs *1* and hence the node *Y* turns *1* setting the RS-latch. At this point, the modulus control of the prescaler

becomes *1* causing the prescaler to divide by *N*. The prescaler continues dividing by *N* until the counter fills up and the reset signal of the RS-latch becomes low. The main reset comes from another *NAND* gate whose inputs are the output bits of the counter directly.

The main advantage of the pulse swallow counter is that it is generic, where *S* can take any value between *0* and 2^N*-1*, which covers all the valid values for *S* (from *0* to *P-1*). In addition, the pulse swallow counter saves power and area by compensating for the swallow counter with some logic gates and RS-latch. The limitation that the pulse swallow counter introduced to the system is decreasing the maximum allowable frequency for the counter to work properly. That is due to increasing the load capacitance of each divide-by-2 circuit, which delays the propagation of the clock through the counter. As we mentioned before a 2^N frequency divider is not synchronized, so the delay of the clock pulses is very critical and may cause errors in the frequency dividing value. To clear this problem, assume this case. We have a PS-divider with *P = 16* (*N = 4*) and the input *S = 2*. The following table shows the actual progress of the output bits of the pulse swallow counter of this divider.

Table 5.9: The effcet of the delay on the set and reset of pulse swallow counters

Clock cycle	Delay	Q_0	Q_1	Q_2	Q_3	Reset	Set	Latch Output	
0	-	0	0	0	0	1	0	0	
1	*1 DFF* delay	1	0	0	0	0	0	0	
2	*1 DFF* delay	0	0	0	0	1	0	0	E
	2 DFFs delay	0	1	0	0	0	1	1	
3	*1 DFF* delay	1	1	0	0	0	0	1	
4	*1 DFF* delay	0	1	0	0	0	1	1	E
	2 DFFs delay	0	0	0	0	**1**	0	**0**	**Error**
	3 DFFs delay	0	0	1	0	0	0	0	
5	*1 DFF* delay	1	0	1	0	0	0	0	

The second column in this table shows the delay from the start of the clock cycle until the count reaches the corresponding state. The delay in the last row of each clock cycle represents the delay that the counter takes until its count stabilizes on its correct value. We see that this delay changes the set and reset of the RS-latch randomly when the count passes by chance by the values *0* (reset becomes *1*) or *S*, equal *2* in this case (set becomes *1*). Some of these random errors have no effect because it happens when the output of the latch is already at the value as the error gives. Such errors are indicated in the table by the letter “E”. While the errors that actually matters are those that invert the value of the output of the RS-latch, indicated in the table by the word “Error” in bold. Figure 5.34 shows a case for a glitch of the set signal where the first three output bits changed to *1* before the clock propagates to the last one and changes it to *0*, so the output of the *AND* gate became *1* for a moment causing an error.

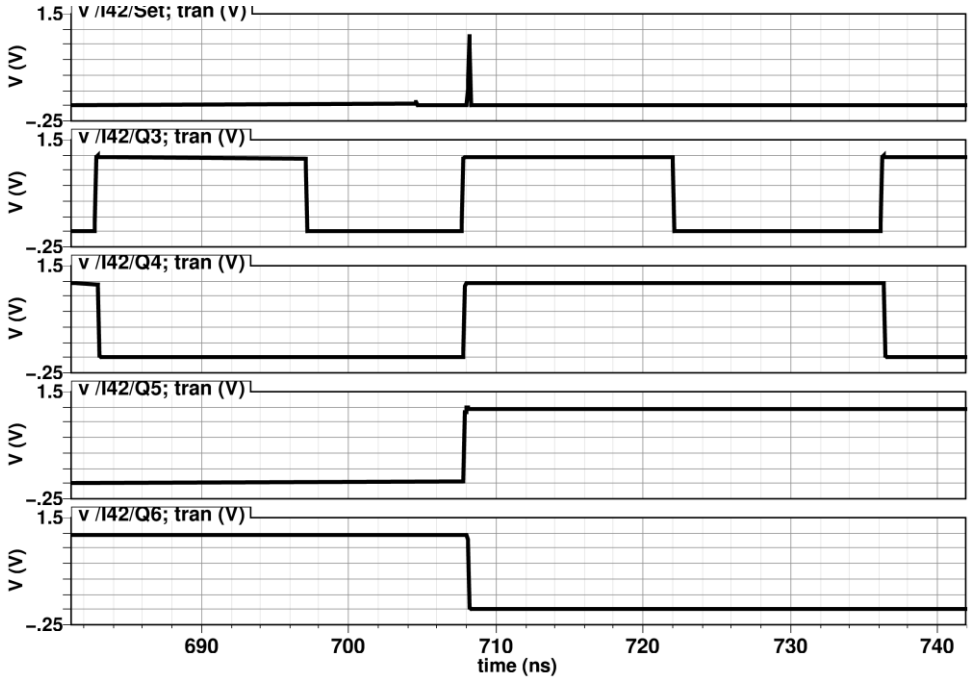


Figure 5.34: A glitch in the set signal due to asynchronization of *D-FFs*

A solution to this problem is to sample the output of the counter bits after certain time T_s from the rising edge of the input clock and enter this sample to RS-latch, where T_s must be more than the maximum delay in the counter. In simplest way, this can be done by entering the inverse of the input clock to two *AND* gates each with the other input is the output of one of the original *AND* and *NAND* gates, then sending the outputs to the set and reset of the RS-latch. That is, T_s in this method is equal to half the input clock period, so it is valid for counter with delay less than half the clock period.

Another problem may occur due to the same reason, which is the delay between *FFs*, is that the set or the reset of the RS-latch do not turn into *1* when they are supposed to do. This occurs when the maximum delay throughout the counter is more than the input clock period. In this case we have to either decrease the length of the counter or decrease the load capacitance seen by each *FF*, so that we keep the delay less than one clock period.

5.3.2 PS-Divider with (N+C)/N Prescaler

In some cases we may need to increase the difference between the dividing ratios of the prescaler to more than *1* integer in order to allow dividing by more numbers. This implies a dividing modulus *M* such that

$$M = (N + C)S + N(P - S) \qquad \text{Equation 5.26}$$

$$\therefore \quad M = CS + NP \qquad \text{Equation 5.27}$$

For example, assume a pulse swallow divider whose modulus $M = S + 3 \times 2^3$. This divider can divide by values from $M = 0 + 3 \times 2^3 = 24$ up to $M = 7 + 3 \times 2^3 = 31$, i.e., *8* possible dividing values. In order to increase this number we have to increase the value *P* of the program counter itself, but increasing *P* would cause changing the ratios between the dividing values.

This is considered a frequency planning tradeoff. This problem appears better when we use an example with frequencies. Assume we have a system whose channel spacing is *8MHz* in the range from *192MHz* up to *264MHz*, i.e., *10* channels. If we use a *PLL* with reference frequency *8MHz*, then we have to obtain dividing values from *24* up to *33* with step *1*. The divider in the above example can only reach *31* as a maximum dividing value. Increasing the value of *P* by *2* in order to increase the available values for *S* will result in shifting the range of the dividing ratios up to be from $M = 0 + 3 \times 2^4 = 48$ to $M = 15 + 3 \times 2^4 = 63$, i.e., 16 possible dividing values, thus we could double the number of possible dividing values. However, we now have a minimum dividing value of *48*, which should be used with the minimum frequency in the system *192MHz*, so we have to change the PLL reference frequency to be $f_{Ref} = 192MHz/48 = 4MHz$ in order to withstand this change. The second frequency channel *200MHz* now should be divided by a ratio that maps it to *4MHz*, thus it should be divided by *50*. As we can see, the step required now becomes *2* instead of *1* in order to map all channels to the same reference frequency, *4MHz*. This means that we still can divide by only *8* dividing values of the *16*, and that we lost the advantage of high *P*.

Using a prescaler with dividing ratios different from each other by *2* can solve this problem, where $M = 2S + 3 \times 2^4$. For *S* varying from *0* to *16*, we can obtain dividing values from *48* to *80* with step *2*. Now, using the *4MHz* reference frequency, we can map all the channels to *4MHz* using this range of dividing ratios. The disadvantage here is that we have to decrease the reference frequency, and thus increase the dividing ratio, which increases the noise effect, the power consumption and the area of the divider. As we will see later in this section, the variable-P pulse swallow divider solves this problem without having to decrease the reference frequency or shifting up the range of the dividing values.

Concluding some results at this point helps understanding the idea better and facilitates taking decisions during frequency planning and divider design. In order to increase the range of the values for the modulus of a PS-divider, we look forward to decreasing the ratio between the minimum dividing value and the maximum dividing value with keeping the ratio between any two successive dividing values constant. The ratio between two successive dividing values is

$$\frac{M_s}{M_{s+1}} = \frac{SC + NP}{(S+1)C + NP} \qquad \text{Equation 5.28}$$

The minimum and the maximum dividing values for the PS-divider are respectively,

$$M_{min} = NP \qquad \text{Equation 5.29}$$

$$M_{max} = (P-1)C + NP = P(N+C) - C \qquad \text{Equation 5.30}$$

Therefore, the ratio between them is

$$\frac{M_{min}}{M_{max}} = \frac{NP}{NP + (P-1)C} = \frac{1}{1 + \frac{(P-1)C}{NP}} \qquad \text{Equation 5.31}$$

Also $$\frac{M_{min}}{M_{max}} = \frac{NP}{P(N+C)-C} = \frac{N}{(N+C)-\frac{C}{P}}$$ Equation 5.32

Before deciding to change anything, we see from Equation 5.28 that changing *C* by factor *a* must be accompanied by changing either *P* or *N* by the same factor *a*, so that the ratio M_S/M_{S+1} remains constant. Similarly, changing either *P* or *N* by factor *a* should be accompanied either by changing the other one by the inverse factor *1/a* or by changing *C* by the same factor *a*. Now, in order to decrease the ratio in Equation 5.31, we have to either increase *C* or decrease *N* or do both. Unfortunately, these two changes are against each other and they both require increasing the value P in order to keep the ratio in Equation 5.28 constant. That is if we increased *C* by factor *a* and decreased *N* by factor *b*, then we would have to increase *P* by factor *ab*. However, we must first be sure that increasing *P* will not result in decreasing the ratio M_{min}/M_{max}, so that we did nothing. From Equation 5.32, we can see that increasing *P* in the ratio M_{min}/M_{max} at constant *N* and *C* is decreasing the ratio, which means that this modification goes along with our target.

Finally, to sum up this part in a simple neat statement we say that, in order to increase the range of the dividing values for a PS-divider, increase *C* by factor *a*, decreases *N* by factor *b*, and increase *P* by factor *ab*, where *a* and *b* are as much as possible. This would shift up all the dividing values by factor a, which requires decreasing the reference frequency by the same factor *a*, and that is what we mean by "as much as possible" here, because the reference frequency has a minimum limit, and the modulus *M* has a maximum limit to avoid large noise, large power consumption and large power consumption.

5.3.3 Variable-P PS-Divider

The purpose behind a PS-divider with variable *P* is to extend down the range of dividing values of an already existing PS-divider by decreasing its minimum dividing value. This can be done by holding up the most significant *D-FFs* in the counter of the divider from counting. One method to do that is to add switches either at the path of their input clock or at their feedback connected to input *D*. These switches can be implemented using logic gates as shown in Figure 5.35.

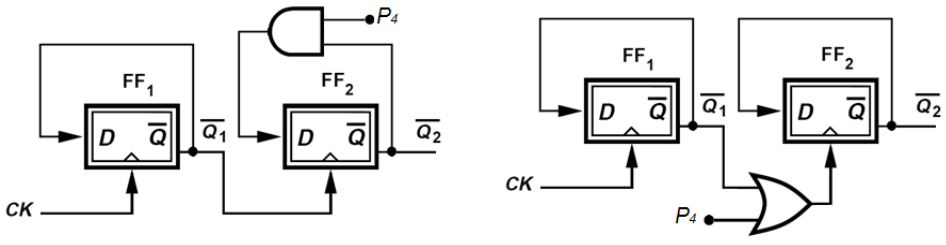


Figure 5.35: Variable-P PS-divider using logic gates at input *D* (left) and at input clock (right)

If the gate used for this purpose was an *AND* or a *NAND*, then the external control signal P_4 should be *1* to allow FF_2 to divide by *2* normally and the counter to be a 2-bit counter, and it should be *0* to hold FF_2 up so that the counter reduces to a 1-bit counter, and the vice versa when the used gate is *OR* or *NOR*. We prefer using the design on the left which disables the

divider through the input *D*, regardless of the gate used. That is because this design allows exploiting the advantage of the divide-by-2 circuit as being just a logic circuit by merging the gate with it, in order to decrease the delay and save area. The implementation of such a modified divide-by-2 circuit is shown below for both CML (Figure 5.36), and TSPC (Figure 5.37) circuits incorporating *NOR* gate. An important note here is that the first latch of those *D-FFs* is already inverting so this modification is equivalent to an *OR* gate, not *NOR*, followed by the *D-FF*.

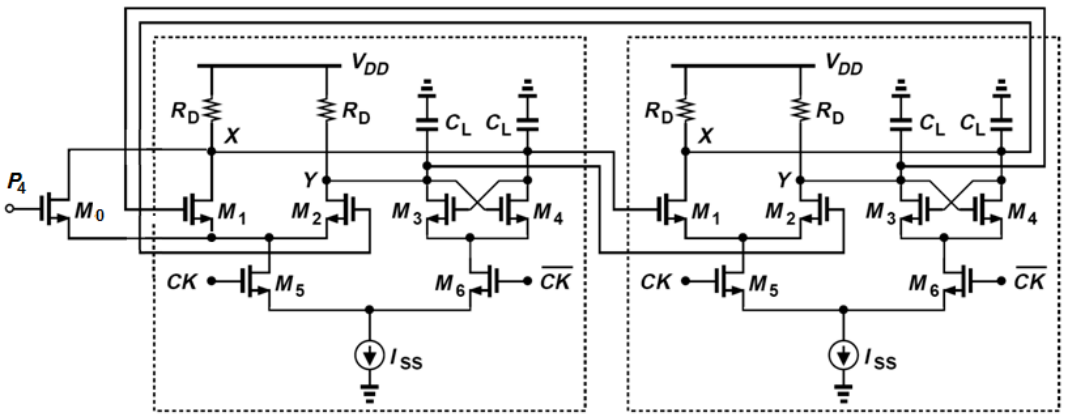


Figure 5.36: *CML* divide-by-2 circuit merged with a *NOR* gate

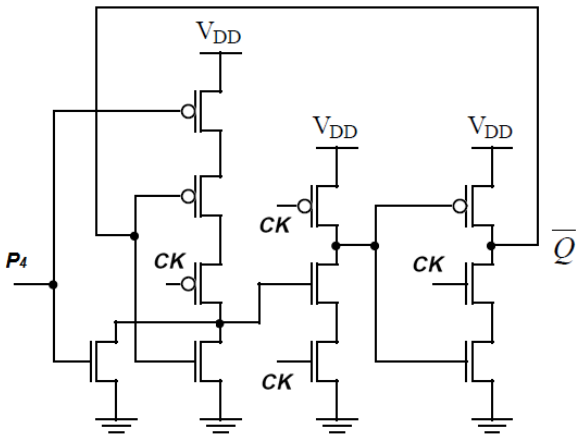


Figure 5.37: *TSPC* divide-by-2 circuit merged with a *NOR* gate

All what the previous modification for the divider's counter has done is that it disabled the unnecessary most significant *FFs* from counting, which means that their outputs would hold on *0* or *1* constantly. Thus, the output of the divider must be taken from the most significant enabled *FF*. That requires allowing multiplexing on the outputs of the *FFs* to choose one of their outputs according to the *FFs* enabled in the counter. Thus, a 1/2/4/8-bit counter circuit would be as shown in Figure 5.38. The select inputs of the *MUX* are a combination of the counter maximum count select inputs $P_4P_8P_{16}$, so according to the inputs $P_4P_8P_{16}$, the counter counts to 4, 8 or 16 as illustrated in Table 5.10

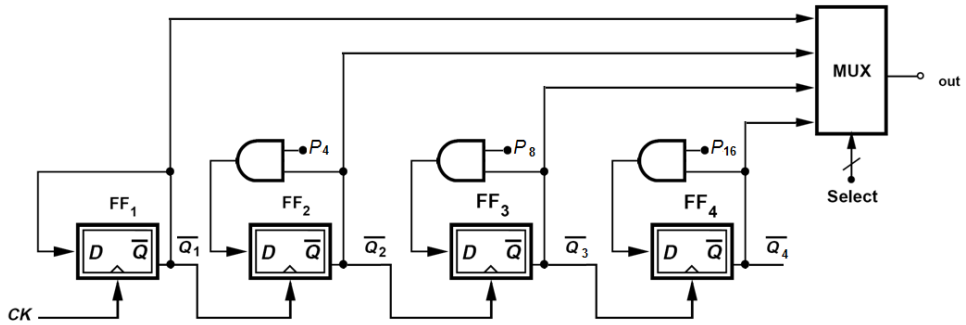


Figure 5.38: 1/2/3/4-bit counter circuit

Table 5.10: Possible values for the maximum-count select inputs and their corresponidng coounting bits

Counter is	P_4	P_8	P_{16}	MUX Select	Out
1-bit counter	0	0	0	00	$\overline{Q_1}$
2-bit counter	1	0	0	01	$\overline{Q_2}$
3-bit counter	1	1	0	10	$\overline{Q_3}$
4-bit counter	1	1	1	11	$\overline{Q_4}$

If the *AND* gates are replaced by *OR* gates, then all *1s* and *0s* in the *Ps* columns in this table are swapped.

5.4 PLL Frequency Planning and Frequency Divider Design

In this section, we explain the design of the frequency divider used in TV-Tuner frequency synthesizer according *ATSC* standards.

5.4.1 PLL Frequency Planning

First, let's enumerate the system specs that we should take into consideration during frequency planning. We have channels from *48 MHz* to *840 MHz* with channel spacing *6 MHz* with total number of *133* possible channels. It can be proven that as the modulus value increases (f_{Ref} decreases), the loop *BW* decreases (lock time increases) and the *VCO* phase noise is more effective. Thus we prefer starting the planning with maximum possible f_{Ref} which is the channel spacing, thus

$$f_{Ref} = 6\ MHz \qquad \text{Equation 5.33}$$

Therefore we have to design a frequency divider that dividing values are

$$M = \frac{f_{ch}}{f_{Ref}} = \frac{48\ MHz \xrightarrow{step=6MHz} 840\ MHz}{6\ MHz} = 8 \xrightarrow{step=1} 140 \qquad \text{Equation 5.34}$$

It is impossible to realize this wide range of dividing values using the normal PS-divider we discussed in Subsection 5.3.1. When trying to apply the modification of the PS-divider by introducing a divide-by-(N+C)/N prescaler as we did in Subsection 5.3.2, and starting to follow the rules we deduced there we reached the following. Initially we started by *C = 1* and decreased *N* as much as possible to be *1*. Then we chose the value of *P* that realizes the minimum required number which is *8*. Thus, we only got *8* possible dividing values

$$M = S + 8 = 8 \xrightarrow{step=1} 15 \qquad \text{Equation 5.35}$$

So we started increasing both *C* and *P* by doubling them, and halving the reference frequency, until we reached the values *C = 32* and *P = 256*, thus we got *256* possible dividing values

$$M = 32\,S + 256 = 256 \xrightarrow{step=32} 8416 \qquad \text{Equation 5.36}$$

This range of dividing values covers the range we want, but it reduces the reference frequency very much to be *6 MHz / 32 = 187.5 KHz*. This oscillator frequency is too small to be compatible with the designs of the *PFD*, *CP* and frequency dividers presented in this work. Also it results in a massive enlargement for the *VCO* noise. To sum up, this planning is not feasible for implementation.

The other solution is to increase the range of the dividing values without shifting it up or changing the reference frequency by using the variable-P PS-divider discussed in

Subsection 5.3.3. Initially, we chose *N* to be *1* and *P* that realizes the minimum required number which is *8*. This would give *8* possible divide values; from *8* to *15*. Then extending the length of the counter of the PS-divider so that *P = 16* would give *16* other dividing values; from *16* to *31*. Then extend it again to obtain other *32* dividing values, and so on, until we cover all the required range of dividing values. This will be at *P = 128*. The following table shows the frequency planning map.

Table 5.11: Frequency planning map for the PS-divider

P	PS-Counter bits	S			M = S + P			f_{vco} (MHz) = M × 6 MHz		
		From	Step	To	From	Step	To	From	Step	To
8	3-bit	0	1	7	8	1	15	48	6	90
16	4-bit	0	1	15	16	1	31	96	6	186
32	5-bit	0	1	31	32	1	63	192	6	378
64	6-bit	0	1	63	64	1	128	384	6	762
128	7-bit	0	1	12	128	1	140	768	6	840

5.4.2 Frequency Divider Design

First, we want to decide which technology we are going to use for the implementation of the required PS-divider. We started this part by implementing a divide-by-2 circuits of both the *CML* and *TSPC* topologies in order to compare between their results according to the following criteria; power consumption, area, maximum available frequency and simplicity of its compatible gates implementation. It is clear that *TSPC* topology is much simpler and easy to construct its compatible gates; just normal digital *CMOS* gates. Using the *130 nm CMOS* technology and *1.2* V_{DD}, here are the results we got.

CML:

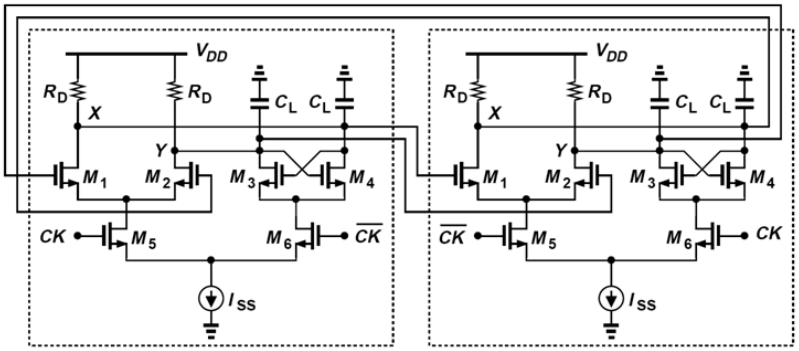


Figure 5.39: CML divide-by-2 circuit design

In order to obtain a *300 mV* output swing, we used R_D *= 1k* and I_{SS} *= 300* μ*A*. Therefore the power consumption from Equation 5.15 is

$$P_{diss} = 2\,V_{DD}I_{ss} = 2 \times 1.2 \times 300\mu = 720\ \mu\text{W} \qquad \text{Equation 5.37}$$

From Figure 5.39, number of transistors is *6× 2 = 12* plus additional transistors for the current source, therefore the total number is *14* transistors. The circuit starts oscillating (dividing) at M_3 and M_4 with widths: $2 \times 2\mu$.

When increasing the widths of the transistors M_1 and M_2, this circuit worked properly up to frequencies more than *8 GHz*. Figure 5.40 shows the input and output waveforms of a *CML* divide-by-2 circuit at *8 GHz*.

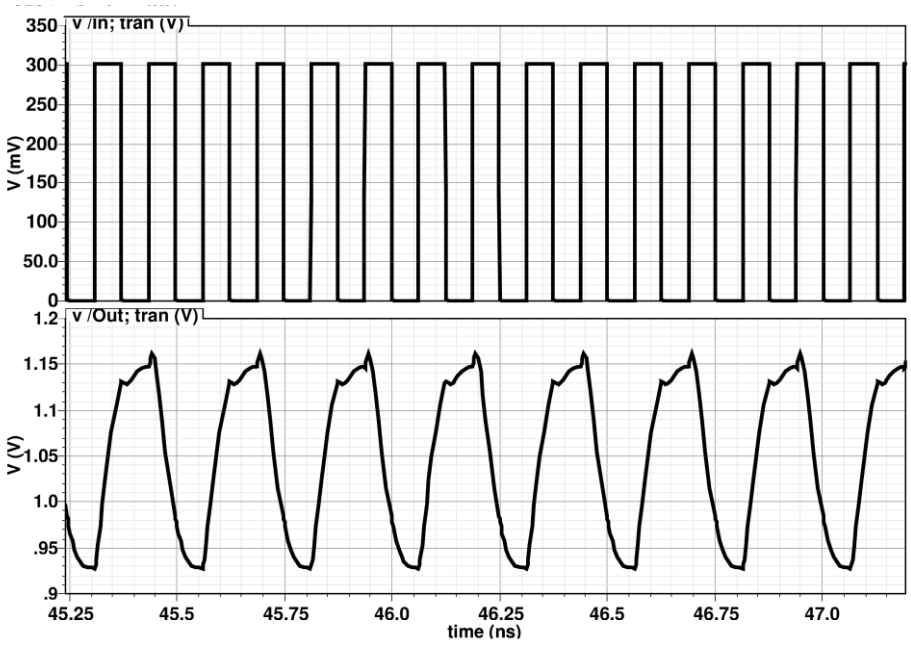


Figure 5.40: Input and output waveforms of *CML* divide-by-2 circuit at *8 GHz*

TSPC:

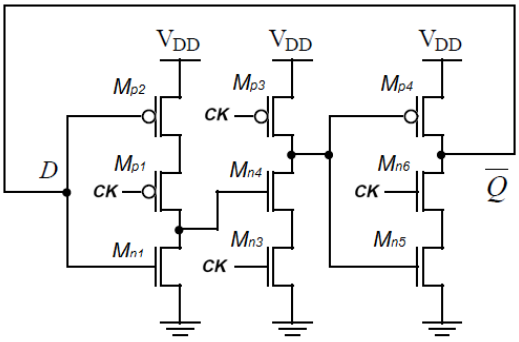


Figure 5.41: TSPC divide-by-2 circuit design

As, discussed in Subsection 5.1.2, we decreased the widths of the transistors of the first stage; M_{n1} and M_{p1} to be the minimum allowable width *150 nm* in order decrease the load capacitance of $\bar{Q}$ to sharpen its edge transitions. Therefore, the capacitances of M_{n1} and M_{p1} can be neglected with respect to the relatively high capacitances of the two transistors of the following *NOT* that $\bar{Q}$ drives, whose widths are 2μ and 4μ. From Equation 5.18, we can estimate the load capacitance at the node $\bar{Q}$ to be

$$C_L = 0.5 \times (W_n + W_p) L C_{ox} = 5.85\ fF \qquad \text{Equation 5.38}$$

Figure 5.42 shows the effect of loading the output of a divide-by-2 circuit by the clock of another one in a cascaded form. The upper curve us the output of the driving circuit and the lower curve is the output of the driven free output circuit. Note that the delay between their *600 mV* transitions is about *0.03 ns*.

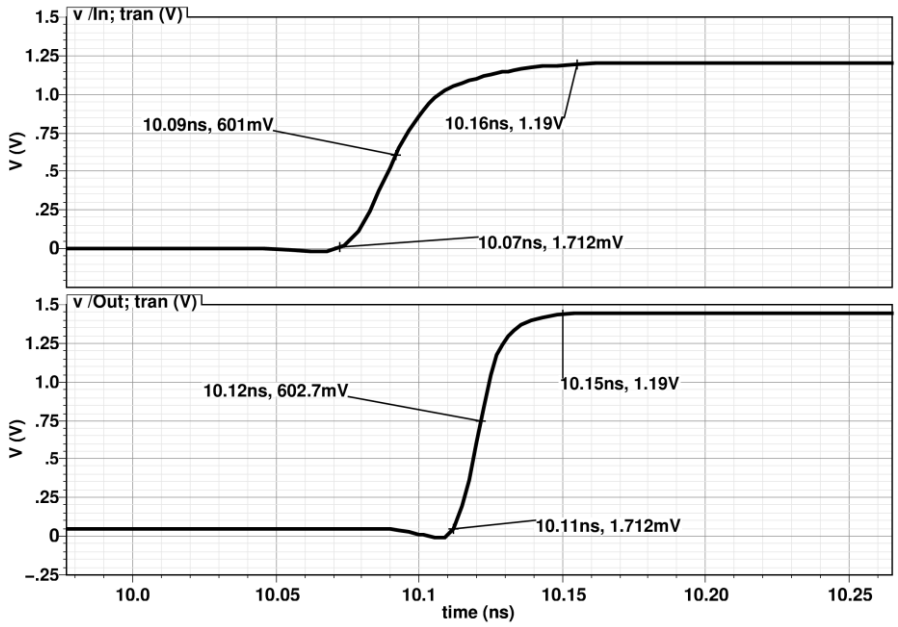


Figure 5.42: Difference between rising edges of loaded output (up) and free output (down) *TSPC* divide-by-2 circuits

Also decreasing the load capacitance decreased the power consumption. From Equation 5.19, we can estimate the power consumption at average frequency of *500 MHz* so that

$$P_{diss} = C_L V_{DD}^2 f = 4.2\ \mu W \qquad \text{Equation 5.39}$$

The maximum frequency at which the *TSPC* divide-by-2 circuit worked properly was more than *4 GHz.* Figure 5.43 shows the input and output waveforms of the *TSPC* divide-by-2 circuit at *4 GHz*.

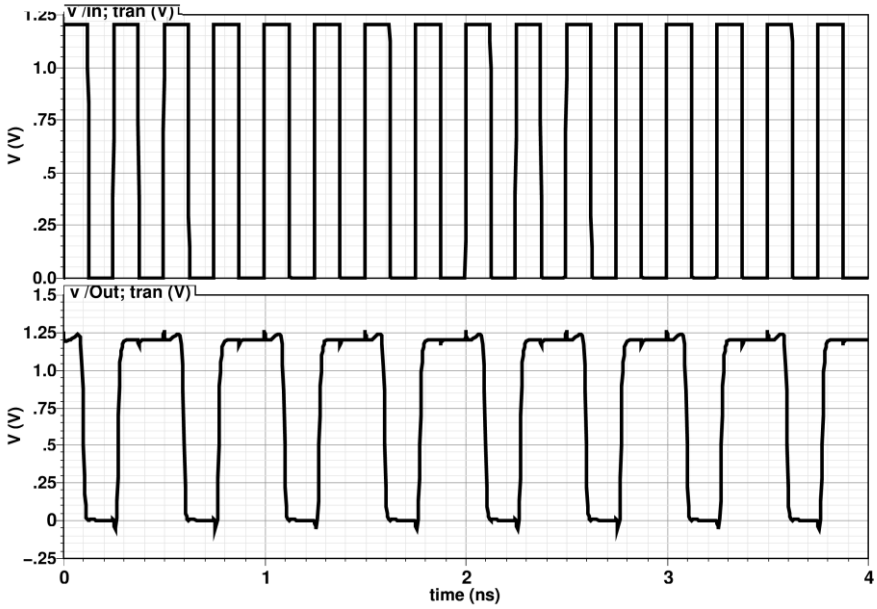


Figure 5.43: Input and output waveforms of *TSPC* divide-by-2 circuit at *4 GHz*

This analysis has cleared that it is better to use the ***TSPC* topology** especially because, as we saw in the frequency planning we just developed, it seems that we are not in need to use a very fast topology, because our maximum operating frequency is *840 MHz*, thus the only advantage of the *CML* is not very necessary here.

Now, in order to implement the PS-divider we described in Subsection 5.4.1, we first implemented the prescaler 2/1 by a divide-by-2 frequency divider followed by a *2*-to-*1 MUX*, where the *MUX* passes the original VCO signal when its select is 0 and passes the divided by 2 signal when its select is 0. This select is considered the prescaler modulus control.

The PS-counter design is implemented as shown in Figure 5.38 with *7* successive *D-FFs*; the most significant *4* of them are incorporating *OR* gates merged with their inputs to allow changing the value *P* as explained in Subsection 5.3.3 and according to the values tabulated in Table 5.11. The circuit has *11* inputs; the *7 Ds* are for determining *S*, and the 4 *Ps* are for determining the value *P*. The inputs D_0 through D_6 are entered in binary according to Table 5.11, while the inputs P_{16} P_{32}, P_{64} and P_{128} are entered according to Table 5.12.

Table 5.12: Possible values for maximum-count select inputs

Counter is	P	P_{16}	P_{32}	P_{64}	P_{128}
3-bit counter	8	1	1	1	1
4-bit counter	16	0	1	1	1
5-bit counter	32	0	0	1	1
6-bit counter	64	0	0	0	1
7-bit counter	128	0	0	0	0

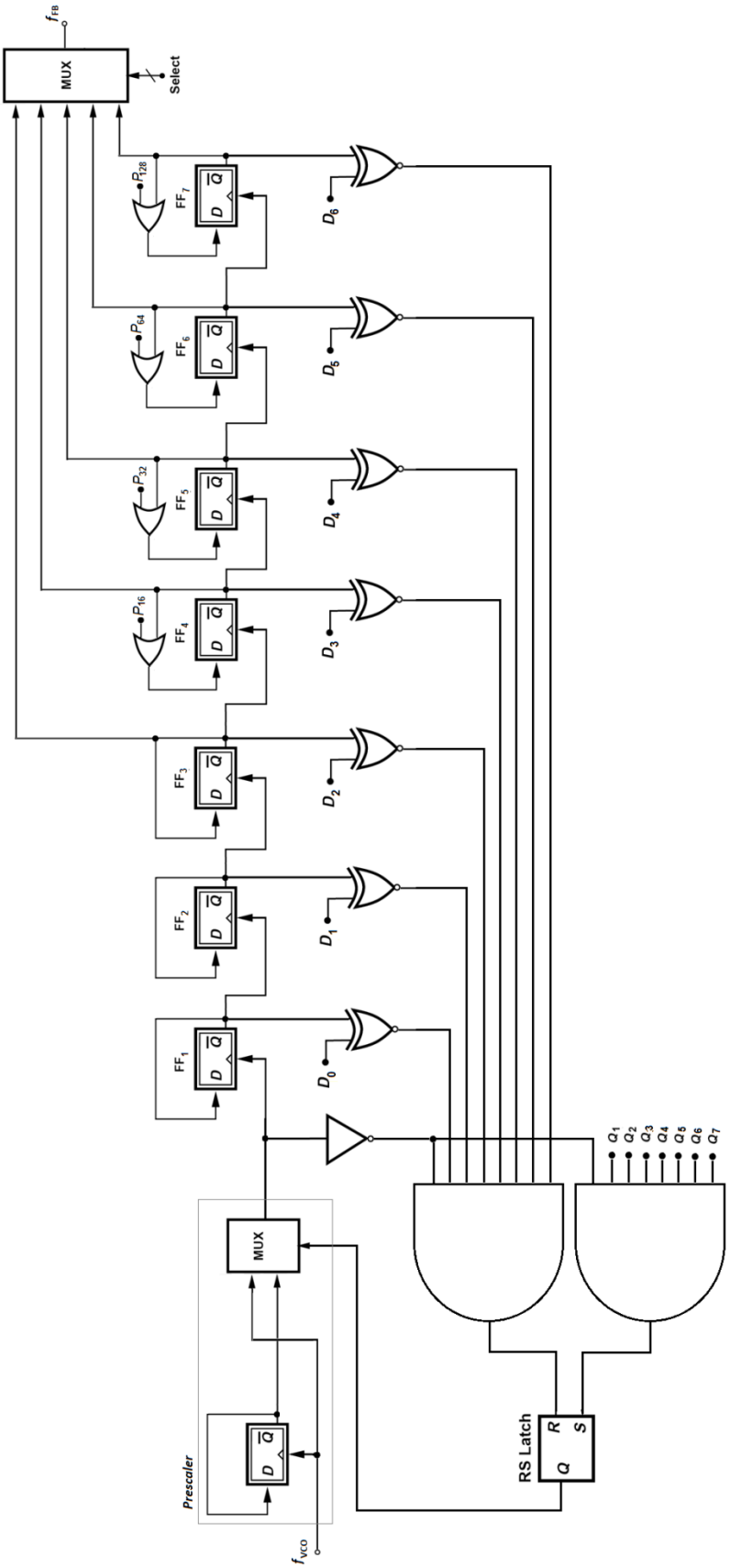


Figure 5.44: The full frequency divider design

Figure 5.45 shows the total delay of seven cascaded divide-by-2 circuits connected to nothing else. It was about 0.*2 ns*, which is approximately equal to *7* times the delay of a single divide-by-2 circuit shown above in Figure 5.42. In Figure 5.46, we present the edges of the input clock to the PS-counter we developed and its corresponding final output (from the *MUX*) when the seven *D-FFs* are enabled (*P = 128*). This shows a total delay between the input and the output of about *0.7 ns*. This increase in delay from that of the seven cascaded *FFs* is due to the effect of the load capacitance of the additional gates such as *XNORs* and the output *MUX*. However, this delay is still below one period of the maximum possible frequency (*1/840 MHz = 1.2 ns*).

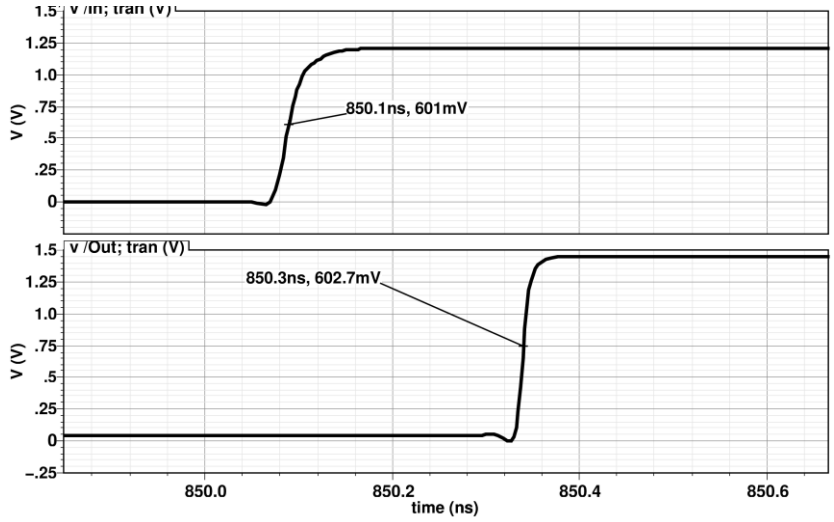


Figure 5.45: Front to end delay of cascaded divide-by-2 circuits

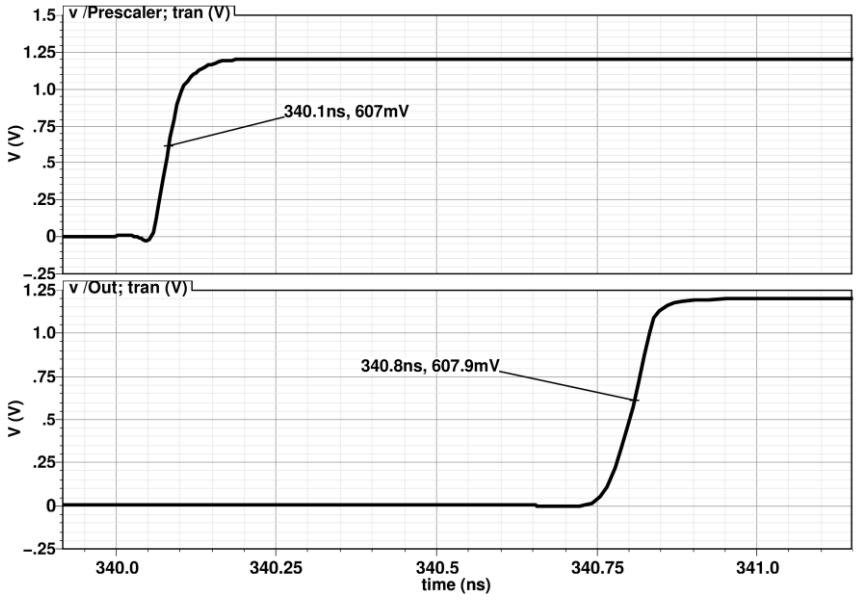


Figure 5.46: Front to end delay of the designed 7-*D-FF* PS-counter

Figure 5.47 shows the set, reset, MC, prescalaer output and the final PS-divider output waveforms at the maximum *VCO* frequency, *840 MHz*. These waveforms shows that when the set signal rises, MC changes to one, causing the prescaler to divide the input frequency by 2, until the reset rises, then the *MC* falls down causing the prescaler to divide by 1 (pass

the input frequency as it is). Also we can see that the output waveform makes a rising edge every about *167 ns* (*1/6 MHz*) which is the reference period. In fact, all the waveforms shown (set, reset, MC and prescaler output) are periodic with the same period *167 ns*. These results prove the correct operation of the divider at the maximum possible frequency which is the most critical case.

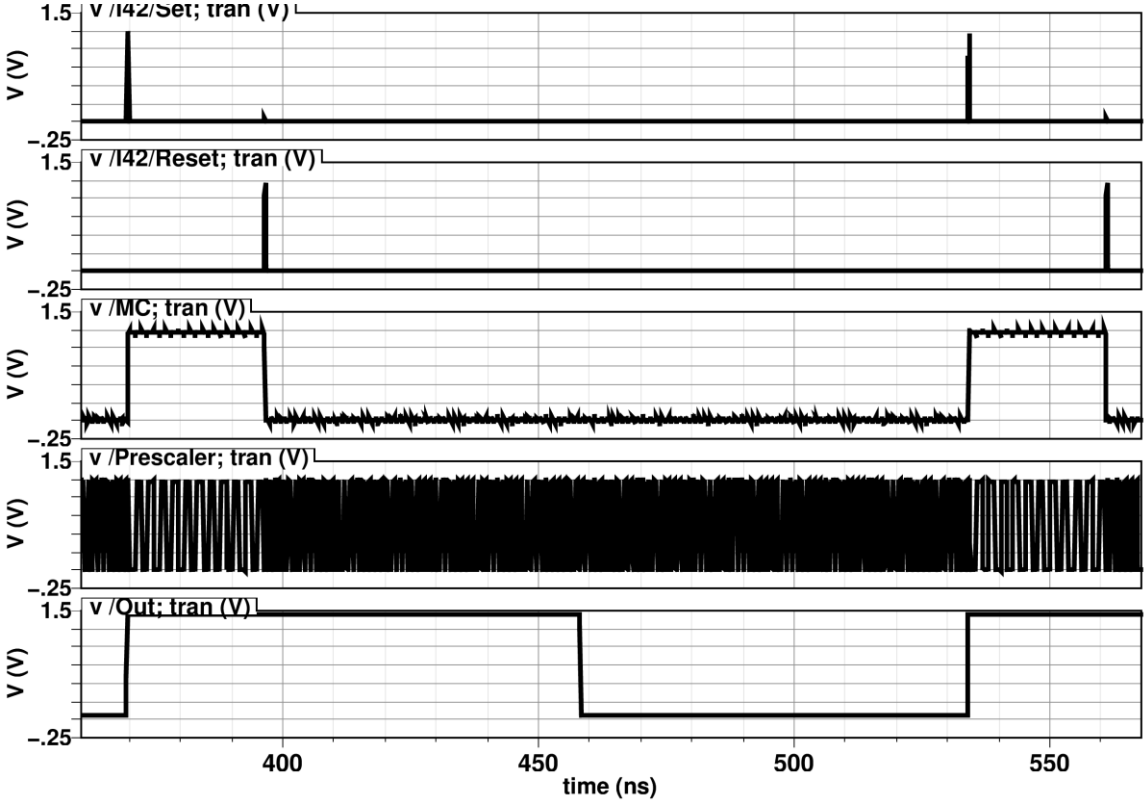


Figure 5.47: Waveforms of different signals

6 Fractional-*N PLLs*

In Fractional-N *PLLs*, it is required to divide by a fractional modulus because this highly relaxes the design of the frequency divider and allows increasing the reference frequency, which minimizes the effect of the phase noise of the *VCO*. Early attempts at fractional-N synthesis alternated between integer divide ratios in a repetitive manner, which resulted in noticeable spurs in the *VCO* output spectrum. More recently, other techniques have been used to generate a random sequence with the desired duty cycle to control the multi-modulus dividers. This causes the increase of the noise floor. Thus, a tradeoff arises here between spurs eliminating (modulus randomization) and noise reduction. However, ∑Δ-modulators can be designed so that most of the noise power in its output sequence is at frequencies that are above the loop bandwidth, and so are largely rejected by the loop.

In general, ∑Δ-modulators are used in *ADCs* and *DACs* in addition to *PLLs* in order to achieve high resolution in the analog domain using digital techniques. Its basic idea is to quantize an analog signal (as in *ADCs* and *DACs*) or a digital data with large number of bits (as in *PLLs*) into a digital data with small number of bits with high resolution. This can be obtained by a feedback error accumulating loop that accumulates the quantization error and compensates for it in the next clock cycle in a periodic manner.

In this chapter, we start by discussing the theory of fractional-N *PLLs* and their different aspects such as spurs elimination and noise shaping. Then we illustrate the implementation of first and second order digital ∑Δ-modulators and their basic components. Then we introduce some techniques for quantization noise reduction. We followed by defining dithering and proposing a new technique that eliminates spurs to a great extent. Finally, we proposed a fractional-N *PLL* frequency planning along with a convenient design for the frequency divider.

6.1 Basic Concept:

Fractional-N PLL solves the problem of Integer-N *PLLs* by releasing the frequency resolution from its dependency on the reference frequency. Thus, finer frequency resolution can be obtained without changing the reference frequency or the loop bandwidth, which allows relaxing the loop filter without leaking the reference frequency component to the VCO input.

This is realized by not limiting the divide ratio *M* to be an integer. Instead, some techniques may be used to allow generalizing the dividing ratio to any fraction. The main technique used for this task depends on changing the modulus of the frequency divider between two values *N* and *N+1*, so that the average dividing ratio *M* is some value between *N* and *N+1*. This average can be controlled by controlling the time ratio at which the modulus is *N* and that at which it is *N+1*.

Figure 6.1 shows an example for a simple fractional-N *PLL*. The divide-by-10 frequency divider at the bottom generates output with frequency *10 MHz/10 = 1 MHz*, and duty cycle *10%*, i.e., every 10 clock cycles of the reference frequency its output is low for 9 of them and high for one. This output is used as modulus control for the main frequency divider (divide-by-10/11), so that every *10* clock cycles of the reference frequency, the divider divides by *10* for *9* of them and divides by *11* during the tenth cycle. This results in a number of pulses at the *VCO* output that is equal to *10×9 + 1×11 = 101* pulses every *10* reference clock cycles. Therefore, the average output frequency is equal to *101 ÷ (10×1μ) = 10.1 MHz*. That is the average value of the modulus is equal to *10.1*. The waveforms at different nodes of this circuit are shown in Figure 6.2.

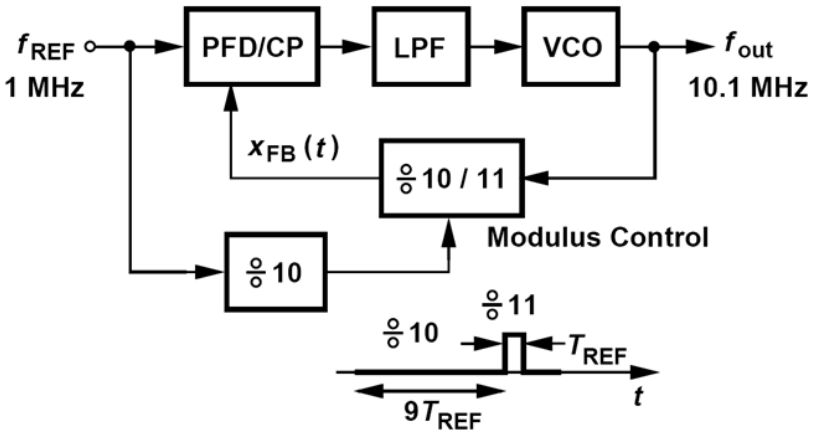


Figure 6.1: Fractional-N loop example

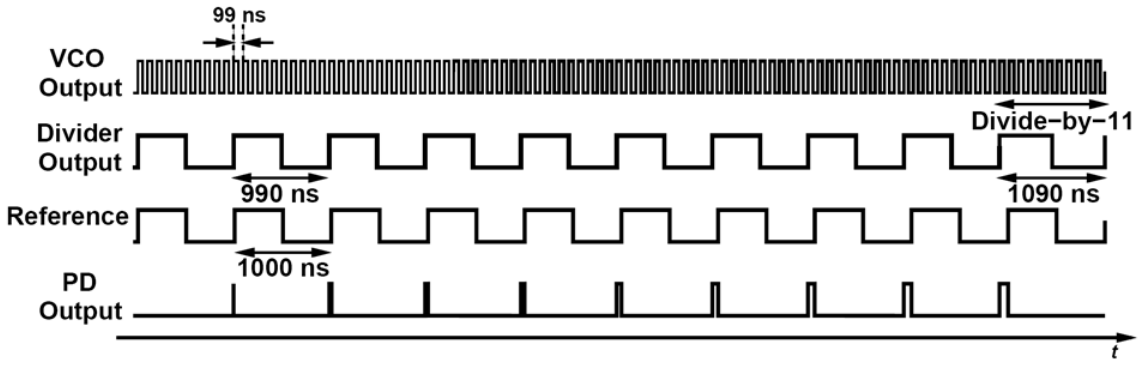


Figure 6.2: Waveforms at different nodes in the fractional-N loop example

In this example, we considered the average value of the modulus, which gives a feedback frequency of average *1 MHz*, and used this value to calculate the *VCO* output frequency, which is not the actual instantaneous value. When considering the actual instantaneous value for the modulus, we find that the feedback signal has fundamental frequency of *0.1 MHz* (*1/10* reference clock periods), which means that the feedback signal contains many sidebands at integer multiples of *0.1 MHz*. When these sidebands pass through the *PFD*, they yield components at harmonics of *0.1 MHz*, which results in modulating the *VCO* output and creating fractional spurs. The feedback signal then can be written as:

$$x_{FB} = A\cos(\omega_{Ref}\, t + \varphi(t)) \qquad \text{Equation 6.1}$$

where the phase error $\varphi(t)$ is periodically changing with time with frequency *0.1 MHz*.

6.1.1 Modulus Randomization (Spurs Elimination)

In order to suppress the output spurs of the *VCO*, we need to remove the *0.1 MHz* sidebands in the feedback signal. This can be achieved by breaking the periodicity of this signal. In other words, we need to randomize the shifting between the two dividing values of the dual-modulus frequency divider, with keeping the control on their average value which is the required dividing value. This approach causes the feedback signal to be as shown in Figure 6.3. This approach in turn results in increasing the total noise level of the loop, which will be solved later in this section.

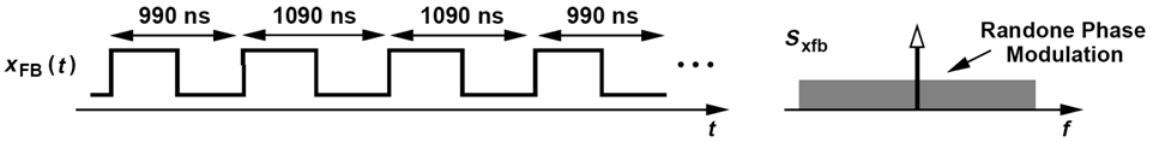


Figure 6.3: Randomized feedback signal of thefractional-N loop example

Now, the feedback signal expression in Equation 6.1 can be modified to:

$$x_{FB} = A\cos(\omega_{Ref}\, t + \varphi_n(t)) \qquad \text{Equation 6.2}$$

where the phase error $\varphi_n(t)$ is now changing randomly with time, so that it is equivalent to phase noise.

In order to calculate the value of this phase noise, suppose that we have two moduli; *N* and *N+1*, and we want to get an average modulus vale of $N+\alpha$, then we can consider the instantaneous modulus to be *N+b(t)*, where *b(t)* is a binary random variable that equals *0* or *1* at any instant with average α. We usually prefer using a zero mean random variable for simplification, so we define another random variable *q(t)* with zero mean such that:

$$b(t) = \alpha + q(t) \qquad \text{Equation 6.3}$$

Thus, the relation between *b(t)* and *q(t)* can be drawn like that in Figure 6.4. The value *q(t)* represents the quantization noise or the instantaneous error due to approximating α to *b(t)*.

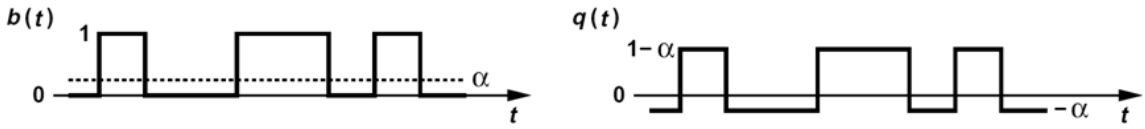


Figure 6.4: Waveforms of binary random variable *b(t)* (left) and quantization noise *q(t)* (right)

The feedback frequency can now be written as:

$$f_{FB}(t) = \frac{f_{out}}{N + b(t)} = \frac{f_{out}}{N + \alpha + q(t)} \qquad \text{Equation 6.4}$$

$$\therefore \quad f_{FB}(t) = \frac{f_{out}}{N+\alpha}\left(1 + \frac{q(t)}{N+\alpha}\right)^{-1} \qquad \text{Equation 6.5}$$

Assuming *q(t) << N+ α:*

$$f_{FB}(t) \approx \frac{f_{out}}{N+\alpha}\left(1-\frac{q(t)}{N+\alpha}\right) = \frac{f_{out}}{N+\alpha} - \frac{f_{out}}{(N+\alpha)^2}q(t) \quad \text{Equation 6.6}$$

Therefore the feedback signal can be written as:

$$V_{FB}(t) = V_0 cos\left(\int f_{FB}(t)dt\right) \quad \text{Equation 6.7}$$

$$\therefore \quad V_{FB} = V_0 cos\left(\frac{2\pi f_{out}}{N+\alpha}t - \frac{2\pi f_{out}}{(N+\alpha)^2}\int q(t)dt\right) \quad \text{Equation 6.8}$$

Thus, the phase noise at the output of the divider is given by:

$$\varphi_{n,div} = \frac{-2\pi f_{out}}{(N+\alpha)^2}\int q(t)dt \quad \text{Equation 6.9}$$

Therefore the effect of quantization noise on the phase can be plotted as in Figure 6.5. This is the phase noise at the output of the divider (feedback phase noise) we wanted to get. Let's now try to see how it affects the phase noise at the output of the *VCO*.

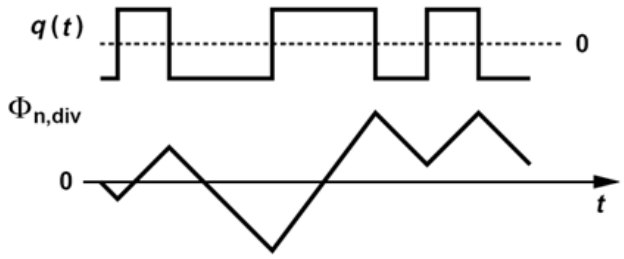


Figure 6.5: Effect of quantization noise on the frequency divider output phase

First, we get the spectrum of the feedback phase noise in frequency domain, so that we can analyze the phase noise with respect to the frequency. So, when applying Fourier Transform on Equation 6.9, we get:

$$\overline{\varphi_{n,div}}^2(f) = \frac{1}{(N+\alpha)^4}\left(\frac{f_{out}}{f}\right)^2 S_q(f) \quad \text{Equation 6.10}$$

It can be proven that the spectrum $S_q(f)$ of the quantization noise *q(t)* is given by:

$$S_q(f) = \frac{\alpha(1-\alpha)}{T_b}\left(\frac{\sin(\pi T_b f)}{\pi f}\right)^2 \quad \text{Equation 6.11}$$

where T_b is the width of the random square pulses. In other words, it is the period of the clock generating the random sequence *b(t)*. The spectrum of the quantization noise is shown in Figure 6.6.

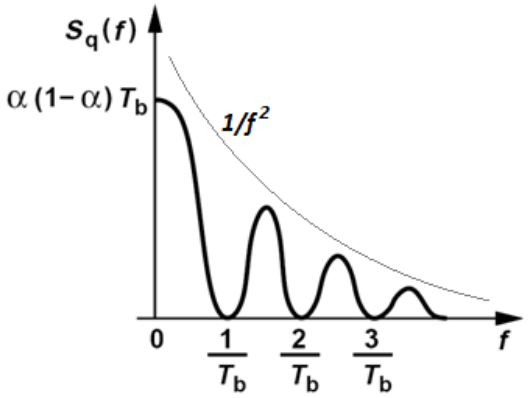


Figure 6.6: Power Spectrum of quantization noise *q(t)*

Now, in order to refer the phase noise at the feedback to a phase noise at the output of the *VCO*, we just multiply it by the dividing ratio *N+* α, so that the phase noise spectrum at the output of the VCO due the quantization noise is given by

$$\overline{\varphi_{n,out}}^2(f) = \frac{1}{(N+\alpha)^2}\left(\frac{f_{out}}{f}\right)^2 S_q(f) \qquad \text{Equation 6.12}$$

As we expected, the quantization noise indeed increased the noise level, while suppressing the spurs.

6.1.2 Noise Shaping

Since all what we care about is to decrease the noise level within the loop bandwidth only, we seek for a technique that generates the random pulses *b(t)* such that the phase noise due to quantization noise exhibits high pass spectrum as shown in Figure 6.7. This technique is so called "*Noise Shaping*".

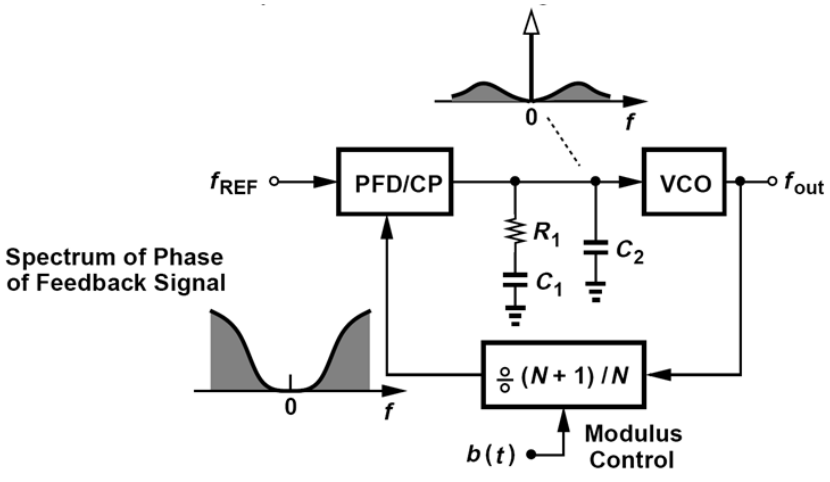


Figure 6.7: *PLL* incorporating modulus randomization and noise shaping

A negative feedback loop with an ideal integrator acts like a high pass system on the noise introduced near the output. For example, if we considered the simple negative feedback loop shown in Figure 6.8 , which has an input source *X(s)* and a noise source *Q(S)* near the output, then the transfer function with respect to the noise is:

$$\frac{Y(s)}{Q(s)} = \frac{s}{s+1} \qquad \text{Equation 6.13}$$

and that with respect to the input is:

$$\frac{Y(s)}{X(s)} = \frac{1}{s+1} \qquad \text{Equation 6.14}$$

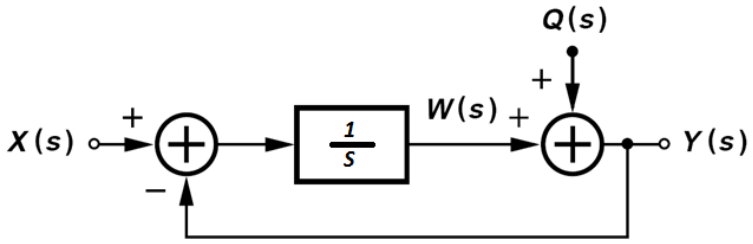


Figure 6.8: Negative feedback loop with noise shaped near output

Therefore, at high frequencies *Y(s)* is approximately equal to *Q(s)*, and at low frequencies it is approximately equal to *X(s)* and the noise effect is eliminated.

Now, we want to convert this loop into digital domain. The integrator is realized in digital domain as shown in Figure 6.9, where:

$$z = e^{-2\pi f T_{CK}} \qquad \text{Equation 6.15}$$

and T_{CK} is the sampling or clock period.

This implies a total loop block diagram as depicted in Figure 6.10. This negative feedback loop is so called ***∑Δ-modulator***.

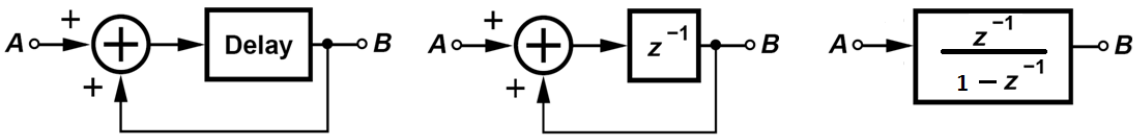


Figure 6.9: Digital integrator block diagram

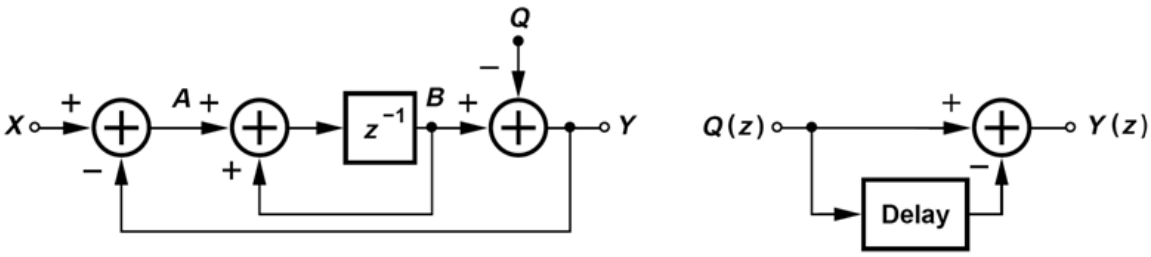


Figure 6.10: ∑Δ-modulator block diagram in digital domain

Then, the transfer functions in Equation 6.13 and Equation 6.14 can be written in the z-domain as following:

$$\frac{Y(z)}{Q(z)} = 1 - z^{-1} \qquad \text{Equation 6.16}$$

$$\frac{Y(z)}{X(z)} = z^{-1} \qquad \text{Equation 6.17}$$

That is, at zero noise, the output is the delayed version of the input by *1* clock period. While at zero input, the output is equal to the difference between the current noise and the delayed version of the noise, i.e., the noise undergoes a derivative effect. Therefore, for high (relative to clock frequency) frequency components of the noise, the noise difference is high, and hence it highly affects the output *Y(z)*. While for the low frequency components of the noise, the difference is almost zero. This is exactly the high pass response of the system with respect to the noise that we are seeking.

It is obvious that the reference for considering a frequency component high or low is the clock frequency, so as the clock frequency increases, more range of frequencies can be considered low and hence more noise frequency components are eliminated. So it is always better to increase the clock frequency. Figure 6.11 shows the effect of increasing the clock frequency on the difference between two successive clock samples of the noise.

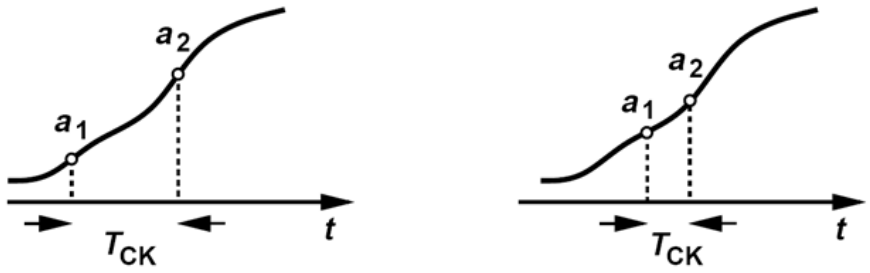


Figure 6.11: Two successive noise samples at low clock frequency (left) and high clock frequency (right)

Let's now study the spectrum $S_y(f)$ of the output *Y(z)*. From Equation 6.15, we can write the transfer function in Equation 6.16 as following:

$$\frac{Y(s)}{Q(s)} = e^{-j\pi f T_{CK}}\left(e^{j\pi f T_{CK}} - e^{-j\pi f T_{CK}}\right) \qquad \text{Equation 6.18}$$

$$\therefore \quad \frac{Y(s)}{Q(s)} = 2je^{-j\pi f T_{CK}}\sin(\pi f T_{CK}) \qquad \text{Equation 6.19}$$

Therefore the output spectrum is:

$$S_y(f) = S_q(f)|2\sin(\pi f T_{CK})|^2 \qquad \text{Equation 6.20}$$

$$\therefore \quad S_y(f) = 2S_q(f)|1 - \cos(\pi f T_{CK})| \qquad \text{Equation 6.21}$$

Thus, the squared magnitude of the transfer function of the noise shaping loop is drawn as in Figure 6.12.

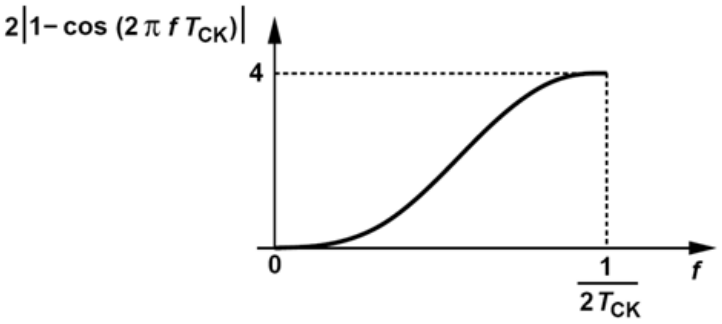


Figure 6.12: The squared magnitude of the noise shaping loop transfer function

Substituting for $S_q(f)$ in Equation 6.21 from Equation 6.11, we get:

$$S_y(f) = 2\frac{\alpha(1-\alpha)}{T_b}\left(\frac{\sin(\pi T_b f)}{\pi f}\right)^2 |1 - \cos(\pi f T_{CK})| \qquad \text{Equation 6.22}$$

Therefore, the spectrum curve for $S_y(f)$ at low frequencies can be deduced by multiplying the curves in Figure 6.6 and Figure 6.12 to be as shown in Figure 6.13.

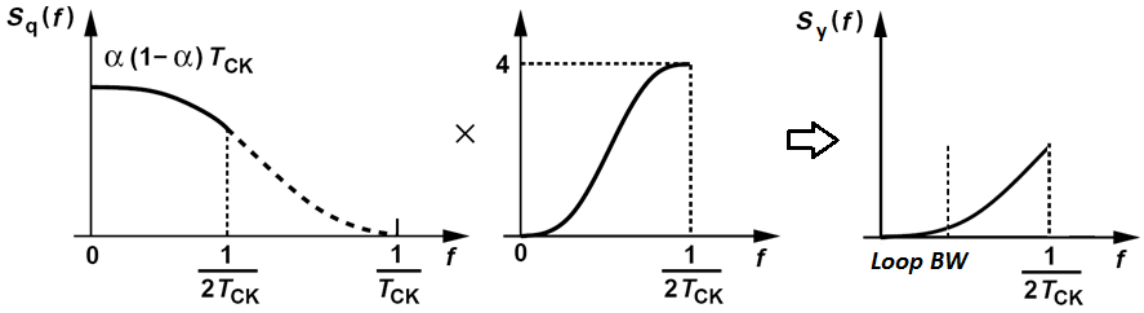


Figure 6.13: Output spectrum due to quantization at low frequencies after noise shaping

If the sampling clock frequency is the reference frequency, then we ensure that it is much more than the loop *BW*, typically about 10 times the loop *BW*. Therefore, as we can see from the product curve $S_y(f)$, the output noise spectrum due to the quantization noise is very small within the loop *BW*.

6.1.3 Higher Order Noise Shaping

The ∑Δ-modulator presented in Figure 6.10, whose noise transfer function is given by Equation 6.16, is called "*first order 1-bit ∑Δ-modulator*". That is because its transfer function is of the first order and its output '*y*' is only *1* bit. Other higher order ∑Δ-modulators can be designed to achieve higher in-band noise suppression. This can be easily predicted from Equation 6.20, where doubling the order of the loop, to be second order, results in doubling the power of the *sine* so that the output spectrum expression is modified to be:

$$S_y(f) = S_q(f)|2\sin(\pi f T_{CK})|^4 \qquad \text{Equation 6.23}$$

At low frequencies ($f << 1/T_{CK}$), this expression can be approximated to:

$$S_y(f) \approx S_q(f)|2\pi f T_{CK}|^4 \qquad \text{Equation 6.24}$$

This implies that at frequencies less than $1/(2\pi T_{CK})$, the noise shaping decreases in proportion to f^4, while in case of first order it decreases in proportion to f^2, i.e., at the second order the output would have a stronger noise suppression than that at the first order.

Now, we want to modify the loop in Figure 6.10, so that the noise transfer function becomes:

$$\frac{Y(z)}{Q(z)} = (1 - z^{-1})^2 \qquad \text{Equation 6.25}$$

This can be achieved by inserting another additional first order ∑Δ-modulator in the loop and converting the integrator of the first ∑Δ-modulator into a non-delaying integrator, as shown in Figure 6.14. This system is called *"1-bit second order ∑Δ-modulator"*.

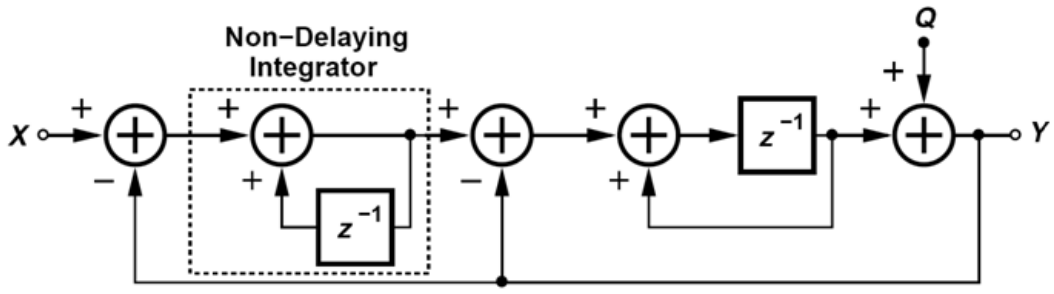


Figure 6.14: Second order digital ∑Δ-modulator block diagram

Figure 6.15 depicts the noise shaping functions of both the first and second order ∑Δ-modulators. It shows that at frequencies less than $1/(2\pi T_{CK})(\approx 1/6T_{CK})$, the second order exhibits a better noise suppression, while at frequencies more than $1/(2\pi T_{CK})$, the output noise of the second order is extremely increasing.

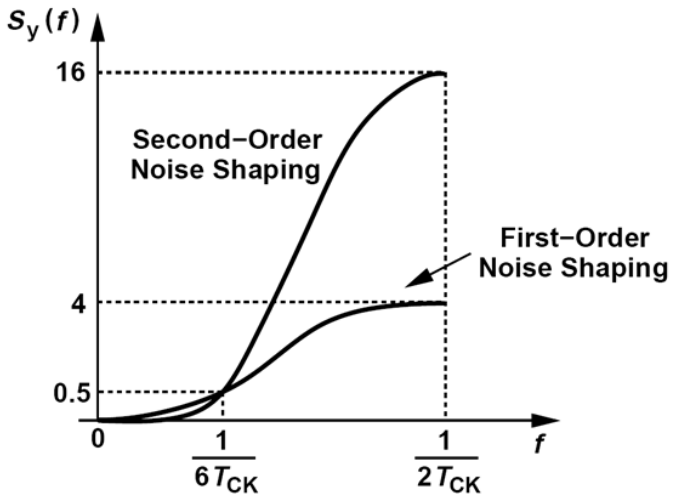


Figure 6.15: Noise shaping funtions for first and second order ∑Δ-modulators

Raising the order of the system increases the in-band noise suppression. However, feedback loops incorporating more than *2* integrators suffer from stability issues. Some techniques may be used for stabilizing these systems.

Another approach for higher order ∑Δ-modulators is called *"MASH ∑Δ-modulator"*. In this approach ∑Δ-modulators are added in a cascaded form with the quantization error of each stage calculated and given as the input to the next stage. Figure 6.16 shows the simplest possible MASH ∑Δ-modulator block diagram which is MASH 1-1 ∑Δ-modulator.

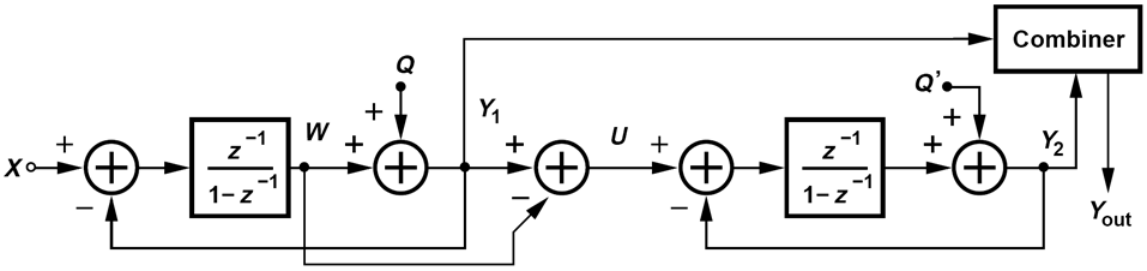


Figure 6.16: MASH 1-1 ∑Δ-modulator

In this system, the quantization noise *Q* is calculated by subtracting the output of the integrator before quantization (*W*) from the output after quantization (Y_1) so that:

$$U = Y_1 - W = Q \qquad \text{Equation 6.26}$$

Then this error is given to another ∑Δ-modulator to be quantized with fine resolution, so that the output Y_2 is an accurate representation for the quantization error *U*. Finally, the two outputs Y_1, representing the input *X*, and Y_2, representing the error in Y_1, are combined together to give Y_{out} that is a more accurate representation of the input *X*.

Form Figure 6.16, we can see write Y_1 and Y_2 in z-domain as following:

$$Y_1(z) = z^{-1}X(z) + (1 - z^{-1})Q(z) \qquad \text{Equation 6.27}$$
$$Y_2(z) = z^{-1}U(z) + (1 - z^{-1})Q'(z) \qquad \text{Equation 6.28}$$

$$\therefore \quad Y_2(z) = z^{-1}Q(z) + (1 - z^{-1})Q'(z) \qquad \text{Equation 6.29}$$

Since we want to combine Y_1 and Y_2 such that the quantization error Q is cancelled, then:

$$Y_{out}(z) = z^{-1}Y_1(z) - (1 - z^{-1})Y_2(z) \qquad \text{Equation 6.30}$$
$$Y_{out}(z) = z^{-2}X(z) - (1 - z^{-1})^2Q'(z) \qquad \text{Equation 6.31}$$

Thus, as we can see from Equation 6.31, the MASH 1-1 ∑Δ-modulator exhibits the same order noise shaping as the second order ∑Δ-modulator. The only difference between the mathematical results of the two topologies is that the output of the MASH 1-1 Y_{out} is delayed from the input X by 2 clock cycles, while the output of the 1-bit second order is delayed from the input by only *1* clock cycle. We also should denote that the output of the MASH 1-1 ∑Δ-modulator is not a single bit like those of the 1-bit first and second order ∑Δ-modulators. That is because the combiner in the MASH 1-1 combines two bits, Y_1 and Y_2, resulting in a 2-bit output. This part is discussed in detail in the following Section.

To sum up, we denote that higher orders of the MASH ∑Δ-modulators can be achieved without compromising the stability. Also the different stages of MASH can be of the second order ∑Δ-modulator.

6.2 ∑Δ-Modulator Implementation

In Fractional-N *PPLs*, ∑Δ-modulators are used to allow modulus randomization along with noise shaping by generating a random bit that is used as the modulus control for a dual-modulus *(N+1)/N* frequency divider. Thus, the divider divides by an average value between *N* and *N+1* according to the time ratio at which it divides by *N* and that it divides by *N+1*. Figure 6.17 shows a typical Fractional-N *PLL*. As shown in the Figure, the ∑Δ-modulator has an input *X*, which is an accurate binary representation of a fraction value α, so that

$$X = \alpha 2^m \qquad \text{Equation 6.32}$$

And

$$0 \leq \alpha \leq 1 \qquad \text{Equation 6.33}$$

where 2^m is the maximum valid value for *X*. The output of the ∑Δ-modulator is, in its simplest form, *1* bit *b(t)*, which is a binary sequence whose average is equal to the input *X*, and hence to the value α. At one extreme, when the required α is *1* (the required dividing ratio is *N+1*), the input *X* should be 2^m, and hence the output bit *b(t)* would be *1* constantly. At the other extreme, when the required α is *0* (the required dividing ratio is *N*), the input *X* should be *0*, and hence the output bit *b(t)* would be *0* constantly. In the middle between these two extremes α can be any fraction with resolution $1/2^m$. Therefore the dividing ratio has the same resolution and the output frequency has a resolution of $f_{Ref}/2^m$.

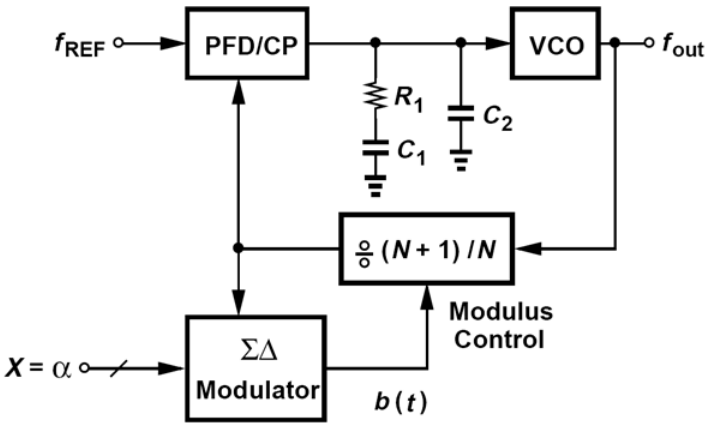


Figure 6.17: *Fractional-N PLL*

In this Section, we are going to discuss how to implement the block diagrams in the previous Section as digital circuit components.

First, we start with the first order ∑Δ-modulator. It consists of a feedback error calculator, an integrator and a quantizer that introduces the quantization noise. In digital circuits, the feedback error calculator is simply a subtractor that subtracts the feedback output *Y* from the input *X*. However, the input *X* consists of *m* bits, which represent the positive input of the subtractor, while the output *Y* is only *1* bit. If this single output bit is fed back to the m^{th} bit of the negative input of the subtractor, then the output of this error calculator is either *X*, when the output bit is *0*, or $-(2^m\text{-}X)$, when the output bit is *1*. The subtractor required must have *2* inputs each with *m+1* bits. The additional bit is the sign bit. Although the two inputs are always positive but their subtraction result may be negative. Even if the subtractor used has inputs with more than *m+1* bits, the output bit must be still fed back to the m^{th} bit in order to get the required average $\alpha = X/2^m$, and all the additional unused bits are considered also sign bit. In fact, the position to which the output bit is fed back in the subtractor is the factor determining the value *m*. Of course using a subtractor with input bits less than *m+1* cannot perform the task. The subtractor is implemented in digital circuits like an adder except that the negative input is inverted and the carry in is one. The implementation of the digital adder will be discussed in detail later in this section. We denote that the output of the subtractor is also of *m+1*, with the *MSB* is also the sign bit.

An integrator in digital circuit is simply implemented using an adder followed by a register clocked with the sampling clock, and the output of the register is fed back to be added to the input by the adder. The *m+1* output bits of the previous subtractor are the input of the integrator. Therefore the adder of the integrator has two (*m+1*)-bit inputs, and one (*m+1*)-bit output. Again the *MSB* here is the sign bit, and we denote that it is always *0*, i.e., the output of the integrator is always positive. This is because the loop starts initially with output *Y=0*, therefore it starts accumulating positive error ($X\text{-}0 > 0$) until it accumulates total error value more than or equal to 2^m. Then the output *Y* becomes *1*, so it accumulates negative error ($X\text{-}2^m < 0$) to its previous high positive error until the total error just becomes less than 2^m, at which the output *Y* returns *0* again and the loop returns to accumulate positive error, and so on. In other words, the output of the integrator is changes around 2^m, therefore it is always positive.

Since quantization is made to choose only one bit representing the output of the integrator, therefore it can be achieved by selecting the *MSB* of the output of the integrator, but as we just mentioned the *MSB* here is the sign bit and it is always *0*, therefore we take the m^{th} bit as the output and the feedback bit that is returned to the m^{th} bit too of the negative input of the subtractor as mentioned before.

This implies implementing the block diagram of the first order ∑Δ-modulator shown in Figure 6.10 as represented in Figure 6.18.

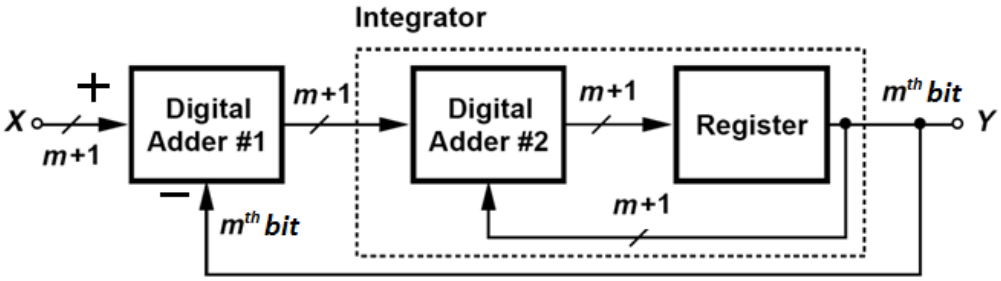


Figure 6.18: First order digital ∑Δ-modulators

Figure 6.19 shows the output of a 3-bit first order ∑Δ-modulator (*m=3*) at input *X = 2* ($\alpha = 2/2^3 = 1/4$) at clock frequency *50 MHz*. The waveform shows that the output is periodic with period $2^3/50\ MHz = 160\ ns$ and with duty cycle *1/4*. Therefore the average of the output over one period is equal to the required value ratio α. Figure 6.20 is another example for the same first order 3-bit ∑Δ-modulator, but at input *X = 5* ($\alpha = 5/2^3 = 5/8$). The waveform gives the same output period of *160 ns,* but different duty cycle with value *5/8*, which is also equivalent to the ratio α.

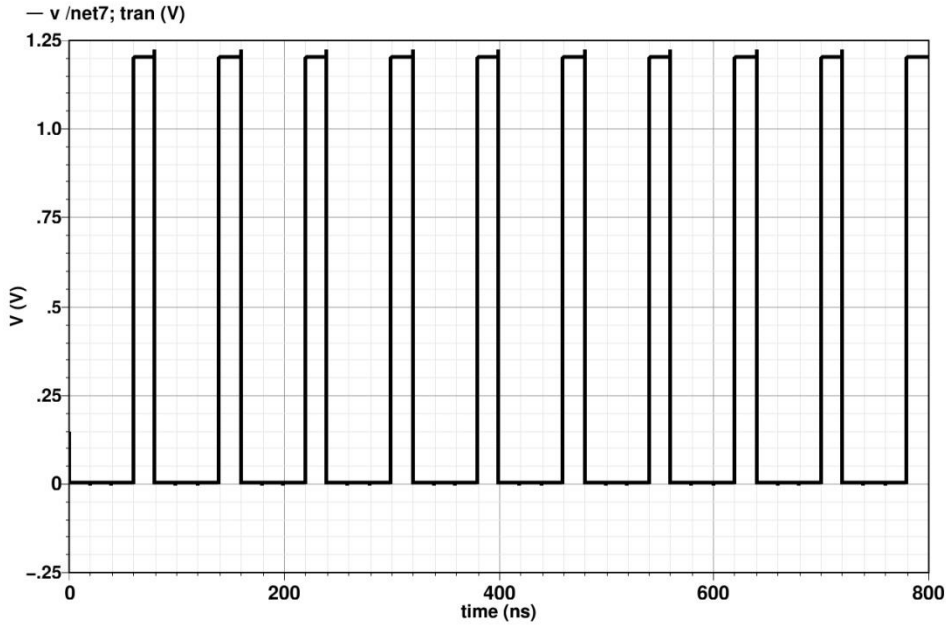


Figure 6.19: Output of a first order 3-bit ∑Δ-modulator at input *X = 2*

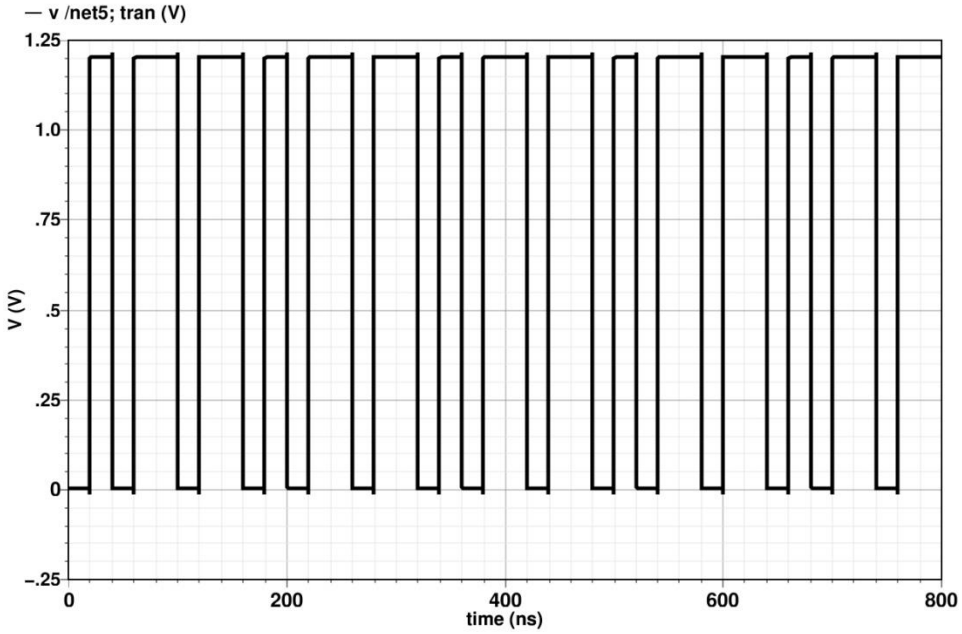


Figure 6.20: Output of a first order 3-bit ∑Δ-modulator at input *X = 5*

Let's now turn to the second order ∑Δ-modulator. As shown in Figure 6.21, we need the second ∑Δ-modulator's subtractors and adders to be with inputs of *m+2* bits. This is because the output of the non-delaying integrator may be more than 2^m, therefore when introduced to another ∑Δ-modulator, it is like an input of *m+1* bits, so it needs a ∑Δ-modulator of *m+2* bits, where the additional bit is the sign bit.

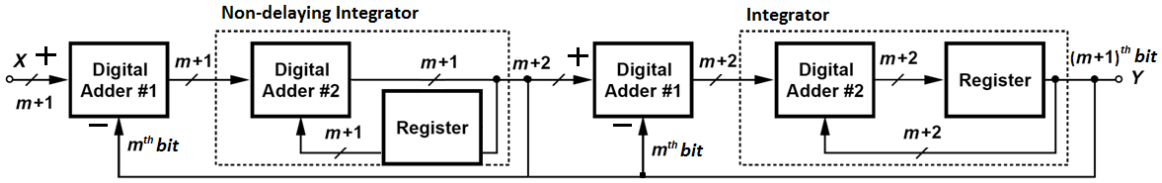


Figure 6.21: Second order digital ∑Δ-modulator

Figure 6.22 and Figure 6.23 show the output of a 3-bit ∑Δ-modulator at input *X = 2* and *X = 5,* respectively, at clock frequency *50 MHz*. The waveforms show that the output is periodic with period *320 ns* (double the period of the corresponding first order ∑Δ-modulator which means less spurs) and with duty cycles *1/4* and 5/8, respectively (same duty cycle as the corresponding first order ∑Δ-modulator which means same average value).

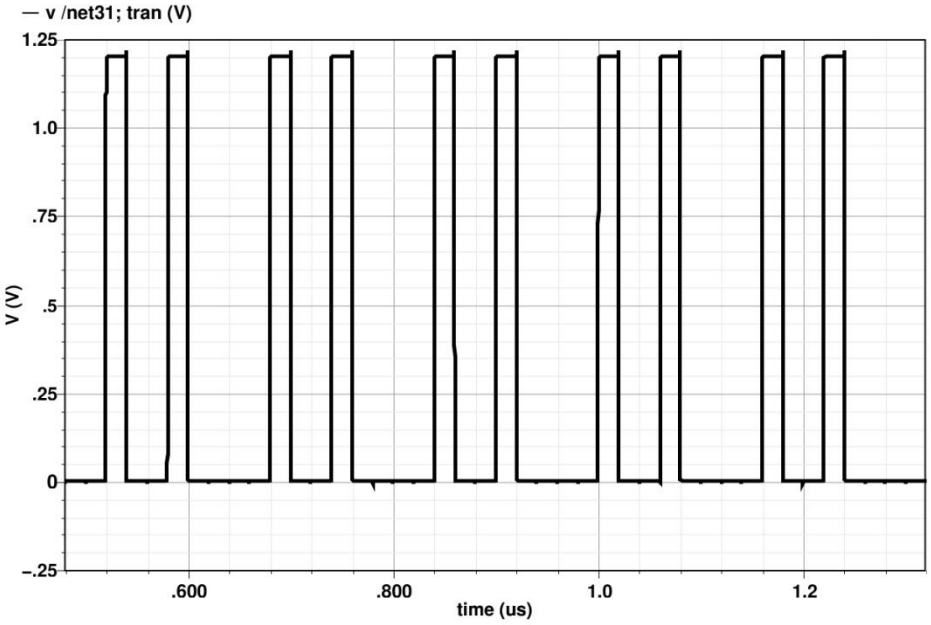


Figure 6.22 Output of a second order 3-bit ∑Δ-modulator at input *X = 2*

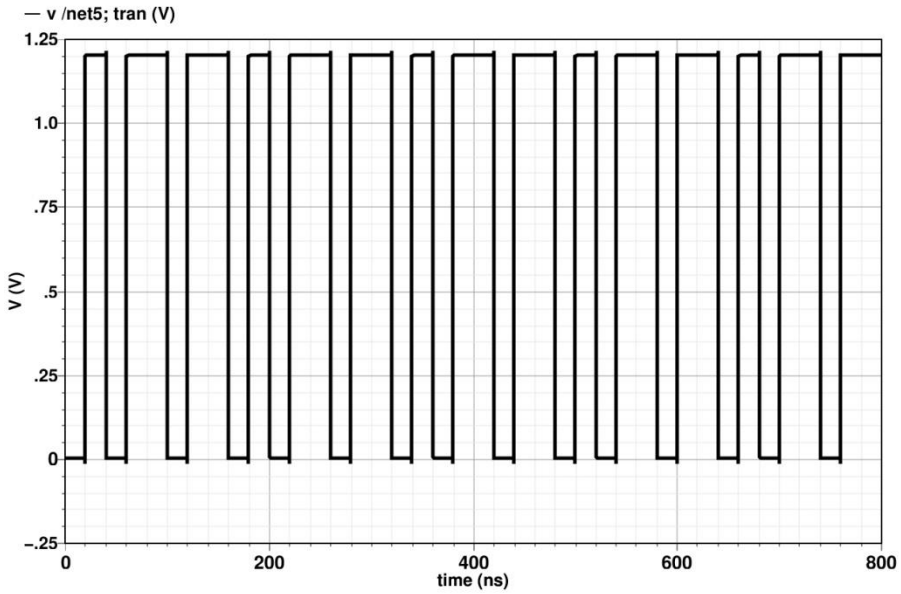


Figure 6.23: Output of a second order 3-bit ∑Δ-modulator at input *X = 5*

The other second order ∑Δ-modulator type is the MASH 1-1 ∑Δ-modulator. Its digital circuit implementation is as shown in Figure 6.24. The idea of this modulator is that it calculates the quantization noise introduced by the first ∑Δ-modulator then quantizes it into one bit using another ∑Δ-modulator again, then combines the two outputs Y_1 and Y_2 as the equations shown in the previous Section.

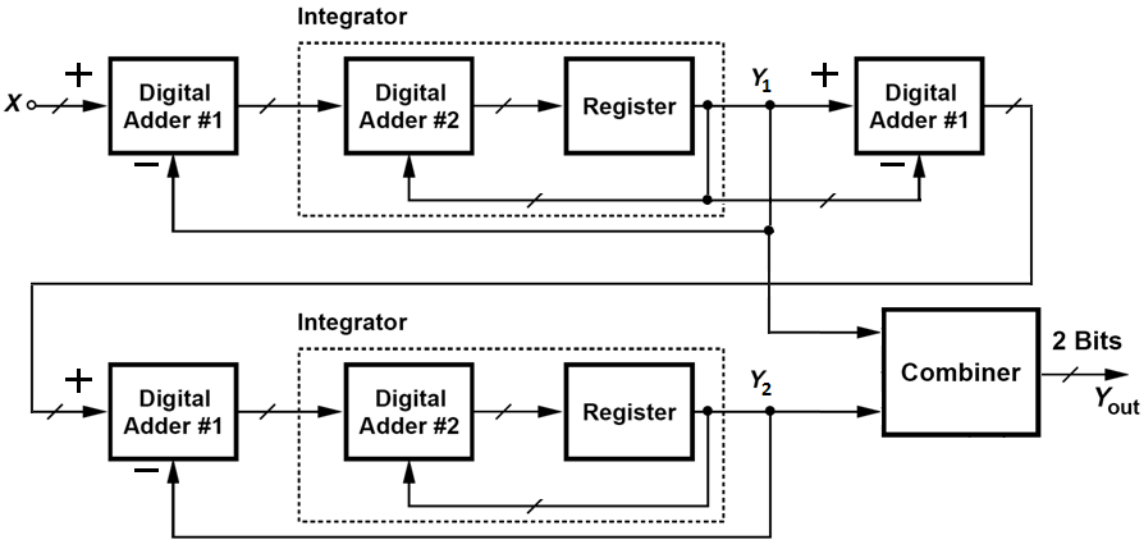


Figure 6.24: MASH 1-1 ∑Δ-modulator

If we used a MASH 1-1 or MASH 2-1 ∑Δ-modulators, according to the needed noise shaping, then the output has two bits, therefore it has four possible values, so we have to use a multi-modulus divider with *4* moduli divide-by-(N-1)/N/(N+1)/(N+2), so that the value of the output Y_{out} of the ∑Δ-modulator corresponds to the divider modulus according to the values in Table 6.1.

Table 6.1: Possible output of MASH 1-1 ∑Δ-modulator and its corresponding moduli

Y_{out}	Modulus
11	N-1
00	N
01	N+1
10	N+2

The combiner in this design can be easily implemented according to Equation 6.31, such that it adds the one clock cycle delayed versions of Y_1 and Y_2 and subtracts the current Y_2 from them. A circuit that does this task is shown in Figure 6.25.

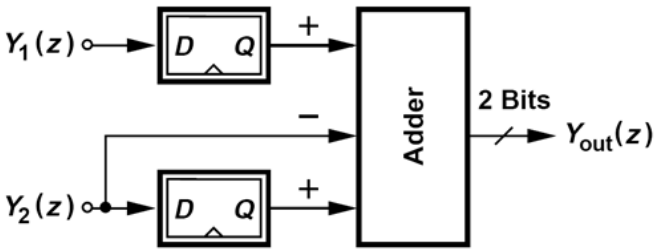


Figure 6.25: Combiner circuit

As we have discussed, ∑Δ-modulators circuits are implemented using only register (*DFFs*) and adders. We have already discussed the different aspects of the *FFs* and their circuits in

Chapter**Error! Reference source not found.**. A *TSPC DFF* proved a successful operation when used in the registers of the ∑Δ-modulators. Now, we want to implement a proper adder for the ∑Δ-modulator. The main challenge in the adder design is the propagation delay of the carry. This problem is critical in large number of bits ∑Δ-modulators, because this delay may exceed the period of the sampling clock, causing sampling over bits from different clock cycles, which yields wrong results. The traditional full adder suffers from this problem because, as shown in Figure 6.26, the output carry waits for the delay of two gates until its correct value is set, so when repeated with many other adder units in a cascaded form in a many-bit ∑Δ-modulator, it gives a severe delay that probably exceeds the clock period. This problem may impose a limitation on the maximum valid clock frequency, while as we explained in the previous Section, we aim to increase the clock frequency as much as possible in order to suppress noise better.

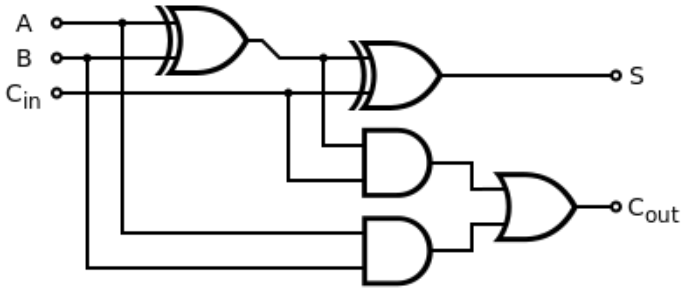


Figure 6.26: Traditional full adder circuit

Another efficient topology for the implementation of the full adder unit is the “Mirror Adder”, shown in Figure 6.27, this adder calculates the carry out very fast, before calculating the sum itself, and sends it to the next unit, so that the carry propagates through the train fast and relaxes the condition on the maximum allowable clock frequency.

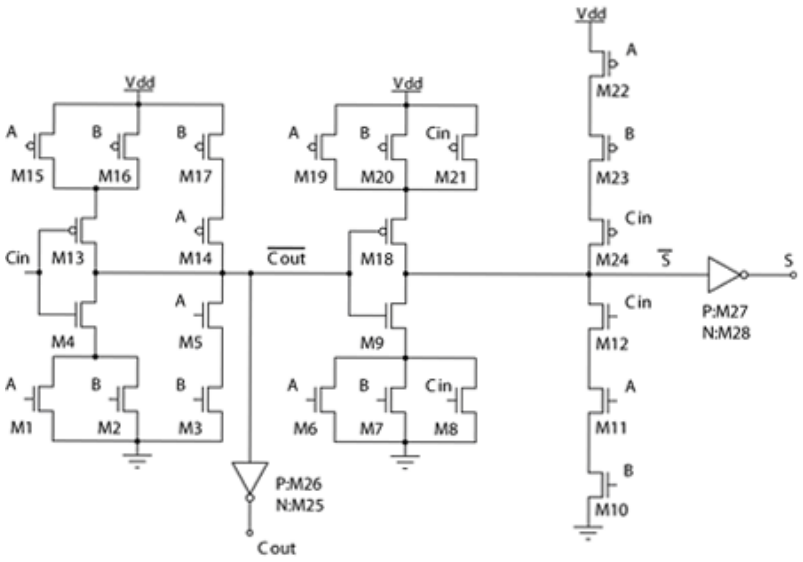


Figure 6.27: Mirror full adder circuit

6.3 Quantization Noise Reduction of ∑Δ-Modulator

In this Section we introduce a new technique that reduces the quantization noise of the ∑Δ-modulator. The idea combines between two well known techniques those depend on decreasing the resolution of the dual-modulus divider to be *N/N+ε*, where ε is a certain fraction.

The first is usually implemented if the *VCO* of the system is providing a multiphase output, where these phase shifted outputs are used to divide by moduli with difference less than *1*. That is, if a *VCO*, with input frequency ω_c, produces *M* output phases with a minimum spacing $2\pi/M$, and a *MUX* is then used to select one of these phases, then the output of the *MUX* is given by

$$V_{MUX}(t) = V_0 \cos\left(\omega_c t - k\frac{2\pi}{M}\right) \qquad \text{Equation 6.34}$$

where k is an integer from (*0* to *M-1*) referring to what phase the *MUX* is selecting at any time. If this variable *k* is changing linearly with time so that $k = \beta t$, then the output of the *MUX* can be written as

$$V_{MUX}(t) = V_0 \cos\left[\left(\omega_c - \beta\frac{2\pi}{M}\right)t\right] \qquad \text{Equation 6.35}$$

which gives a frequency of $(\omega_c - \beta\ 2\pi/M)$, i.e., this technique could divide the input frequency by $[1 - (\beta/\omega_c)(2\pi/M)]$. In order to obtain another dividing value we can repeat the circuit with different value of β. For example, assume a *VCO* that can produce four $\pi/2$-shifted phases. These four phases can be used to obtain dividing ratios with steps of *0.25* instead of *1*. Figure 6.28, shows a divide-by-1.25 circuit that uses the four $\pi/2$-shifted phases of the multiphase *VCO*. The waveforms show that, from time *0* to t_1 the *MUX* selects the output V_I, then from t_1 to t_2 the *it* selects V_Q, and so on, resulting in dividing ratio of *1.25*. Other dividing ratios of steps 0.25 can be obtained using different generating function for the select inputs of the *MUX*.

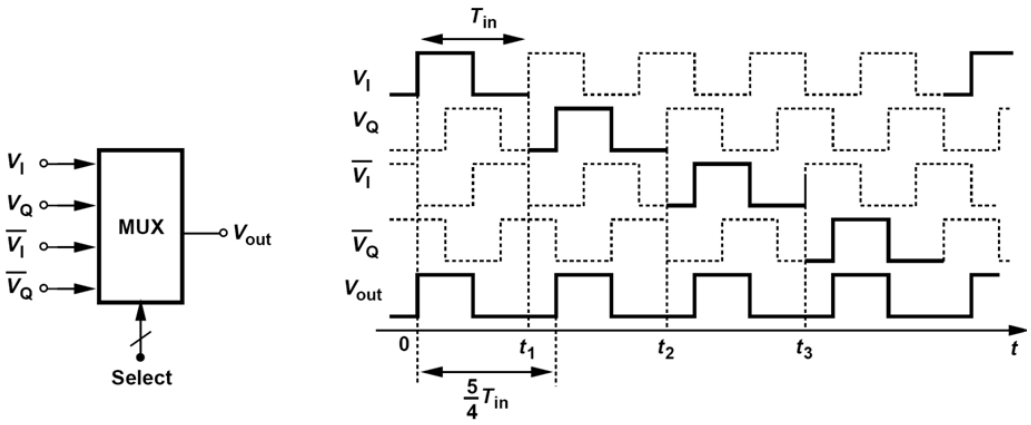


Figure 6.28: Divide-by-1.25 using outputs of multiphase *VCO* and *MUX*

Since the step difference is decreased to *0.25*, then the quantization noise falls by *20 log(4) = 12 dB*. The main challenge in this technique is the implementation of the circuit generating the select inputs of the *MUX*, especially when more than one modulus value is required; each modulus would require its own MUX select inputs generating circuit. This usually enlarges the size of the divider and complexes the design.

The other technique that our idea depends on is using *DET FFs* in the dividers. As we discussed in Section 5.2, *DET FFs* can be used in dividers to obtain half the modulus obtained by its corresponding *SET* divider. See the divide-by-1.5 circuit in Figure 5.24. If such dividers are used beside normal *DET* dividers, a modulus step of *0.5* can be realized, thus the quantization noise falls by *20 log(2) = 6 dB*.

Assuming a multiphase *VCO* with four $\pi/2$-shifted phases, our proposed idea is to divide the *VCO* output frequency by *1/2* (multiply them by *4*) by XORing the in-phase and quad-phase outputs of the *VCO*, as shown in Figure 6.28 . Then use the result as an input to a multi-modulus *DET* divider. We know that the *DET* divider gives a modulus step of *0.5*, and we have already multiplied *VCO* output frequency by *2*, thus we got a modulus step of *0.25*, which decreases the quantization noise by *20 log(4) = 12 dB*, avoiding the complexities of the *MUXs* and their select generating circuits.

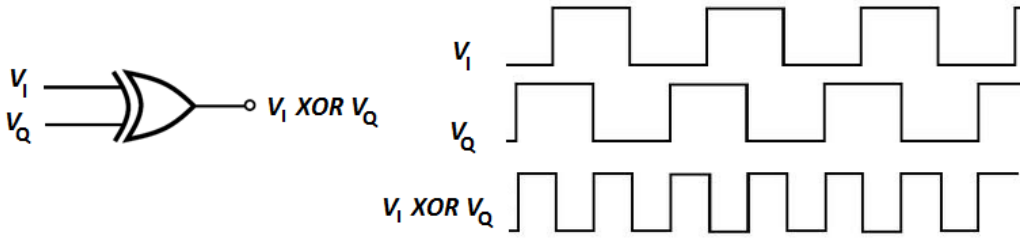


Figure 6.29: Dividng by *0.5* using quad-phase components and *XOR*

If we designed a *DET* divider such that the total circuit (the *DET* divider plus the *XOR* of the quad-phases) divides by *N*, *N+0.25*, *N+0.5*, *N+0.75* and *N+1*, then we can use it as the multi-modulus divider that is driven by the output of the ∑Δ-modulator. If we considered a ∑Δ-modulator with output one bit, then it can be used to shift the modulus between two moduli of step *0.25*, according to the required output fraction. Such type of divider can be implemented by a divider similar to that in Figure 5.29, but with *DET FFs*.

Other noise reduction techniques can be used such as *DAC* feed forward shown in Figure 6.30. In this technique the quantization error is converted into an analog value that is added as a current to the charge pump to compensate for the quantization noise introduced by the ∑Δ-modulator.

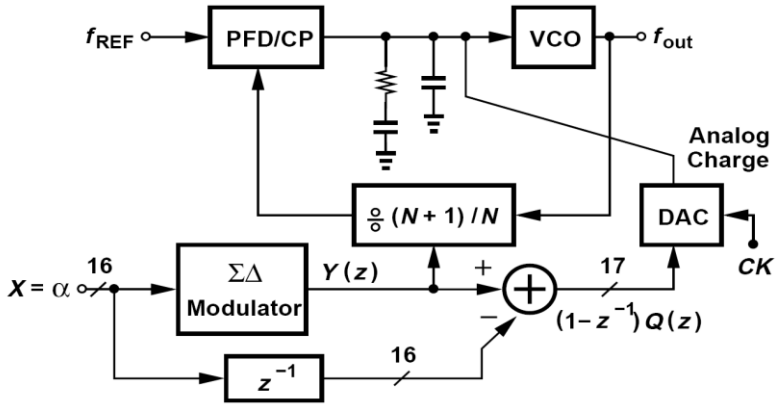


Figure 6.30: *DAC* feed forward in fractional-N *PLL*

Another method to decrease the quantization noise is to double the sampling clock frequency using a delay circuit with time delay *ΔT* and *XOR* gate, as shown in Figure 6.31.

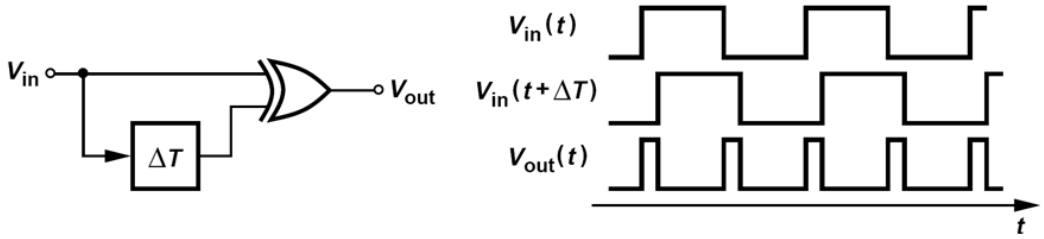


Figure 6.31: Doubling the clock frequency using a delay circuit

6.4 Dithering (Spurs Elimination) in ∑Δ-Modulator

In order to increase modulus randomization, and hence increase spurs suppression, dithering techniques are used in ∑Δ-modulators, where a randomly changing bit sequence is introduced somewhere in the loop in order to break its periodicity, such that the average value of the output is still equivalent to the input value. Typically, pseudo random sequences are generated using linear feedback shift register (*LFBSR*) for this purpose. A *LFSR* generates pseudo random output bit sequence with maximum period (2^k-1) T_{CK}, where k is the number of bits (*FFs*) of the register. In case of maximum period, the *LFSR* generates the number of *1s* in the 2^{k-1} output sequence is more than the number of *0s* by one. Figure 6.32 shows a 3-bit *LFSR*.

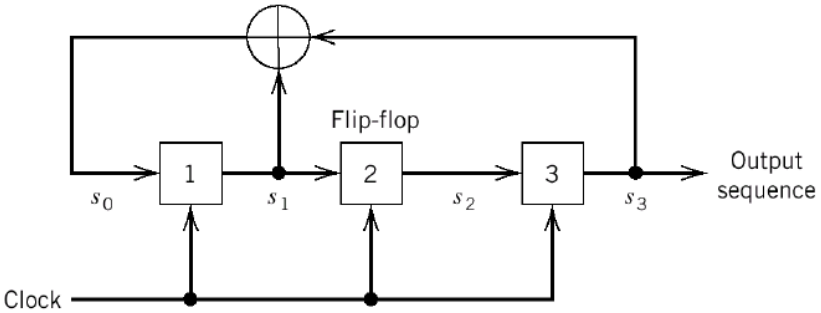


Figure 6.32: *LFSR* with *3* bits

One of the dithering techniques that depend on the pseudo random noise generation in ∑Δ-modulators is to toggle the *LSB* of the input *X* randomly between *0* and *1*. This lowers the sours in the output waveform but increases the noise floor, which can be solved by a higher order noise shaping.

Here, we introduce another technique for dithering. We suggest ANDing the output bit(s) of the ∑Δ-Modulator with the output sequence of a long *LFSR*, so that approximately half the sequence is *1s* and half it is *0s*. Then, the result of the ANDing process neglects half the output bits of the modulator itself and only the other half is passed normally. So, if we multiplied the input *X* by two, this would compensate for the neglected half and the other passed half would be perfectly randomized with keeping their average equivalence to the input *X*. This results in spreading the lower half of the possible inputs *X* (from *0* to 2^{m-1}) to cover all the available range for the input (from 0 to 2^m) and get rid of the upper half (from 2^{m-1} to 2^m) out of the range totally. This problem can be solved by subtracting 2^{m-1} from the inputs more than 2^{m-1} and pass the output through an *OR* gate, instead of an *AND* gate, with the output sequence of the *LFSR*.

6.5 Fractional-N *PLL* Frequency Planning and Divider Design

In this section we propose a design for a fractional divider for a TV-Tuner fractional-N frequency synthesizer according to ASTC standards.

6.5.1 Frequency Planning

We start by frequency planning for the system. First, we should exploit using the ∑Δ-modulator by increasing the reference frequency as much as possible, so we choose it to be equal to the minimum channel frequency, which is *48 MHz*. So, we have to divide all the other channel frequencies to be *48 MHz*.

To perform this task we first use a 2^N frequency divider with variable value *N*, as discussed in Chapter 1, that maps all the frequencies to the range between 48 *MHz* and *96 MHz*, and then use a multi-modulus frequency divider that is driven by a ∑Δ-modulator in order to divide them by a certain fractional ratio *1+α* between *1* and *2*.

Table 6.2: Frequency planning for TV-Tuner fractional-N *PLL*

N	2^N	1+ α			α			f_{VCO} (MHz) = 2^N(1+ α)		
		From	Step	To	From	Step	To	From	Step	To
0	1	1	1/8	15/8	0	1/8	7/8	48	6	90
1	2	1	1/16	31/16	0	1/16	15/16	96	6	186
2	4	1	1/32	63/32	0	1/32	31/32	192	6	378
3	8	1	1/64	127/64	0	1/64	128	384	6	762
4	16	1	1/128	140/128	0	1/128	140	768	6	840

6.5.2 ∑Δ-Modulator and Divider Design

In order to determine the required number of bits for the∑Δ-modulator, we assumed a frequency resolution of *1 ppm*, therefore $1/2^m = 10^{-6}$, therefore *m = 20* bits. Thus, for this resolution we need a 10-bit ∑Δ-modulator.

According to the noise shaping requirement, we may use a 2-bit second order, a MASH 1-1 or a MASH 2-1 ∑Δ-modulator. Let's consider the case of a MASH 1-1 or a MASH 2-1, so we need a divider with *4* possible moduli as illustrated in Table 6.1. According to our frequency planning, where we need *N* to be *1* so that the dividing ratio is between *1* and *2,* this would result in a problem when trying to get the dividing value *N-1* at *N=1*, because this would give a zero dividing value!!

Instead, we will use the technique we proposed is Section 6.3, where the required dividing values convert from (*N-1*), *N*, (*N+1*) and (*N+2*) into (*N-0.25*), *N*, (*N+0.25*) and *(N+0.5)*. However these values will only give average moduli from *N* to (*N+0.25*). So, in order to get average moduli from *N+0.25* and *N+0.5* we have to shift these *4* moduli up by *0.25*, so that they are *N*, (*N+0.25)*, (*N+0.5*) and *(N+0.75)*, and to get moduli from (*N+0.5)* to (*N+0.75)* shift

the up by another *0.25* and so on. This requires a divider that can divide by moduli from *N-0.25* up to *N+1.25* with step *0.25*.

Since, in our case, *N=1*, therefore we need a divider that deivides by moduli from *0.75* up to *2.25*, with step *0.25*. The circuit shown in Figure 6.33, along with double the frequency obtained from XORing the quad-phases achieve the required task.

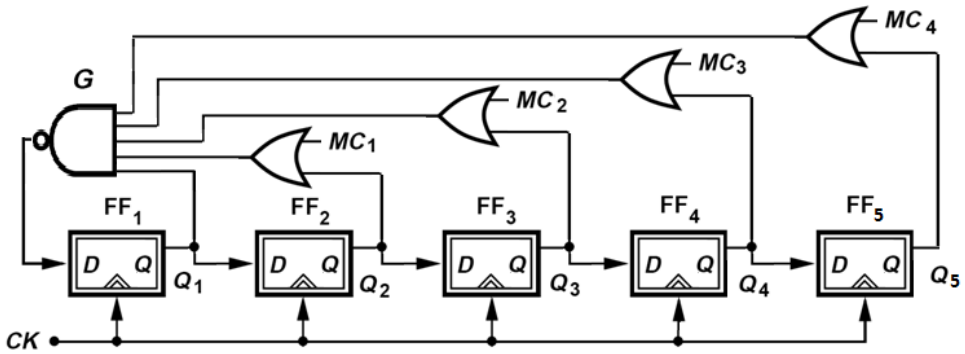


Figure 6.33: Divide-by-1.5/2/2.5/.../5 using *DET FFs*

If the clock in this circuit is double the frequency, then it operates according to the values in Table 6.3.

Table 6.3: Divding values and their corresponding modulus controls for the proposed design

Modulus	MC_1	MC_2	MC_3	MC_4
0.75	1	1	1	1
1	0	1	1	1
1.25	0	0	1	1
1.5	0	0	0	1
1.75	0	0	0	0
2	1	0	0	0
2.25	1	1	0	0

Figure 6.34 and Figure 6.35 show the input and output waveforms of the divider at modului *0.75* and *2.25*, respectively.

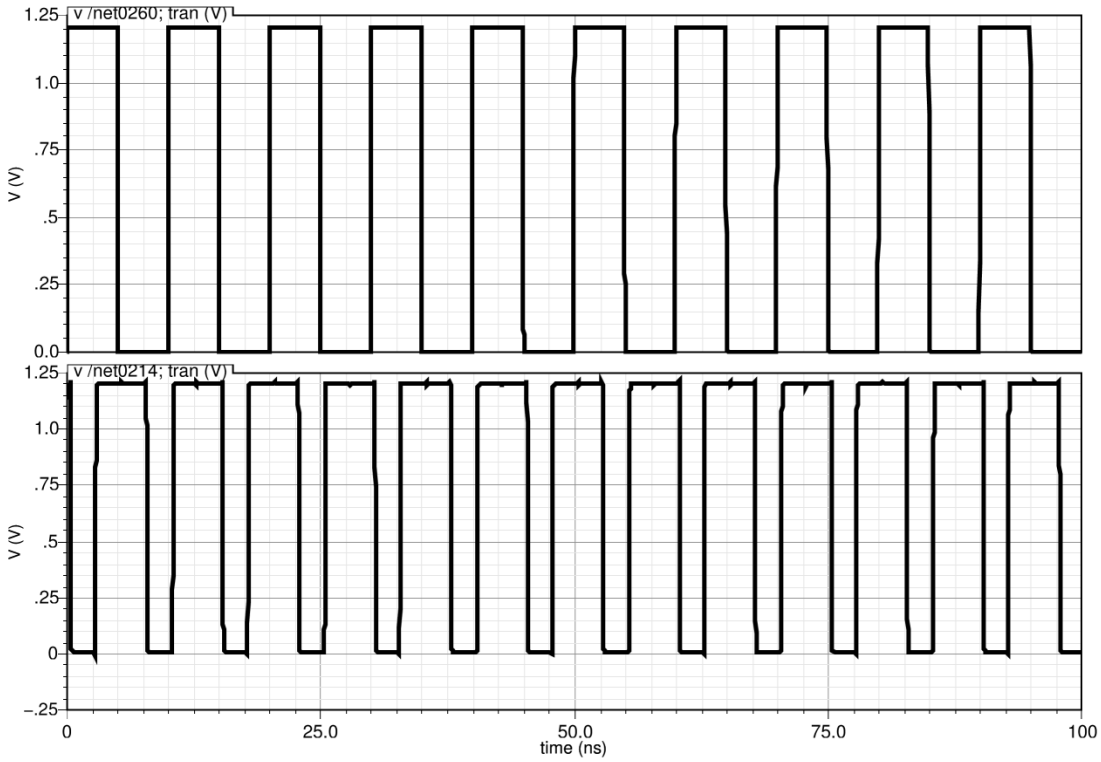


Figure 6.34: Output of the divider at modulus *0.75*

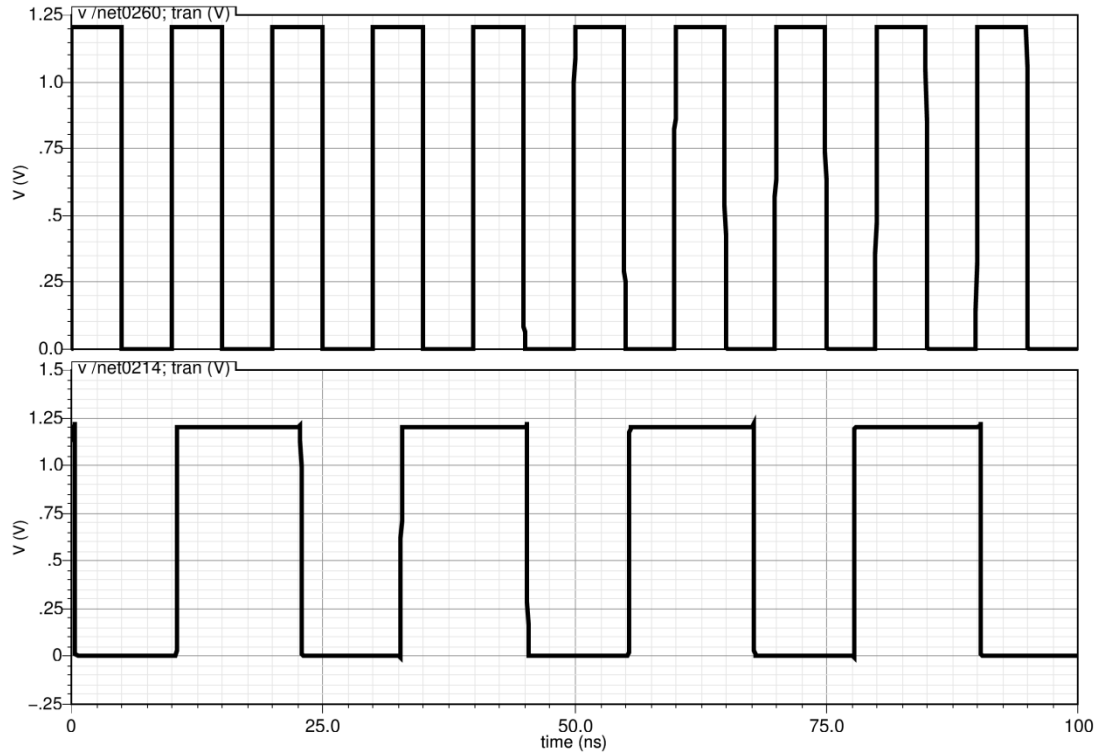


Figure 6.35: Output of the divider at modulus 2.25

Part B-2: Variable Gain Amplifier:

7 VGA:

7.1 Introduction

Variable gain amplifiers (VGAs) have a wide range of applications in electronic systems, especially in those requiring an automatic gain control loop. In communication systems, they play an indispensable role in receivers by controlling the incoming signal's power level and normalizing the average amplitude of the signal to a reference value. This helps in optimizing system capabilities and reducing the complexity of circuits designed to extract the correct timing information and data at the receiving end.

Figure 7.1 shows the block diagram of a UWB transceiver. The VGA takes its position between the notch filter and the analog to digital converter in the receiver portion of the transceiver.

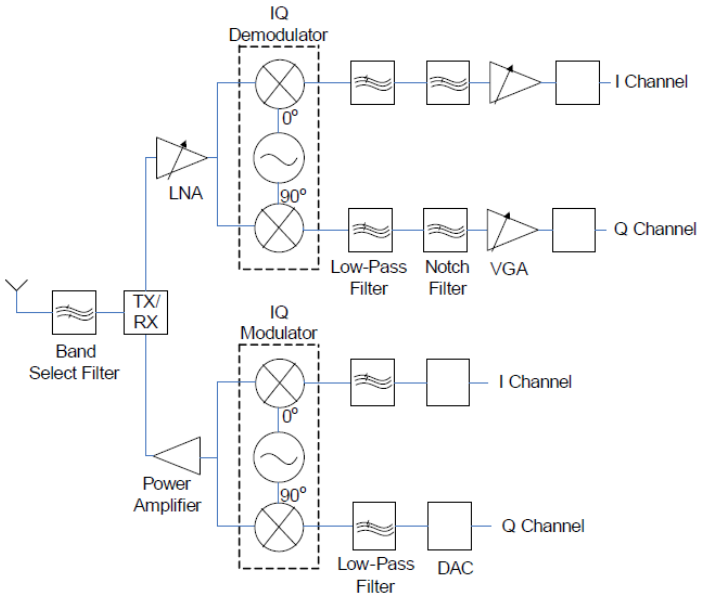


Figure 7.1 Block diagram of a UWB transceiver

The variable gain amplifier takes its position in the baseband section after the filters and before the analog-to-digital converter, in order to provide appropriate gain settings depending on the required amplification of the incoming signal. The gain of the VGA is controlled by the DSP section.

If the VGA gain varies exponentially with respect to the control voltage, the settling time of the AGC loop is constant and independent of the absolute gain. This improves the efficiency of the AGC loop and enables it to operate over a wide dynamic range. The linearity and the dynamic range of the VGA affect the overall third order input intercept point (IIP3) and noise figure of the receiver.

Hence one of the main requirements of the VGA is to provide good linearity for a wide range of signal swing. Other important aspects of the VGA design include bandwidth, group delay, DC offset cancellation, power supply and common mode rejection ratios, temperature and supply independence and power consumption.

For optimum performance, the bandwidth, group delay response and the output noise should remain relatively constant over the entire gain range of the VGA. Dynamic DC offset compensation has to be incorporated in the design since a small DC offset can be amplified by the VGA to a level that saturates the following stages or may cause the output signal to be clipped.

To achieve a gain that varies dB-linearly with respect to the control voltage, a VGA that implements the following pseudo exponential function is proposed with a gain given in Equation 7-1.

$$gain = \left(\frac{1+x}{1-x}\right)^n \qquad \text{Equation 7-1}$$

However, the region over which the polynomial approximates the logarithmic function is limited. To overcome this limitation, Equation 7-2 uses a Taylor series approximation of the log function.

$$e^x = 1 + x + \frac{x^2}{2} \qquad \text{Equation 7-2}$$

This method increases the range over which the gain is dB-linear with respect to the control voltage, but at the cost of reduced bandwidth and increased design complexity.

A disadvantage of this solution is the linear-in-magnitude gain control, which increases the settling time of the AGC loop in which the VGA is used.

Various issues that have been addressed in the paper are linearity, noise performance, power consumption, DC offset cancellation and minimization of temperature, supply and process variations.

The fundamental concepts involved in the design of the VGA and various techniques implemented in the circuit to improve its performance are described and analyzed in detail in the next section.

7.2 Fundamentals of VGA Design

The design of an amplifier requires a detailed analysis of the trade-offs involved in meeting the specifications. For instance, the higher the gain of the amplifier, the lower its bandwidth and the higher its non-linearity. Hence, selection of a particular topology is based upon the feasibility of the design meeting most of the specifications, as well as a careful consideration of the compromises that need to be made for certain parameters while ensuring that the system still works as expected.

7.2.1 Gain and Bandwidth Specifications

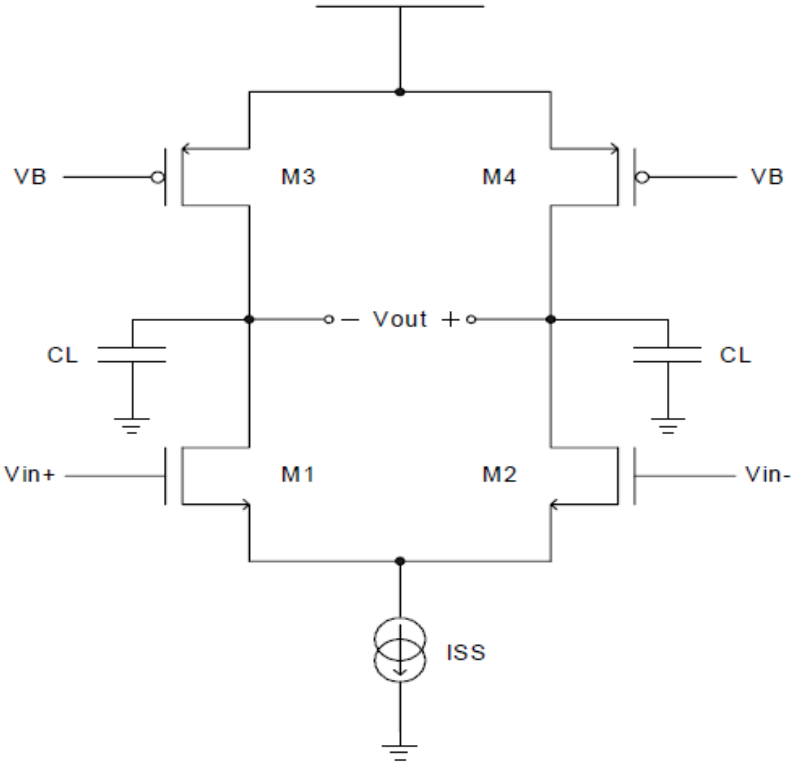


Figure 7.2 Differential amplifier with active load

A simple differential amplifier is given in Figure 7.2. The gain of this amplifier is given in Equation 7-3.

$$A_v = g_m(r_{o2} \| r_{o4}) \qquad \text{Equation 7-3}$$

where g_m is the transconductance of the input transistors M1 and M2. The -3dB bandwidth of the amplifier is given in Equation 7-4.

$$\omega_{-3dB} = \frac{1}{C_L(r_{o2} \| r_{o4})} \qquad \text{Equation 7-4}$$

Equation 7-3 and Equation 7-4 indicate that the gain of the amplifier is directly proportional and the -3 dB bandwidth is inversely proportional to its output resistance, leading to a tradeoff between the maximum gain that can be achieved and the speed of the amplifier.

Two stage amplifiers could be implemented to obtain higher gain, at the cost of additional poles and increased power consumption. So now we are going to discuss various techniques to obtain higher gain and their relationship with other specifications.

7.2.1.1 Cascoding

By compromising the output voltage swing, the same gain as a two stage amplifier could be obtained by using a cascode structure with lower power dissipation. The gain of the cascode stage shown in Figure 7.3 is given in Equation 7-5.

$$A_v = g_{m1}r_{o1}[(g_{m2} + g_{mb2})r_{o2} + 1] \cong g_{m1}g_{m2}r_{o1}r_{o2} \qquad \text{Equation 7-5}$$

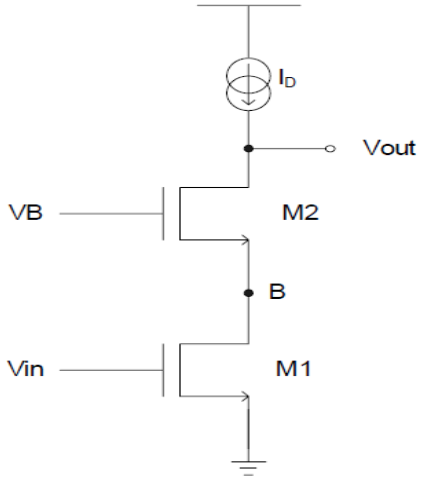


Figure 7.3 Cascode amplifier

An advantage of the cascode structure over a common-source stage is the significant reduction in the Miller effect observed by the gate-drain capacitor C_{GD1} due to the low impedance seen by the capacitor, looking into node B, for small values of R_D. The pole associated with the capacitors at node B is given approximately by Equation 7-6.

$$\omega_{-3dB} = \frac{g_{m2} + g_{mb2}}{2C_{GD1} + C_{DB1} + C_{SB2} + C_{GS2}} \qquad \text{Equation 7-6}$$

This normally results in a better frequency response of the cascode structure as compared to a simple common-source amplifier. A disadvantage of the cascade structure is its limited output voltage swing, as a result of which it is not used frequently in low voltage applications. Higher voltage swing can be obtained by using a folded cascade structure as shown in Figure 7.4.

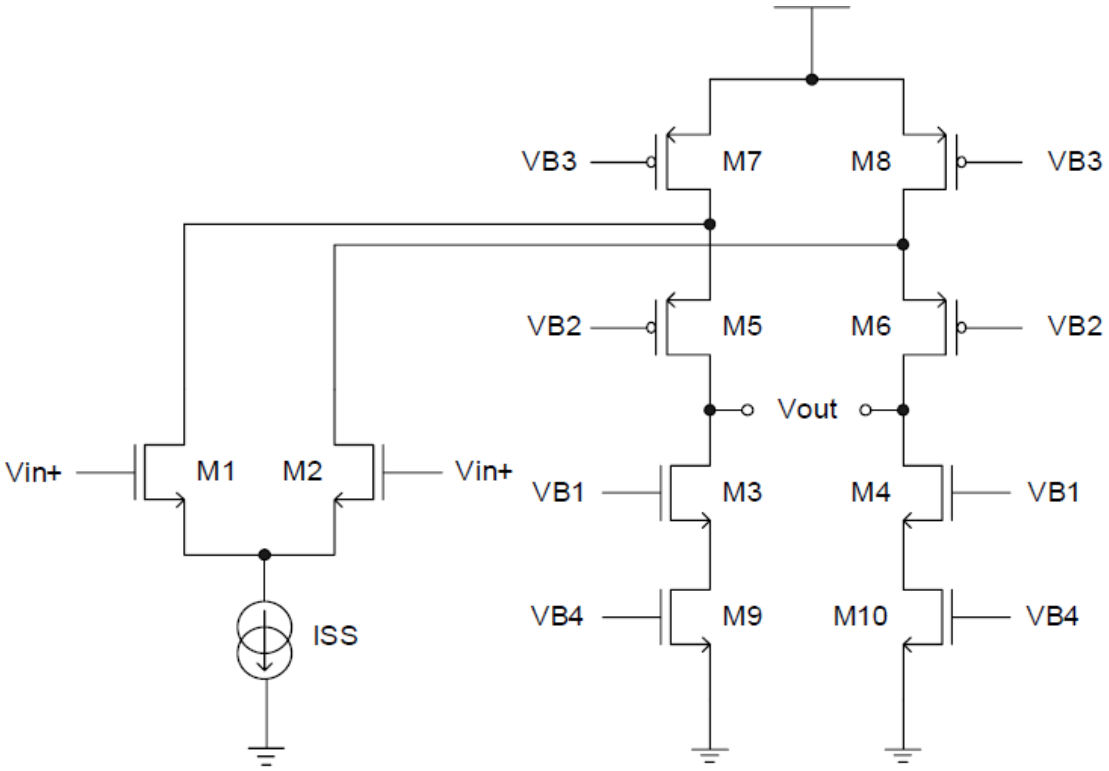


Figure 7.4 Folded cascode amplifier

The primary advantage of this topology is the availability of more headroom for the transistors, hence avoiding stacking of the cascode transistor on top of the input device. However, the folded cascode amplifier generally provides lower gain at lower bandwidth (due to lowering of the pole at the folding point) while consuming higher power.

7.2.1.2 Gain Boosting in Differential Amplifiers with Diode Connected Loads

In differential pair amplifiers with diode-connected loads, the loads consume voltage headroom, limiting the output voltage swing, gain and the input common mode range. In order to obtain a higher gain, the transconductance of the load transistor has to be decreased. This can be done by decreasing the W/L value of the load. However, a disadvantage of this solution is the corresponding increase in the overdrive voltage, which in turn lowers the output common mode level as well as the voltage swing. This problem can be avoided by adding PMOS current sources in parallel to the load transistors, as indicated in Figure 7.5. Since the current is now split between the load and the current source, the W/L value of the load transistor can be decreased without changing the overdrive voltage. Hence, the transconductance of the load can be decreased without compromising the output voltage swing.

If transistors M5 and M6 of Figure 7.5 carry 40% of the drain current of M1 and M2, and the load transistors M3 and M4 carry the remaining 60%, their transconductance decreases by a factor of 2/5 since the W/L ratios of M3 and M4 can also be decreased by the same amount without affecting their overdrive voltage. Thus, the differential gain increases by approximately 5/2 times that of the gain when the PMOS current sources are not included in the circuit. A disadvantage of

this method of increasing the gain is that the current sources add parasitic capacitances to the output node of the circuit, slightly lowering the -3dB bandwidth.

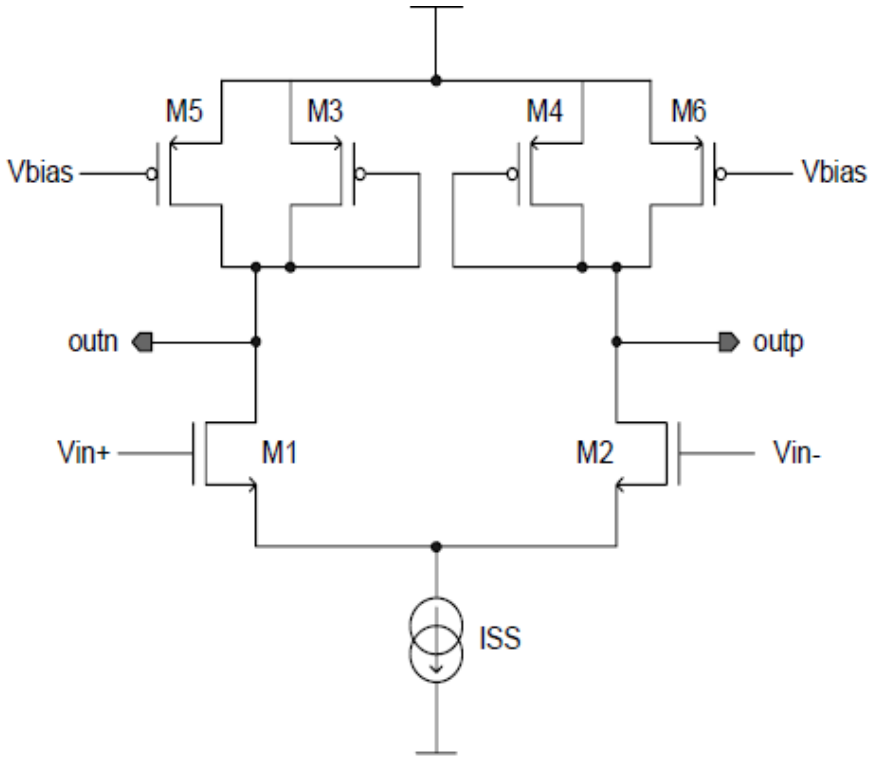


Figure 7.5 Addition of current sources to increase the gain of differential amplifier with diode connected load

7.2.1.3 *Capacitive Neutralization*

One of the commonly used techniques other than cascoding to reduce the Miller effect experienced by the gate-to-drain capacitor of the input transistors in a fully differential amplifier is capacitive neutralization which is illustrated in Figure 7.6. This technique is sometimes used in wideband circuits to increase the bandwidth of multistage amplifiers.

Capacitors C_{GDN} are implemented as transistors M21 and M22 as shown in Figure 7.7. Since M1 and M2 are in the saturation region,

$$V_{DS}(M1, M2) \geq V_{GS}(M1, M2) - V_{T1} \qquad \text{Equation 7-7}$$

$$V_{GD}(M1, M2) \leq V_{T1} \qquad \text{Equation 7-8}$$

$$V_{GS}(M21, M22) \leq V_{T1} \qquad \text{Equation 7-9}$$

Hence, transistors M21 and M22 operate in the cut-off region, with C_{GDN} being equal to the sum of their gate-to-drain and gate-to-source overlap capacitances.

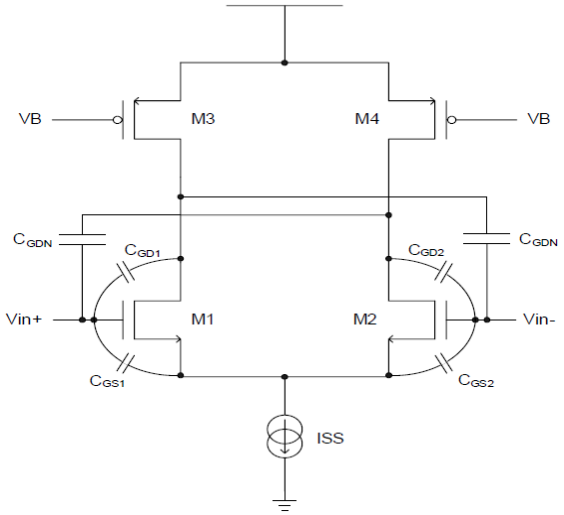


Figure 7.6 Differential amplifier with capacitive neutralization

A disadvantage of the capacitive neutralization technique is that the junction capacitances of transistors M21 and M22 in Figure 7.7 load the nodes to which their drains are connected. This results in a lowering of the pole associated with that node.

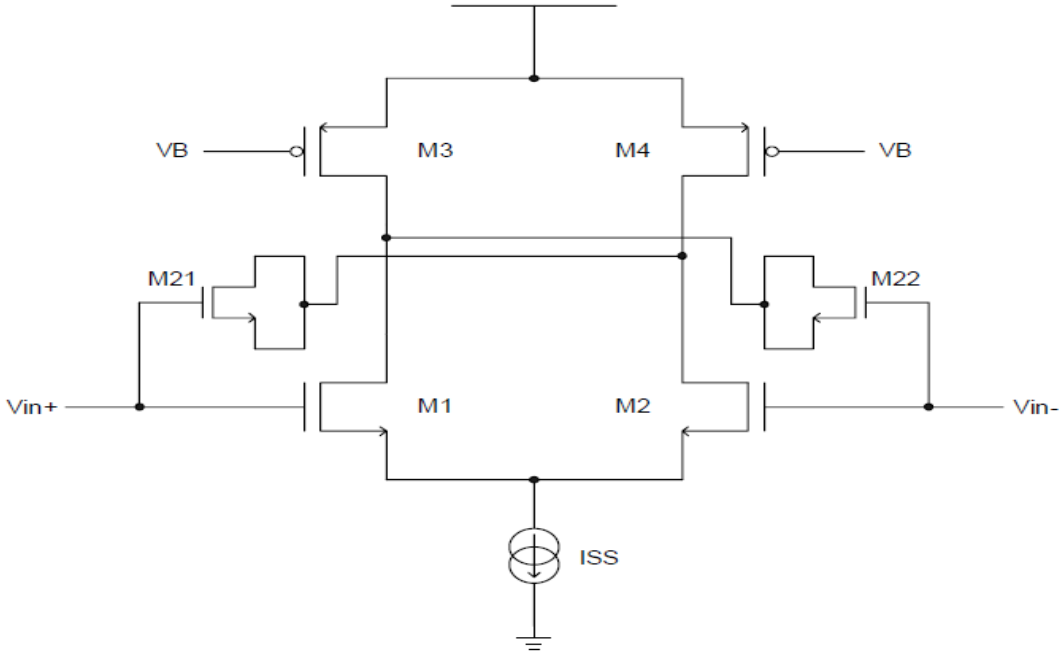


Figure 7.7 Capacitive neutralization with MOS transistors as capacitors

7.2.2 Common Mode and Power Supply rejection Ratios (CMRR) - (PSRR)

A fully differential structure has the advantages of high CMRR and PSRR, and higher output voltage swing compared to the single ended structures.

7.2.2.1 Small Signal Characteristics of Differential Amplifiers

Differential amplifiers are normally considered to have three input terminals (v_{i1}, v_{i2} and ground) and three output terminals (v_{o1}, v_{o2} and ground). Each output can be represented as shown in Equation 7-10 and Equation 7-11.

$$v_{o1} = A_{11}v_{i1} + A_{12}v_{i2} \quad \text{Equation 7-10}$$

$$v_{o2} = A_{21}v_{i1} + A_{22}v_{i2} \quad \text{Equation 7-11}$$

For the given loading conditions, A_{11}, A_{12}, A_{21} and A_{22} are the small signal voltage gains as defined from Equation 7-12 to Equation 7-15.

$$A_{11} = \left.\frac{v_{o1}}{v_{i1}}\right|_{v_{i2}=0} \quad \text{Equation 7-12}$$

$$A_{12} = \left.\frac{v_{o1}}{v_{i2}}\right|_{v_{i1}=0} \quad \text{Equation 7-13}$$

$$A_{21} = \left.\frac{v_{o2}}{v_{i1}}\right|_{v_{i2}=0} \quad \text{Equation 7-14}$$

$$A_{22} = \left.\frac{v_{o2}}{v_{i2}}\right|_{v_{i1}=0} \quad \text{Equation 7-15}$$

The differential output, v_{od} and the common mode output v_{oc} are respectively given in Equation 7-16 and Equation 7-17.

$$v_{od} = v_{o1} - v_{o2} \quad \text{Equation 7-16}$$

$$v_{oc} = \frac{v_{o1} + v_{o2}}{2} \quad \text{Equation 7-17}$$

From Equation 7-14 to Equation 7-19, v_{od} and v_{oc} can be represented as shown in Equation 7-18 and Equation 7-19.

$$v_{od} = \left(\frac{A_{11} - A_{12} - A_{21} + A_{22}}{2}\right)v_{id} + (A_{11} + A_{12} - A_{21} - A_{22})v_{ic} \quad \text{Equation 7-18}$$

$$v_{oc} = \left(\frac{A_{11} - A_{12} + A_{21} - A_{22}}{4}\right)v_{id} + \left(\frac{A_{11} + A_{12} + A_{21} + A_{22}}{2}\right)v_{ic} \quad \text{Equation 7-19}$$

where v_{id} and v_{ic} are the differential and common mode input signals respectively, given in Equation 7-20 and Equation 7-21.

$$v_{id} = v_{i1} - v_{i2} \quad \text{Equation 7-20}$$

$$v_{ic} = \frac{v_{i1} + v_{i2}}{2} \quad \text{Equation 7-21}$$

From Equation 7-22 to Equation 7-25, v_{od} and v_{oc} can be rewritten as shown in Equation 7-22 and Equation 7-23.

$$v_{od} = A_{dm}v_{id} + A_{cm-dm}v_{ic} \quad \text{Equation 7-22}$$

$$v_{oc} = A_{dm-cm}v_{id} + A_{cm}v_{ic} \quad \text{Equation 7-23}$$

where A_{dm} is the differential mode gain which is the change in the differential output voltage per unit change in the differential input, A_{cm} is the common-mode gain which is the change in the common mode output voltage per unit change in the common-mode input, A_{dm-cm} is the differential to common mode gain which is the change in the common mode output per unit change in the differential input, A_{cm-dm} is the common mode to differential mode gain which is the change in the differential output per unit change in the common mode input.

The main advantage of differential amplifiers over single ended amplifiers is that they amplify only changes in the differential input while they reject changes in the common mode input. Hence, for a perfectly balanced differential amplifier, when the input is purely differential, the output will also be purely differential. Therefore, such structures exhibit a high CMRR since the same common mode change appears at both outputs due to symmetry. CMRR is defined in Equation 7-24.

$$CMRR = \left|\frac{A_{dm}}{A_{cm}}\right| \quad \text{Equation 7-24}$$

For the same reason, differential structures also exhibit a higher PSRR, which represents the ability of amplifiers to reject noise in the supply lines.

Due to non-ideal behavior, differential amplifiers usually have a certain common mode gain (A_{cm}) as a result of which the output common mode voltage changes when there is a change in the input common mode, though it does not affect the differential output voltage if the circuit is perfectly balanced. If the output DC level changes by such an extent that it changes the operation region of certain transistors in the same stage or the input transistors of the following stage, the amplifier would fail to function as required. Hence, to fix the output DC level to a certain value and minimize its variation due to changes in the input common mode, common mode feedback circuits are normally employed for this purpose.

7.2.2.2 Common Mode Feedback Circuit (CMFB)

In differential amplifiers with resistive loads, the output common mode voltage is clearly defined as $V_{DD} - I_R R$ where I_R is the current through the load. However, in high gain amplifiers with active current source loads, the common mode voltage is usually not very well defined. The main problem with high gain amplifiers is that normally, a small difference exists in the currents through the PMOS loads and the NMOS input transistors, which flows through the output resistance, creating an output voltage change as given in Equation 7-25.

$$\Delta V = (I_{PMOS} - I_{NMOS}) R_{out} \qquad \text{Equation 7-25}$$

Since the output impedance is quite high and the current error depends on mismatches, the voltage error may be large, pushing certain transistors into the triode region. To avoid this problem and fix the common mode voltage to a certain value which would maximize the output voltage swing and also ensure that all the transistors are in the saturation region, a common mode feedback (CMFB) circuit is designed. This circuit, with a high loop gain, introduces negative feedback and adjusts the output DC level to a specific value irrespective of the changes in the input common mode voltage. It also decreases the variation of the output DC voltage with temperature, supply and process.

A CMFB circuit primarily consists of three blocks; a CM sense block that detects the change in the common mode level, an error amplifier that amplifies the difference between the detected common mode voltage and a reference voltage (V_{CM}) and thirdly, a CM control block which is part of the main amplifier. By varying some parameter of the CM control block such as a voltage or current, the common mode voltage is forced back to the value of V_{CM}. The block diagram of a differential amplifier with a CMFB loop is given in Figure 7.8.

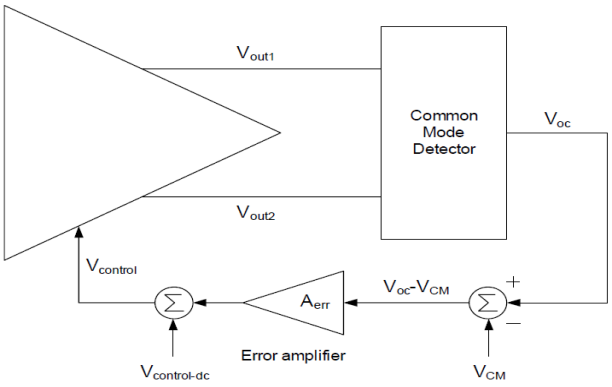


Figure 7.8 Block diagram of a CMFB loop

7.2.3 Linearity

In communication systems, linearity of amplifiers is normally measured by evaluating the third order input intercept point (IIP3). Due to the inherent non-linearity of amplifiers, an input signal $x(t)$ results in an output as given below:

$$y(t) = k_0 + k_1 x(t) + k_2 x^2(t) + k_3 x^3(t) \qquad \text{Equation 7-26}$$

The above relation is based on the assumption that the circuit is memory-less and is driven by a small signal excitation reasonably below the 1-dB compression point (1 dB compression point is the point at which the gain deviates from its ideal small signal value by 1dB). It follows from Equation 7-26 that when the input signal is of the form $x(t) = x\cos(\omega_1 t) + x\cos(\omega_2 t)$, the in-band output of interest is given by Equation 7-27.

$$y_{in-band}(t) = k_1 x[\cos(\omega_1 t) + \cos(\omega_2 t)] + \frac{k_3}{4}x^3\{9\cos(\omega_1 t) + 9\cos(\omega_2 t) + 3[\cos(2\omega_2 - \omega_1)t + \cos(2\omega_1 - \omega_2)t]\}$$ Equation 7-27

It can be observed from the above equation that the third order distortion components include nine new mixing products at *ω1* and *ω2* and three at frequencies $2\omega_1 - \omega_2$ and $2\omega_2 - \omega_1$. The components at $2\omega_1 - \omega_2$ and $2\omega_2 - \omega_1$ are the intermodulation distortion components.

The fundamental component in Equation 7-27 increases with a slope of 1dB/dB while the third order intermodulation component rises at a rate of 3dB/dB with respect to the input power. The third order input intercept point is defined as the input power for which the distortion power at $2\omega_1 - \omega_2$ (or $2\omega_2 - \omega_1$) is the same as the linear output power at ω_1 (or ω_2).

7.2.4 DC Offset

There are two types of offsets in analog circuits that affect the performance of the circuits which are systematic offset and Random offset.

Systematic offset occurs due to the non-idealities of a circuit even when all the elements of the circuit are perfectly matched. Random offset occurs due to mismatches in supposedly matched devices. For example, the threshold voltage variation in matched devices in the same chip gives rise to differences in their overdrive voltages, resulting in random mismatch. These mismatches give rise to a nonzero input offset voltage in amplifiers. The input offset voltage is the differential input voltage that has to be applied to force the differential output to zero. In multistage amplifiers with high gain in each stage, the input-referred offset voltage mainly depends on the design of the first stage.

7.2.4.1 Systematic Offset Voltage

Systematic offset voltage is closely related to the DC power supply rejection ratio of amplifiers. Its dependence on the supply voltage is a more realistic problem in circuits in which the bias currents depend on the supply voltage.

7.2.4.2 Random Offset Voltage

Random offset could occur due to process variations. There could be mismatch in the loads of amplifiers, channel length and width mismatches between transistors, threshold voltage mismatch and many other such variations. Mismatch in the threshold voltage results in a constant offset component independent of the bias current. The circuit has to be laid out as symmetrically as possible to minimize random offset.

7.3 Architectures of VGA

7.3.1 G_m Cell Design

As we discussed before that the differential pair is more preferred than the single ended structure. So, in this section we will realize the results of the G_m cell that is usually used in the design of the VGA and it is the main component in our architectures that will be discussed in the following sections. The G_m cell gain is given in Equation 7-28.

$$A_v = -G_m R_L \qquad \text{Equation 7-28}$$

where G_m is the conductance of the input transistors and R_L is the resistive load.

Figure 7.9 shows the design of the G_m cell. So we are going to show some results for this cell and show the maximum and minimum values that G_m can reach which will help us in the following architectures to get the gain. Also, we will some results for the Total Harmonic Distortion (THD) that will help us achieving the linearity of the system.

The design is analyzed and results are calculated using Cadence tool.

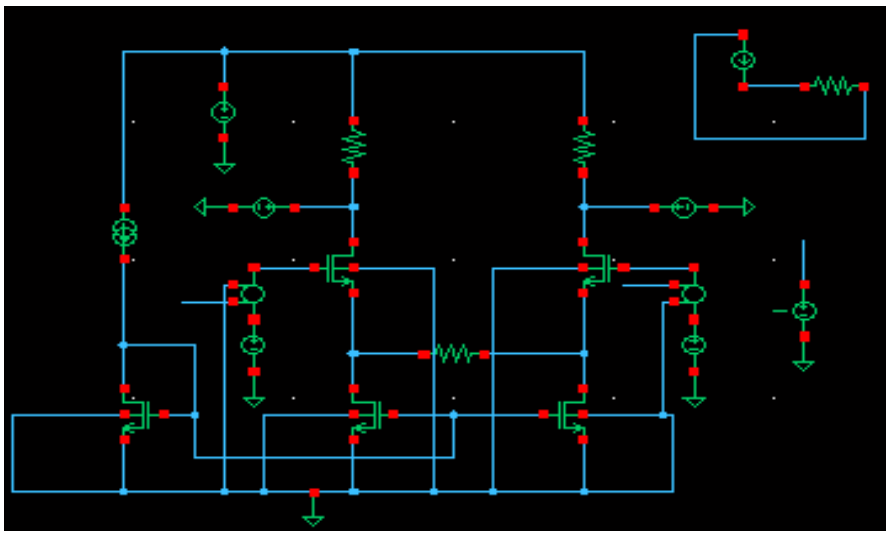


Figure 7.9 G_m Cell

The degeneration resistor (R) plays an important role in varying the G_m. Figure 7.10 analyzes the relation between G_m and R. For $R = 1\Omega$, $G_m = 10\ mA/V$ while for $R = 290\Omega$, $G_m = 3.5\ mA/V$ and so on.

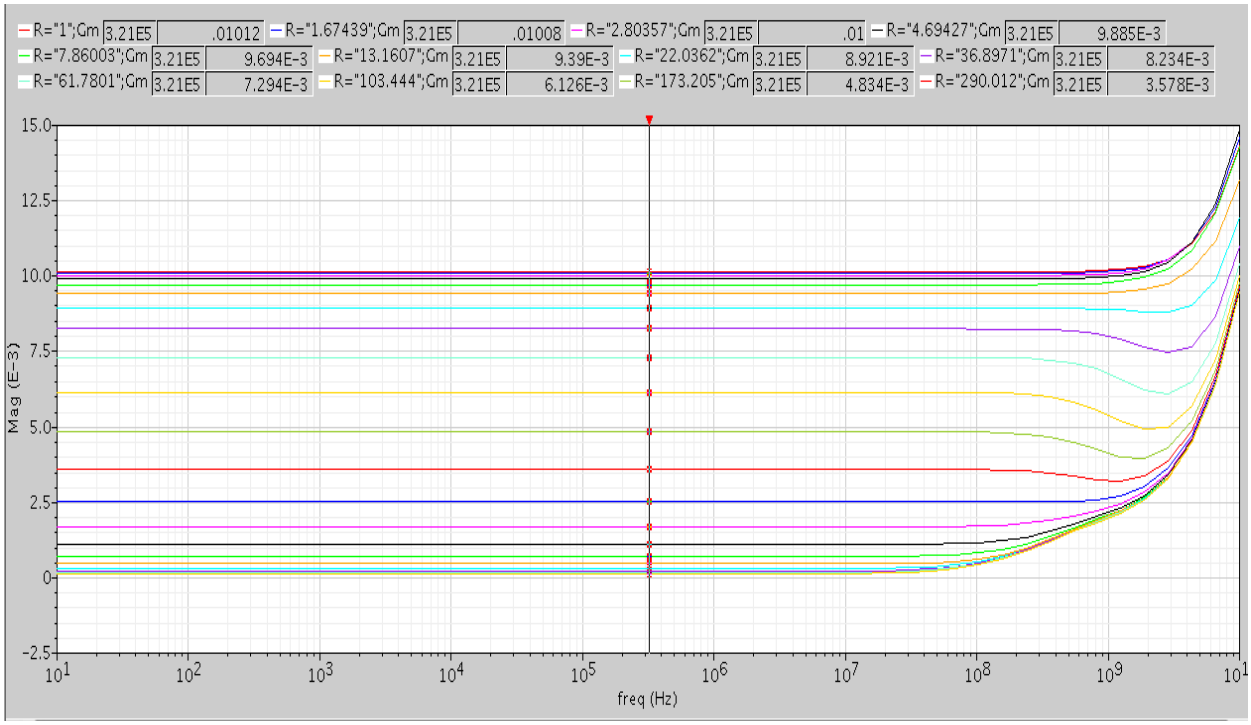


Figure 7.10 *Gm vs R*

7.3.1.1 High G_m

For $R = 1\Omega,\ G_m = 10\ mA/V$

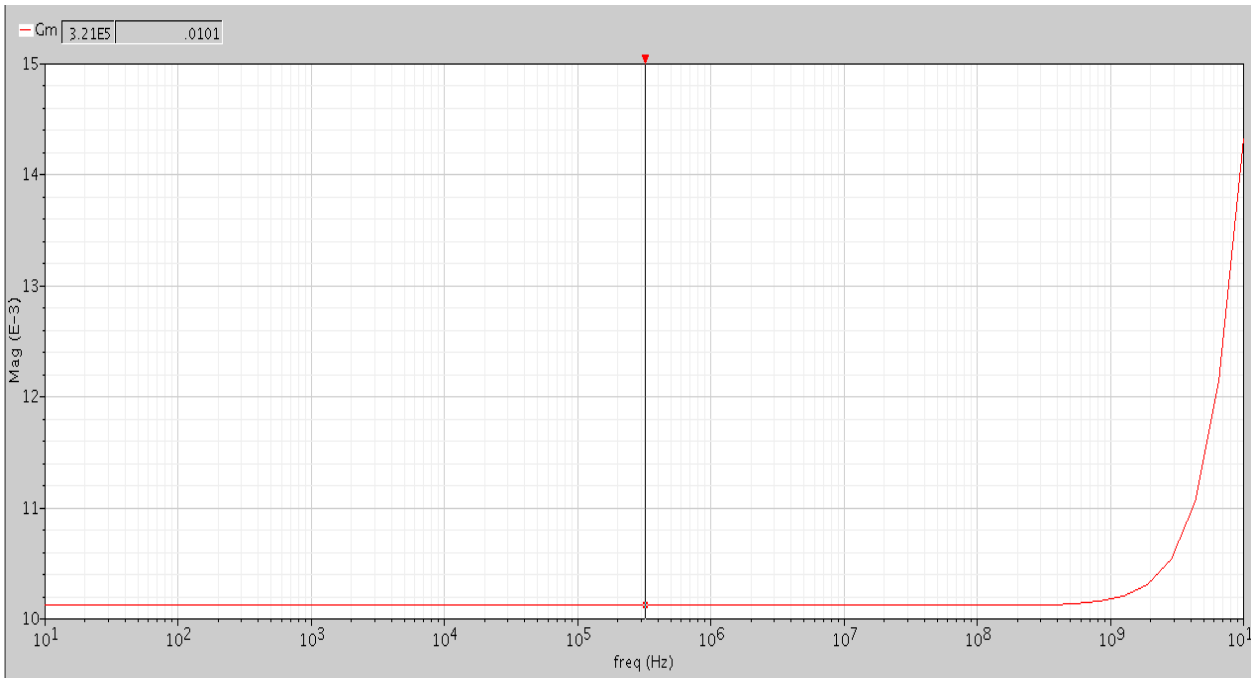


Figure 7.11 High *Gm*

$$THD = 30\ dB$$

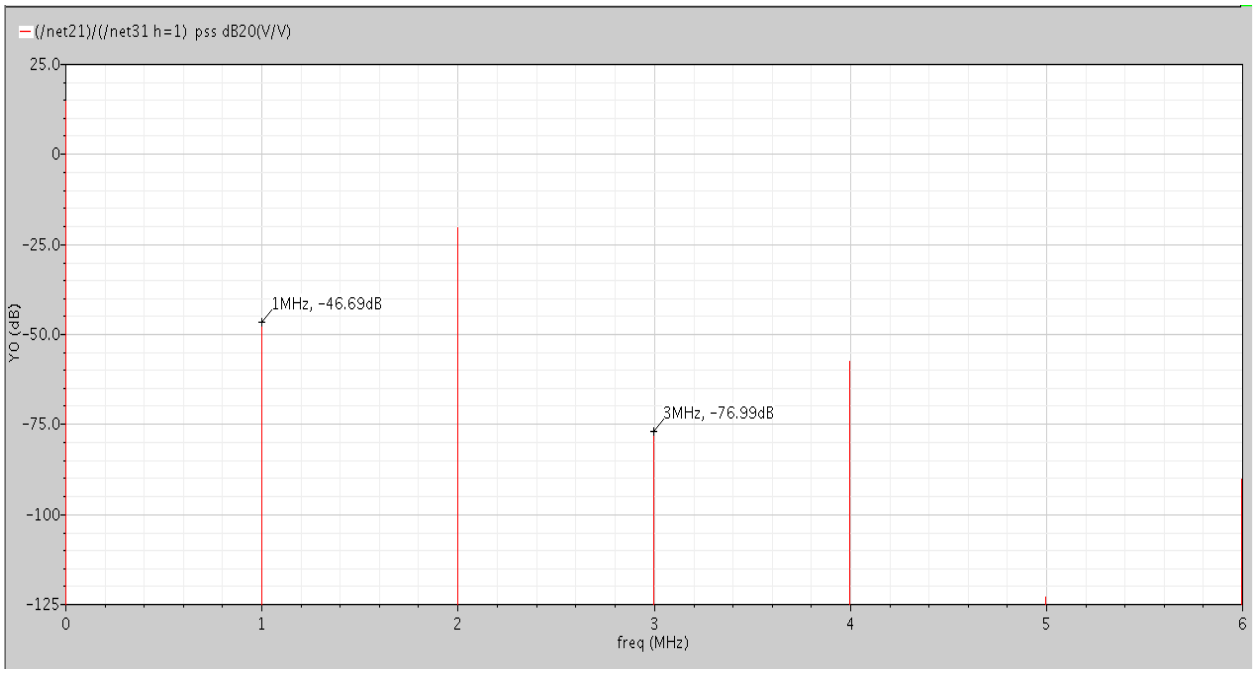


Figure 7.12 THD for high Gm

7.3.1.2 Low G_m

For $R = 25\ K\Omega,\ G_m = 0.1\ mA/V$

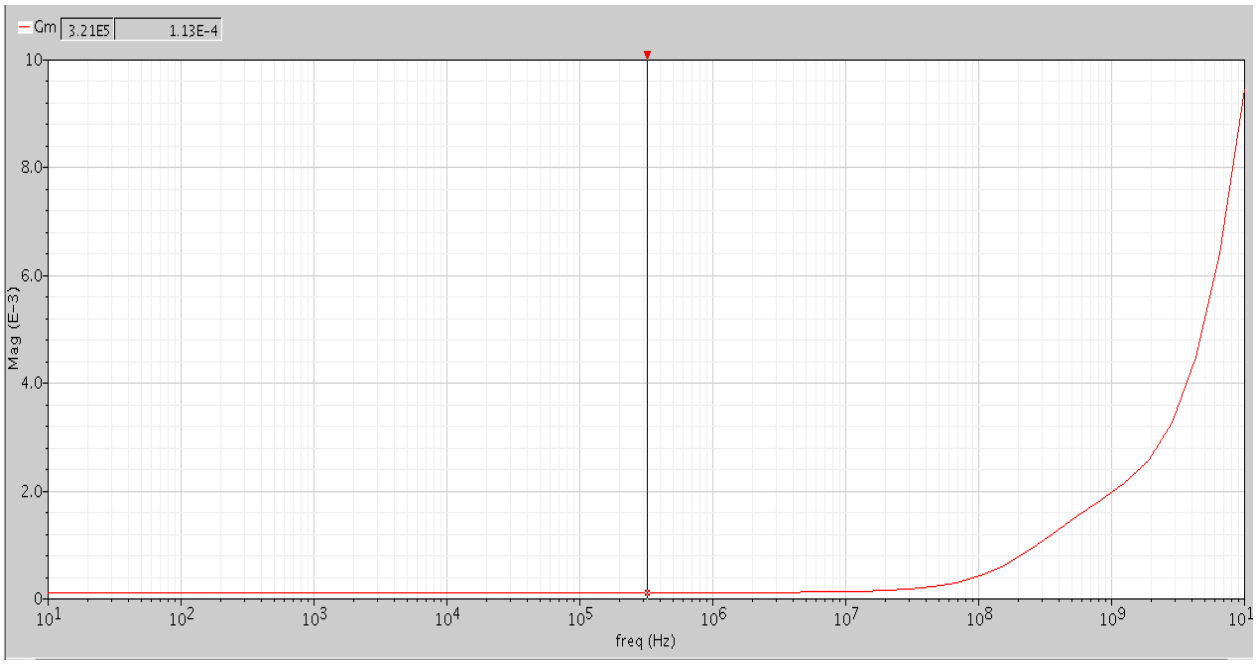


Figure 7.13 Low Gm

$$THD = 90\ dB$$

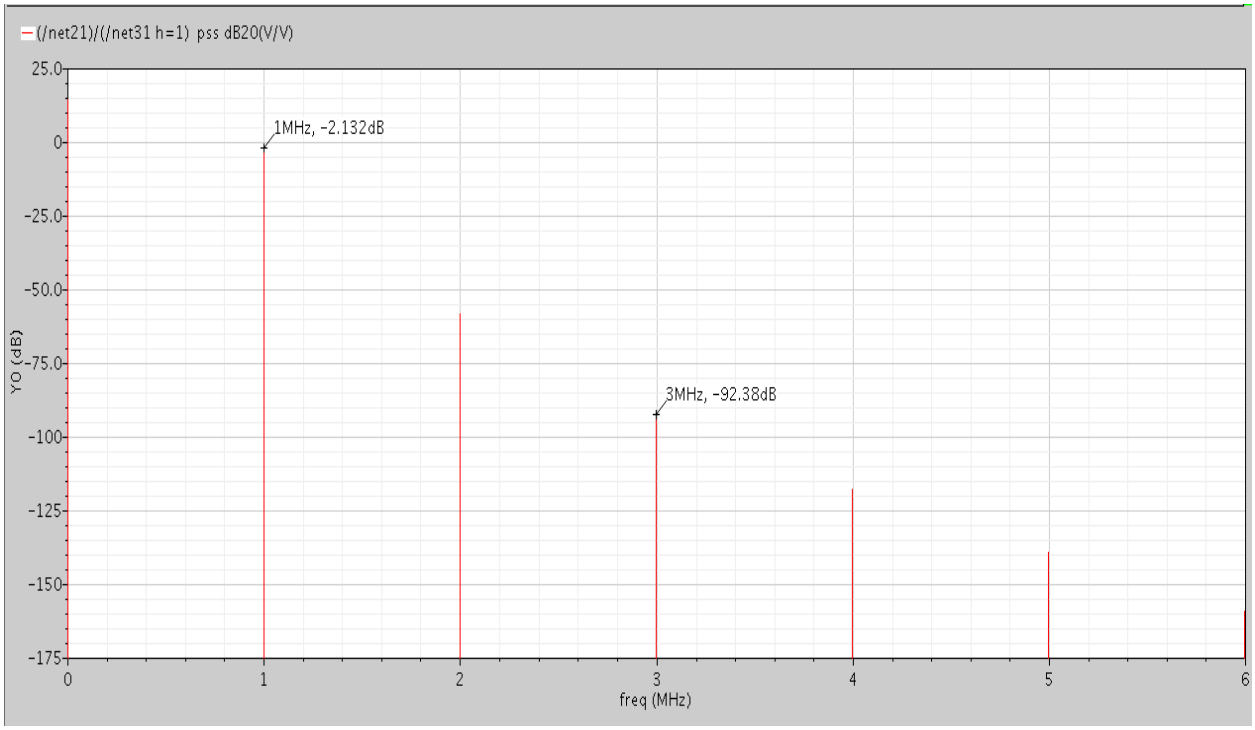


Figure 7.14 THD for low Gm

7.3.2 84 dB-Linear Low-Power VGA

7.3.2.1 Introduction

The VGA proposed utilizes a new exponential approximation equation which is given in Equation 7-29.

$$e^{2ax} = \frac{e^{ax}}{e^{-ax}} \cong \frac{[k + (1 + ax)^2]}{[k + (1 - ax)^2]} \qquad \text{Equation 7-29}$$

where k is a constant. For $k = 1$, the numerator and the denominator of Equation 7-29 are the second-order Taylor series approximation of the exponential function. The dB-linear range extends to more than $60\ dB$ with a linearity error of less than $\pm\ 0.5\ dB$, which is a serious improvement compared to the Taylor series approximation and pseudo-exponential functions. Moreover, the input range of x for the new approximation equation is much larger than that of the Taylor series approximation and pseudo-exponential functions.

7.3.2.2 The Control Circuit Block

The circuit for generating the numerator and the denominator of Equation 7-29 is given in Figure 7.15, where all transistors operate in saturation mode. In Figure 7.15, the body terminals of P-MOS and N-MOS transistors are tied to $V_{DD}$and V_{SS}, respectively, and the lengths of transistors M1 and M2 are chosen long enough that the second order effects can be neglected.

In Figure 7.15, to guarantee the saturation-mode operation of transistors M1 and M2, the control voltage V_C must stay within a range of $(V_{SS} + V_{THn}) \sim (V_{DD} - |V_{THp}|)$. The drain currents of transistors M1 and M2 in Figure 7.15 are given as shown in Equation 7-30 and Equation 7-31.

$$I_{D1} = K_p\left(V_C - V_{DD} + |V_{THp}|\right)^2 \qquad \text{Equation 7-30}$$

$$I_{D2} = K_n(V_C - V_{SS} - V_{THn})^2 \qquad \text{Equation 7-31}$$

where Kp and Kn are constants, and $VTHp$ and $VTHn$ the threshold voltages of the P-MOS and N-MOS transistors, respectively.

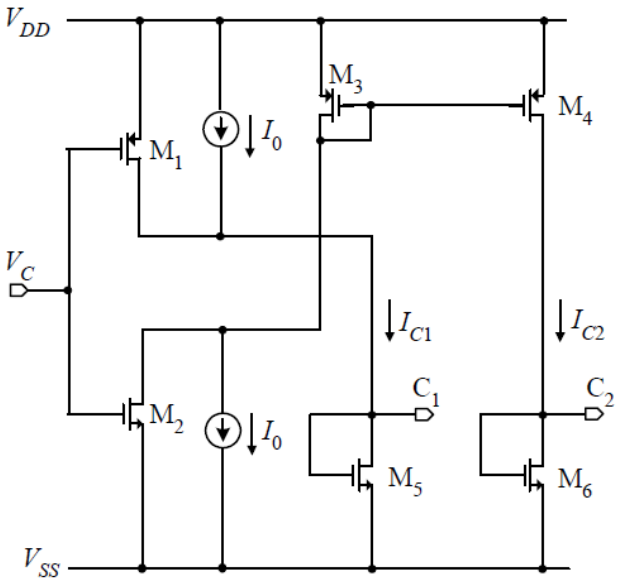


Figure 7.15 The proposed circuit schematic of control block

In Figure 7.15, the currents $I_{D1}$and I_{D2} are added by a bias current I_o. The resulting currents I_{C1} and I_{C2}, after some mathematical manipulation, are given in Equation 7-32 and Equation 7-33.

$$I_{C1} = K_p(V_{DD} - |V_{THp}|)^2 * \left\{ \frac{I_o}{K_p(V_{DD} - |V_{THp}|)^2} + \left(1 - \frac{V_C}{(V_{DD} - |V_{THp}|)}\right)^2 \right\} \quad \text{Equation 7-32}$$

$$I_{C2} = K_n(V_{SS} + V_{THn})^2 * \left\{ \frac{I_o}{K_n(V_{SS} + V_{THn})^2} + \left(1 - \frac{V_C}{(V_{SS} + V_{THn})}\right)^2 \right\} \quad \text{Equation 7-33}$$

Assuming $K_p = K_n = K, V_{DD} = -V_{SS}$ *and* $|V_{THp}| = V_{THn} = V_{TH}$

From Equation 7-32 and Equation 7-33, the ratio I_{C2}/I_{C1}, the relation that is utilized in the amplifying block which will be described later, can be given by Equation 7-34.

$$\frac{I_{C2}}{I_{C1}} = \frac{\frac{I_o}{K(V_{DD}-|V_{TH}|)^2} + \left(1 + \frac{V_C}{(V_{DD}-|V_{TH}|)}\right)^2}{\frac{I_o}{K(V_{DD}-|V_{TH}|)^2} + \left(1 - \frac{V_C}{(V_{DD}-|V_{TH}|)}\right)^2} = \frac{[k + (1+ax)^2]}{[k + (1-ax)^2]} \quad \text{Equation 7-34}$$

where $k = \frac{I_o}{K(V_{DD}-|V_{TH}|)^2}$, $a = \frac{1}{(V_{DD}-|V_{TH}|)}$ *and* $x = V_C$

As can be seen in Equation 7-34, the current ratio I_{C2}/I_{C1}, which is a function of the control voltage VC, is equivalent to Equation 7-29. In Equation 7-34, adjusting the bias current I_o can vary the value of the constant k, resulting in different dB-linear ranges.

From Equation 7-31 and Equation 7-32, by assuming $V_{DD} = -V_{SS}$, the exponential approximation equation as a function of the control voltage VC is obtained and shown in Equation 7-34. Considering the case where $V_{DD} = -V_{SS} = 0.9V$; if we shift all supply voltage nodes

by $0.9V$, $V_{DD} = 1.8V$, $V_{SS} = 0V$ and the input voltage VC is from V_{THn} to $1.8 - |V_{THp}|$. In this case, the exponential relation shown in Equation 7-34 is still maintained.

7.3.2.3 Variable Gain Amplifier Circuit:

The variable gain circuit block is composed of an input source-coupled pair (M9 and M12) and diode-connected loads (M10 and M11). The sum of currents through input pair and loads is equal to that of the upper PMOS current sources (M7 and M8). The two currents I_{C1} and I_{C2} from the control block in Figure 7.15 are mirrored to M13 and M14 in Figure 7.16. The currents in the loads and the input pair are respectively controlled by I_{C1} and I_{C2}.

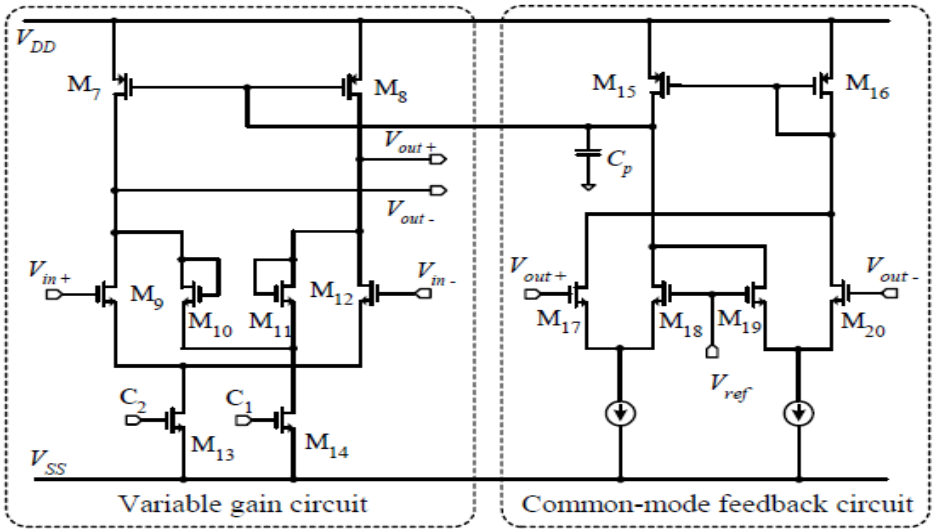


Figure 7.16 Circuit schematic of amplifying block

From Equation 7-32, Equation 7-33 and Equation 7-34, the sum of the currents through M7 and M8 is equal to 2(k+1+a^2x^2), which is a function of the control voltage $V_C(x = V_C)$. Upon the change of currents in M7 and M8, the common-mode feedback circuit with a high gain is used to stabilize the output common mode level as shown in Figure 7.16.

The differential gain of the amplifier in Figure 7.16 is expressed as shown in Equation 7-35.

$$A_v = \frac{g_{m-input}}{g_{m-load}} = \sqrt{\frac{(W/L)_{input} I_{C2}}{(W/L)_{load} I_{C1}}} \qquad \text{Equation 7-35}$$

where $g_{m-input}$ is the transconductance of the input transistors (M9 and M12), g_{m-load} is the transconductance of the diode connected loads (M10 and M11), and I_{C2} and I_{C1} are given in Equation 7-34.

From Equation 7-34 and Equation 7-35 the differential gain is calculated as in Equation 7-36

$$A_v = \frac{g_{m-input}}{g_{m-load}} = \sqrt{\frac{(W/L)_{input}}{(W/L)_{load}} * \frac{[k + (1+ax)^2]}{[k + (1-ax)^2]}} \qquad \text{Equation 7-36}$$

which is the same form of expression as Equation 7-29.

As in Equation 7-36, the differential gain is a function of the control voltage V_C. By adjusting the bias current I_o to get $k = 0.15$, the amplifying block in Figure 7.16 can provide more than $60\ dB$ of the gain variation.

The bandwidth of the proposed VGA is a function of the output impedance and capacitance. The load is mainly composed of two diode-connected transistors (M10 and M11), so that the output impedance is proportional to g_{m-load}, which is determined by the current I_{C1}. Since I_{C1} varies with the square root of the control voltage Vc, the bandwidth of the proposed VGA has different values between the low and high-gain modes. The g_{m-load} increases and decreases at low and high-gain modes, leading to wide and narrow bandwidths of the VGA, respectively. Figure 7.16 shows the block diagram of the overall VGA.

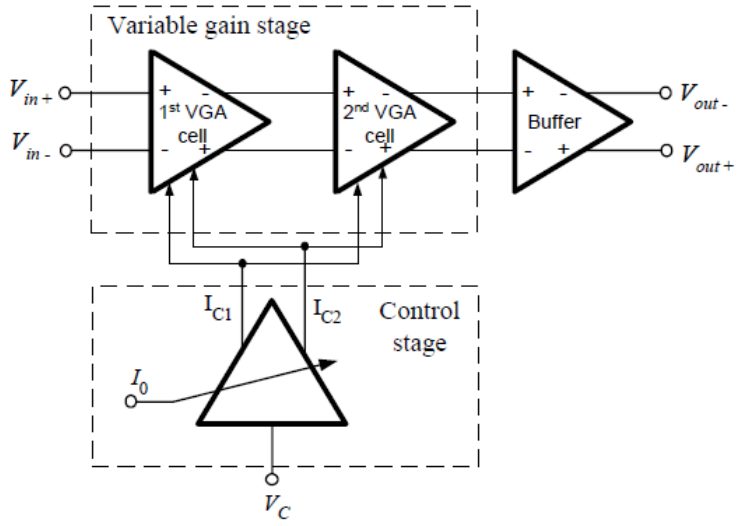


Figure 7.17 Block diagram of the proposed 2-stage VGA

The VGA adopts two amplifying blocks in cascade so that more than $120\ dB$ of the gain variation can be obtained. The first VGA cell is the same as the second one, which is shown in Figure 7.16. The control stage in Figure 7.17 generates two currents I_{C1} and I_{C2}, which are functions of the control voltage Vc. The bias current, I_o, can be tunable to change the value of k in Equation 7-36 so that the gain control range of the VGA in Figure 7.16 can be adjusted. The buffer circuit shifts the differential gain to positive values, and provides 50Ω output impedance for the conveniences of the measurements.

7.3.2.4 Simulation Results

The threshold voltages of NMOS and PMOS transistors in the given technology are about $0.4V$, hence V_C is from 0.4 to $1.4V$. Figure 7.18 shows the measured gain versus control voltage V_C at $20MHz$.

In Figure 7.18, the VGA shows a total gain range of $84\ dB$ and an $80\ dB$ of the gain variation with a linearity error of less than $\pm 1\ dB$ while dissipating less than $3.6mA$. At maximum gain, g_{m-load} is minimal so that the bandwidth is smallest; the measured bandwidth at $36\ dB$ gain

is $40MHz$. The bandwidth increases when the gain is reduced. At minimum gain, g_{m-load} is maximized and the 3dB-bandwidth is about $1GHz$.

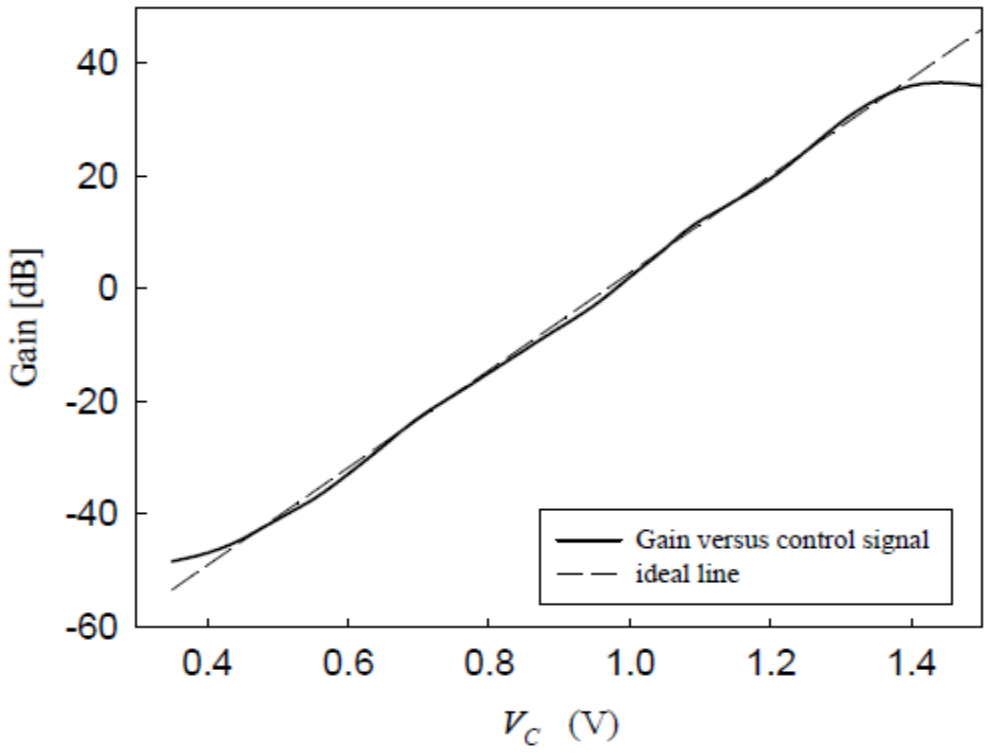


Figure 7.18 Measured gain versus control voltage Vc

7.3.2.5 Conclusion

The new exponential approximation function with a very wide dB-linear control range is utilized, so that fewer VGA stages are needed to satisfy the required dB gain range. Consequently, the chip size (or the cost) and the power consumption are reduced drastically, while still achieving a very wide gain control range.

The VGA topology that implements the new exponential approximation function is newly proposed, implemented, and verified through the measurement. The proposed VGA is implemented in $0.18\mu m$ CMOS technology and measurements show the gain variation of more than $84\ dB$ in two cascaded stages, and 80 dB-linear range with a linearity error of less than $\pm 1\ dB$. The 3dB-bandwidth varies from 40MHz bandwidth at maximum gain to 1GHz bandwidth at minimum gain. The power dissipation is less than $3.6mA$ from $1.8V$ supply.

7.3.3 DB-Linear VGA with Pre-distortion Compensation

7.3.3.1 Introduction

In wireless LAN applications (e.g. IEEE 802.11a/g), receiver sensitivity is typically around $-80\ dBm$ and signal bandwidth is $20\ MHz$ (double sided). Assuming that a direct conversion receiver RF front-end can provide $25\ dB$ gain and the required signal swing is over 1 V to efficiently drive an analog-to-digital converter, the VGA needs to have a dynamic range of $60\ dB$ with a bandwidth larger than $10\ MHz$ over the entire adjustable gain range.

In an automatic gain control (AGC) loop, to maintain its settling time independent of the input signal levels and a large dynamic control range, an exponential gain control characteristic is required. However, in CMOS technology, it is difficult to realize the exponential or logarithmic function because of its inherent square or linear characteristics. Although the transistor operating in sub-threshold region has exponential characteristic, it is generally not preferred due to other unfavorable effects, such as noise and bandwidth. Thus, some of the reported CMOS dB-linear VGAs are based on a pseudo-exponential function given in Equation 7-37.

$$e^x \cong \frac{1+x}{1-x} \qquad \text{Equation 7-37}$$

The approximation errors of this pseudo-exponential function to the ideal one is within 5% only when $|x| < 0.32$. Alternatively, the Taylor's series can be utilized to approximate the exponential function. Typically, the second-order approximation can be expressed as

$$e^x \cong 1 + x + \frac{x^2}{2} \qquad \text{Equation 7-38}$$

The approximation error of Equation 7-38 can be less than 5% when -0.575 < *x* < 0.815 is satisfied. Obviously, Taylor's series approximation has a larger input dynamic range than the pseudo-exponential method for the same approximation error. In this paper, a new dB-linear VGA based on Taylor's series approximation is proposed.

The block diagram of the proposed VGA is shown in Figure 7.19 It consists of three cascade linear VGA stages and an exponential function circuit. The linear VGA provides a $60\ dB$ variable gain with a good linearity. The gain of the linear VGA is controlled by a newly proposed exponential function circuit, and thus a dB-linear VGA with large input dynamic range is realized.

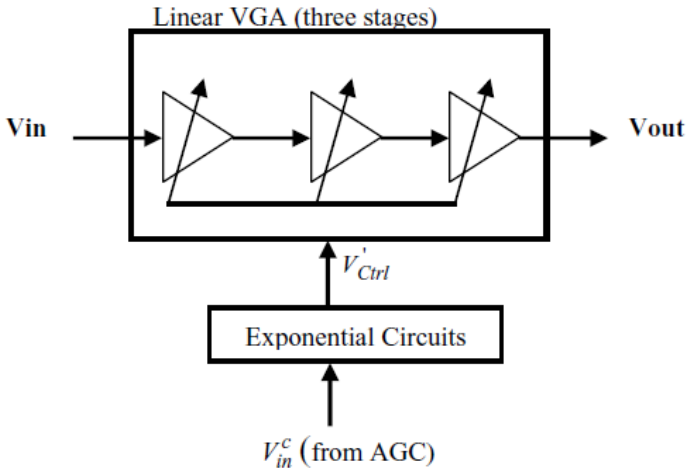


Figure 7.19 Block diagram of the dB-linear VGA

7.3.3.2 *Differential Linear Variable Gain Amplifier*

The design of VGA is a demanding task because of the needs for wide dynamic range, precise gain control, low noise figure and good linearity. In CMOS technology, basically there are three methods to control the gain of a VGA, namely, by varying (1) the tranconductance of a MOS device operated in the saturation region, (2) the load resistance, and (3) the source degeneration resistance which is often implemented by a MOS device operated in the linear region. Among them, the last method has the advantages of good linearity, low noise figure and low power dissipation because the source degeneration does not impose any penalty, due to the overdrive voltage $V_{gs} - V_{th}$, on voltage headroom in a simple differential pair. Thus, in this paper, the differential amplifier with source degeneration is chosen to implement three linear VGA stages.

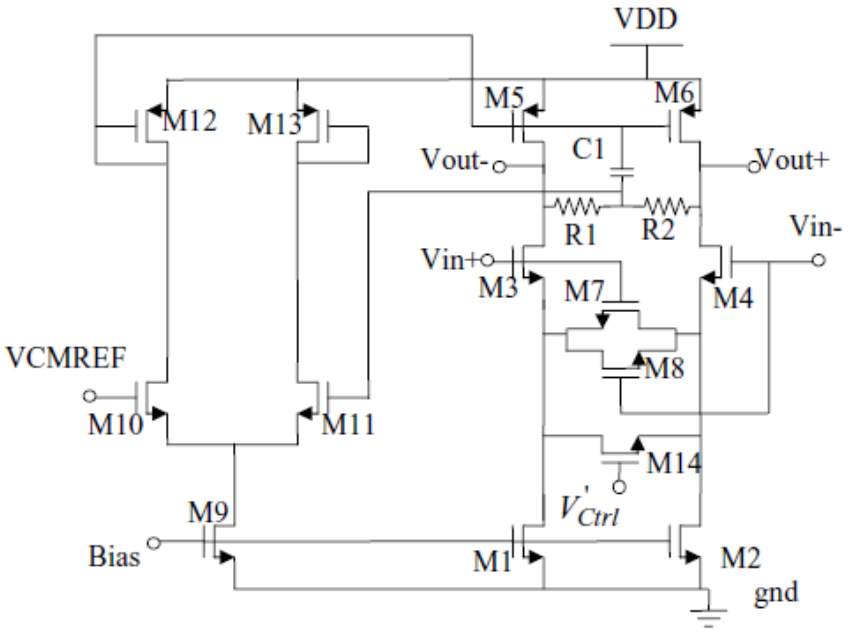


Figure 7.20 Linear VGA

One stage of the differential linear VGA is shown in Figure 7.20. M3 and M4 form the linear transconductance pair. M5 and M6 act as the active load to provide high gain. M7 and M8 are used to improve the linearity. The common-mode feedback circuit consists of R_1, R_2 and M10-M13. The

gain of the VGA can be adjusted continuously over a large range ($\sim 20\ dB$) through the source degeneration transistor M14.

The gain of the VGA can be expressed as shown in Equation 7-39.

$$A_v = -G_s R_d \frac{g_m}{g_m + G_s} \qquad \text{Equation 7-39}$$

where g_m, G_s and R_d represent the transconductance of the input transistor, the conductance of source degeneration transistor, and the load resistance, respectively. G_s is given by Equation 7-40.

$$G_s = \mu_n C_{ox} \left(\frac{W}{L}\right)(V_{gs} - V_{th}) \qquad \text{Equation 7-40}$$

Obviously, if $g_m \gg G_s$ and $\frac{g_m}{g_m + G_s} \cong 1$, A_v can be rewritten as shown in Equation 7-41.

$$A_v = -G_s R_d \frac{g_m}{g_m + G_s} \approx -\mu_n C_{ox} \left(\frac{W}{L}\right)_{14} (V_{gs14} - V_{th}) \qquad \text{Equation 7-41}$$

Therefore, the gain of the VGA can be linearly controlled by the gate voltage of M14.

7.3.3.3 Exponential Function Generation Circuit

The Taylor's series expansion of a general exponential function can be expressed as shown in Equation 7-42.

$$e^{\frac{b}{a}x} = 1 + \frac{b}{a}x + \frac{1}{2!}\left(\frac{b}{a}x\right)^2 + \frac{1}{3!}\left(\frac{b}{a}x\right)^3 + \cdots \frac{1}{n!}\left(\frac{b}{a}x\right)^n + \cdots \qquad \text{Equation 7-42}$$

where *a* and *b* are two constants. If $\left|\frac{b}{a}x\right| \ll 1$, the higher order terms can be neglected, and Equation 7-42 becomes

$$2a^2 e^{\frac{b}{a}x} \approx a^2 + (a + bx)^2 \qquad \text{Equation 7-43}$$

A large range of x in Equation 7-43 can be attained if the constant a and b are carefully chosen to ensure $-0.575 < \frac{b}{a}x < 0.815$ and 5% approximation error could be accepted. Based on Equation 7-43, a wide-range exponential voltage generation circuit is proposed and shown in Figure 7.21. It includes three building blocks, namely, a linear V-I converter, a constant current source and a current square circuit (CSC).

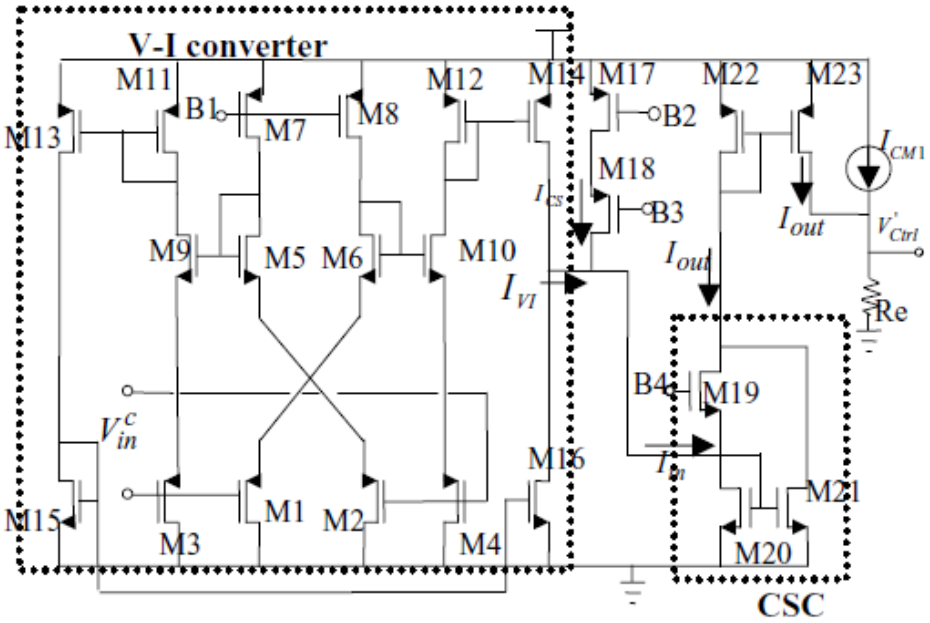


Figure 7.21 Exponential control circuit where B1 to B4 denote four different dc bias voltages

The output current of the CSC can be written as in Equation 7-44

$$I_{out} = 2I_o + \frac{I_{in}^2}{8I_o} \qquad \text{Equation 7-44}$$

The output current of the V-I converter can be expressed by $I_{VI} = 2g_b V_{in}^c$ where g_b is the equivalent transconductance of the V-I converter. By adding a constant current $I_{CS} = 4I_o$ to I_{VI} and assuming that the input current of CSC, $I_{in} = I_{VI} + I_{CS}$, as shown in Figure 7.21, the output current of CSC can be rewritten as in Equation 7-45

$$\begin{aligned} I_{out} &= 2I_o + \frac{(I_{VI} + I_{CS})^2}{8I_o} \qquad \text{Equation 7-45} \\ &= \frac{1}{8I_o}[(4I_o)^2 + (4I_o + 2g_b V_{in}^c)^2] \end{aligned}$$

By properly sizing the transistors, $-0.575 < \frac{2g_b V_{in}^c}{4I_o} < 0.815$, can be ensured for the entire operation range of V_{in}^c.

Using this condition and Equation 7-43, Equation 7-45 can be rewritten as

$$I_{out} \approx \frac{1}{8I_o} 2(4I_o)^2 exp\left(\frac{2g_b V_{in}^c}{4I_o}\right) = 4I_o exp\left(\frac{g_b V_{in}^c}{2I_o}\right) \qquad \text{Equation 7-46}$$

Thus an approximately exponential current is realized in Equation 7-46. Furthermore, the exponential control voltage V_{Ctrl} can be easily generated by passing the CSC output current through a resistor R_e as shown in Equation 7-47.

$$V_{Ctrl} = R_e I_{out} = 4R_e I_o exp\left(\frac{g_b V_{in}^c}{2I_o}\right) \quad \text{Equation 7-47}$$

This voltage is used to control the gain of the linear VGA in Figure 7.20 and hence a dB-linear VGA is realized.

7.3.3.4 dB Linearity Compensation

Ideally, a linear VGA whose gain is controlled by an exponential function voltage would exhibit good dB linearity. However, in practice, it is affected by two factors:

1. The nonzero source and the threshold voltage of the degeneration transistor M14
2. The increase of Gs with V_{in}^c.

7.3.3.4.1 The Non-zero Source and the Threshold Voltage

Firstly, considering the effect of gate-source voltage and the threshold voltage of M14, rewrite Equation 7-39 as

$$\begin{aligned} A_v &= -\mu_n C_{ox}\left(\frac{W}{L}\right)(V_g - V_s - V_{th})R_d \\ &= K(V_{Ctrl} - V_{st})R_d \end{aligned} \quad \text{Equation 7-48}$$

where $V_{st} = V_s + V_{th}\ and\ K = \mu_n C_{ox}\left(\frac{W}{L}\right)$

Substituting in Equation 7-47 and Equation 7-48 and taking logarithm of the both sides, it yields to Equation 7-49.

$$20\log|A_v| = 20\log KR_d + 20\log\left[4R_e I_o \exp\left(\frac{g_b V_{in}^c}{2I_o}\right) - V_{st}\right] \quad \text{Equation 7-49}$$

Obviously, since V_{st} is not zero, the gain of the VGA will not be dB-linearly proportional to V_{in}^c. To compensate this nonlinearity, a fixed current $I_{CM1} = \frac{Vst}{Re}$ is added to I_{out} and the control voltage becomes

$$V'_{Ctrl} = (I_{out} + I_{CM1})R_e = V_{Ctrl} + V_{st}$$

Substituting it into Equation 7-48 and taking logarithm of both sides, yielding to Equation 7-50.

$$\begin{aligned} 20\log|A_v| &= 20\log KR_d + 20\log\left[4R_e I_o \exp\left(\frac{g_b V_{in}^c}{2I_o}\right)\right] \\ &= 20\log(4KR_e R_d I_o) + \frac{10g_b}{I_o}V_{in}^c \end{aligned} \quad \text{Equation 7-50}$$

Therefore, the effect of V_{st} is removed.

7.3.3.4.2 The Increase of Gs with V_{in}^{c}

Secondly, the gain linearity of the VGA is also affected by the dependence of Gs on V_{in}^{c}. To elaborate this, Substituting Equation 7-40 and Equation 7-48 into Equation 7-39 and assuming the nonlinearity caused by V_{st} has been removed, it yields to Equation 7-51.

$$A_v = -R_d g_m \frac{4KR_e I_o exp\left(\frac{g_b V_{in}^{c}}{2I_o}\right)}{g_m + 4KR_e I_o exp\left(\frac{g_b V_{in}^{c}}{2I_o}\right)} \qquad \text{Equation 7-51}$$

When taking logarithm of the both sides, Equation 7-51 becomes

$$20\log|A_v| = 20\log(4KR_eR_dI_og_m) + 10\frac{g_bV_{in}^{c}}{I_o} - 20\log\left\{g_m + K\left[4I_o exp\left(\frac{g_bV_{in}^{c}}{2I_o}\right)\right]\right\} \qquad \text{Equation 7-52}$$

Similar to Equation 7-49, the third term in Equation 7-52 deteriorates the dB linearity. This effect becomes more severe when the input control voltage V_{in}^{c} is high.

To compensate this effect, the rate of the gain variation versus the control voltage V_{Ctrl} when V_{in}^{c} is high may be purposely made faster than that defined by the exponential function. This is essentially a pre-distortion technique that, to some extent, compensates the nonlinearity introduced by the third term in Equation 7-52.

One of possible implementations of this pre-distortion is to make $I_{CS} > 4I_o$. It can be proved that with such an implementation, the output voltage of the exponential circuit (V_{Ctrl}) increases in a faster rate than the ideal exponential one when V_{in}^{c} is high.

7.3.3.5 Simulation Results:

Figure 7.22 shows the simulation result for the exponential circuit with the pre-distortion technique when I_{CM1} is set to zero.

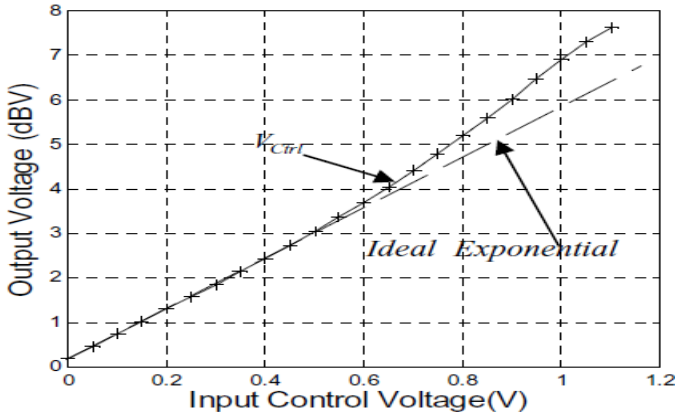


Figure 7.22 V_{ctrl} vs. the input of exponential function circuit V_{in}^{c}

It can be seen that at high input control voltage (V_{in}^{c}), V_{Ctrl} increases faster than the ideal dB-linear rate. As described in the previous section, this can be used to compensate the effect of the increased Gs on the overall gain linearity.

Figure 7.23 shows the results of the dB-linear VGA. For the uncompensated VGA (linear VGA + exponential function circuit without dB-linearity compensation), it shows a log-like gain variation with the input control voltage. However, for the compensated VGA (linear VGA + exponential function circuit with dB-linearity compensation), the dB-linearity has been greatly improved and is close to the ideal case. This demonstrates the effectiveness of the proposed compensation techniques.

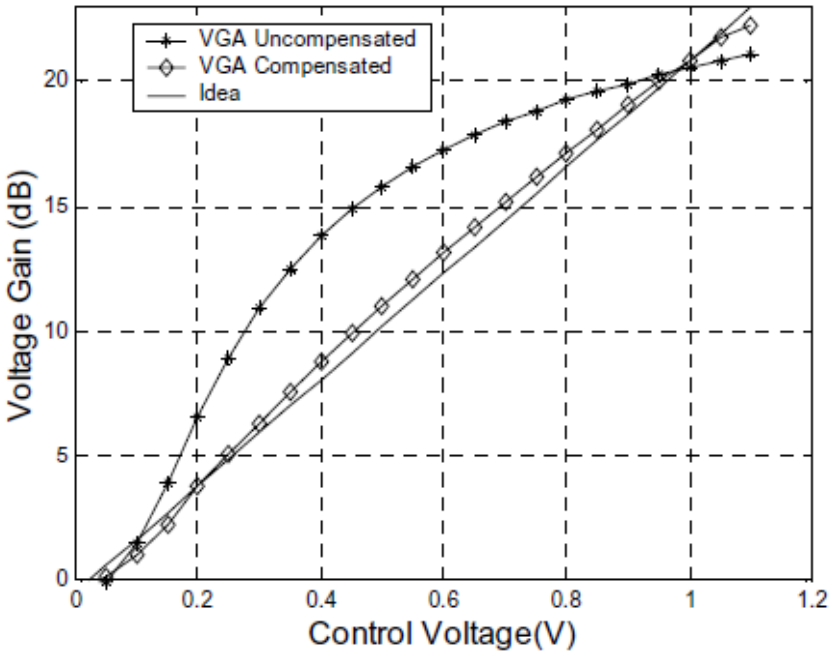


Figure 7.23 Characteristics of the compensated and uncompensated dB-linear VGA

7.3.3.6 Practical Results

In this section we are going to show some practical results for our design. Cadence tool is used in the simulation.

7.3.3.6.1 Design Schematic

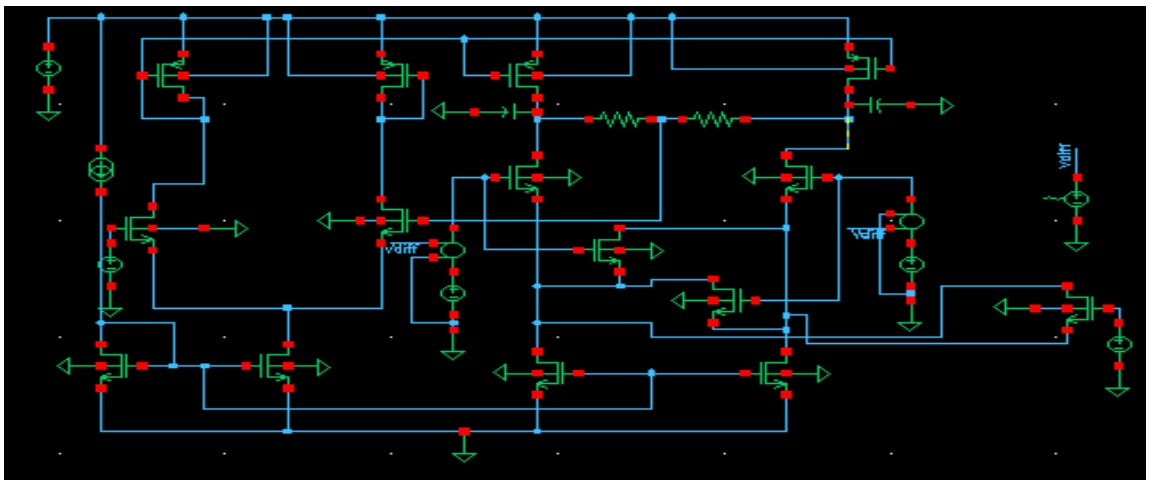


Figure 7.24 Linear VGA designed on Cadence

7.3.3.6.2 Low Gain

It is achieved with: $V_{in}^{+} = V_{in}^{-} = 0.75\ v, V_{CMFB} = 0.68\ v\ , V_{Ctrl} = 0.7\ v\ , C_1 = 5p$

$g_m = 19.94\ mA/V$ and $G_s = 3.527\ \mu\mho$ with $R_1 = R_2 = 550\ \Omega$

$$A_v = -5.18\ dB$$

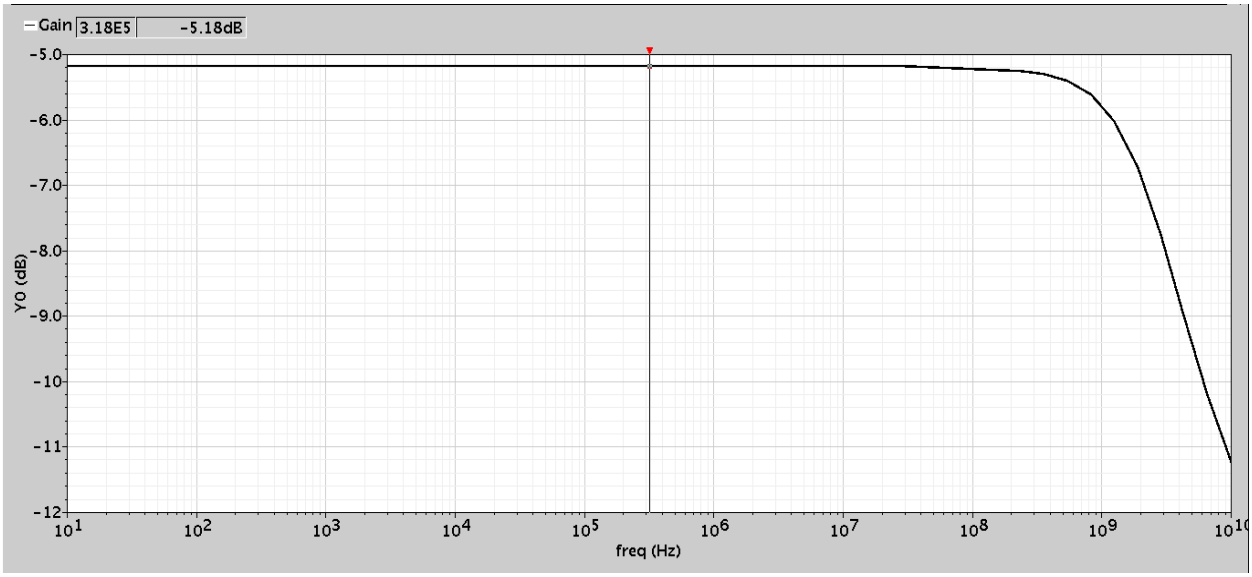


Figure 7.25 Lowest Gain achieved

$$3dB\ BW = 3.31\ GHz$$

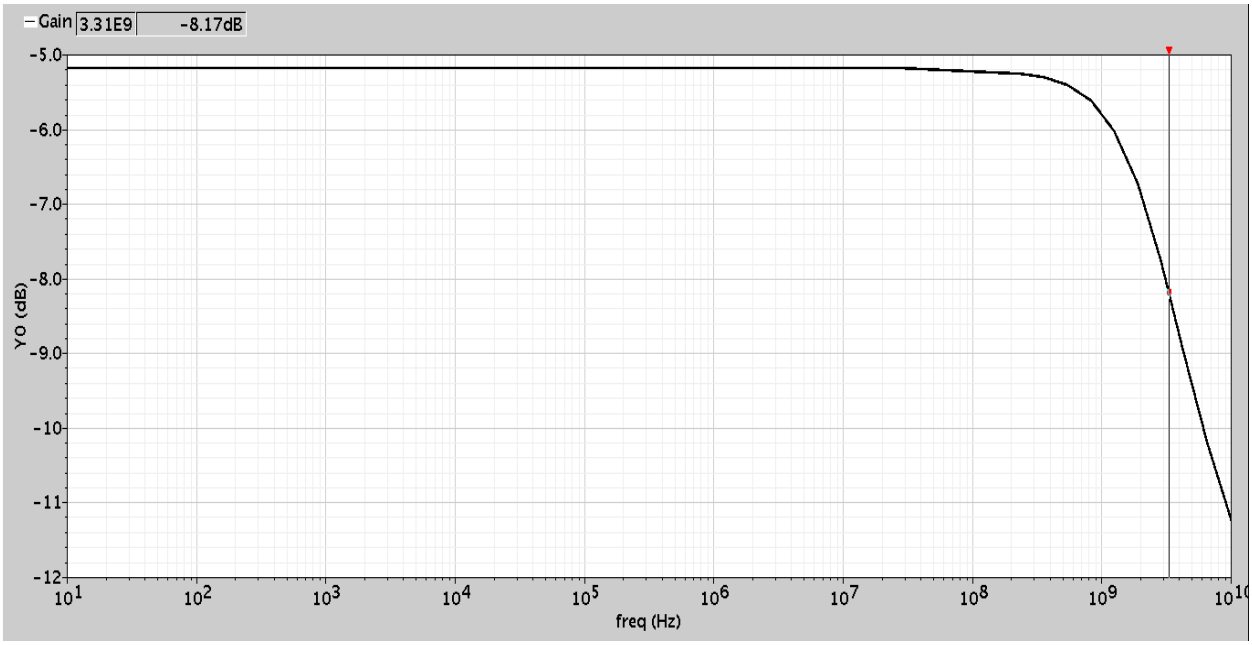


Figure 7.26 3dB Bandwidth for Low Gain

$$IP3 = 1.24551\ dBm$$

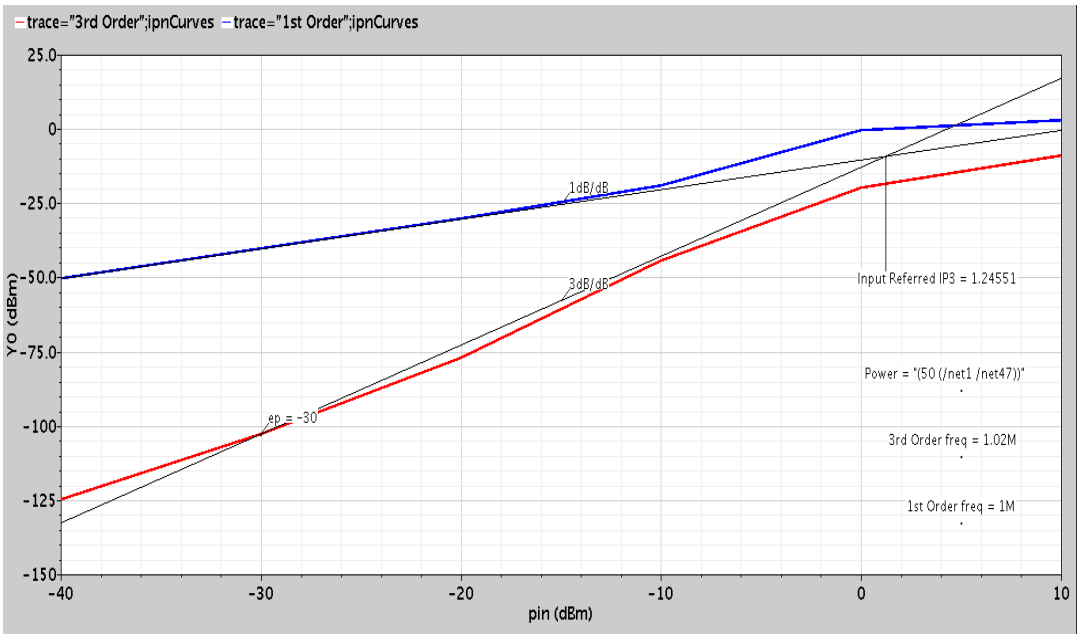


Figure 7.27 Third Order Intercept Point for Low Gain

7.3.3.6.3 High Gain

It is achieved with: $V_{in}^{+} = V_{in}^{-} = 0.75\ v, V_{CMFB} = 0.7\ v\ , V_{Ctrl} = 0.75\ v\ , C_1 = 5p$

$g_m = 16.7\ mA/V$ and $G_s = 10.09\ m\mho$ with $R_1 = R_2 = 600\ \Omega$

$$A_v = 10.43\ dB$$

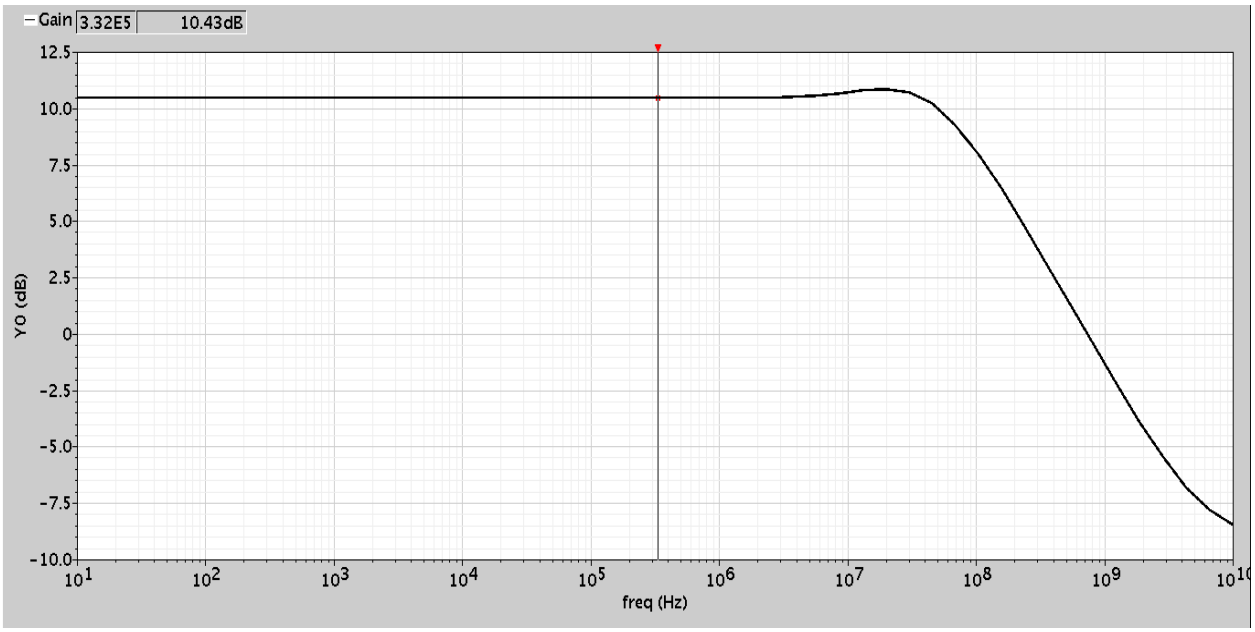


Figure 7.28 Highest Gain achieved

$$3dB\ BW = 0.119\ GHz$$

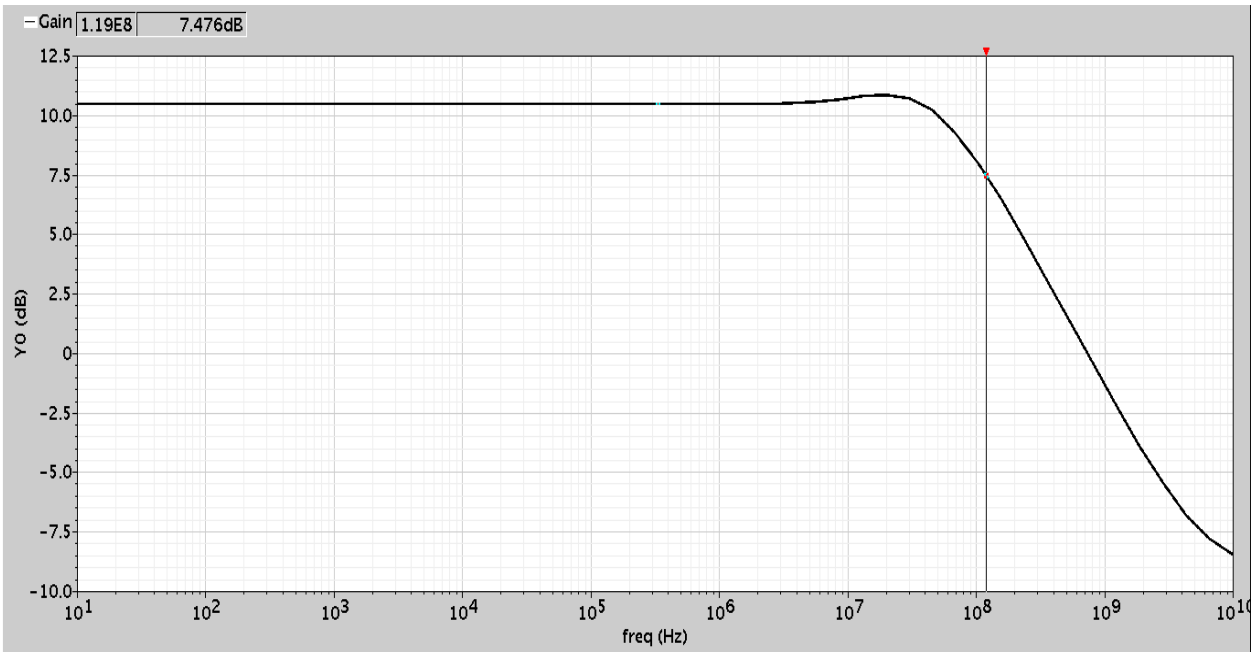


Figure 7.29 3dB Bandwidth for High Gain

$$IP3 = 1.69517\ dBm$$

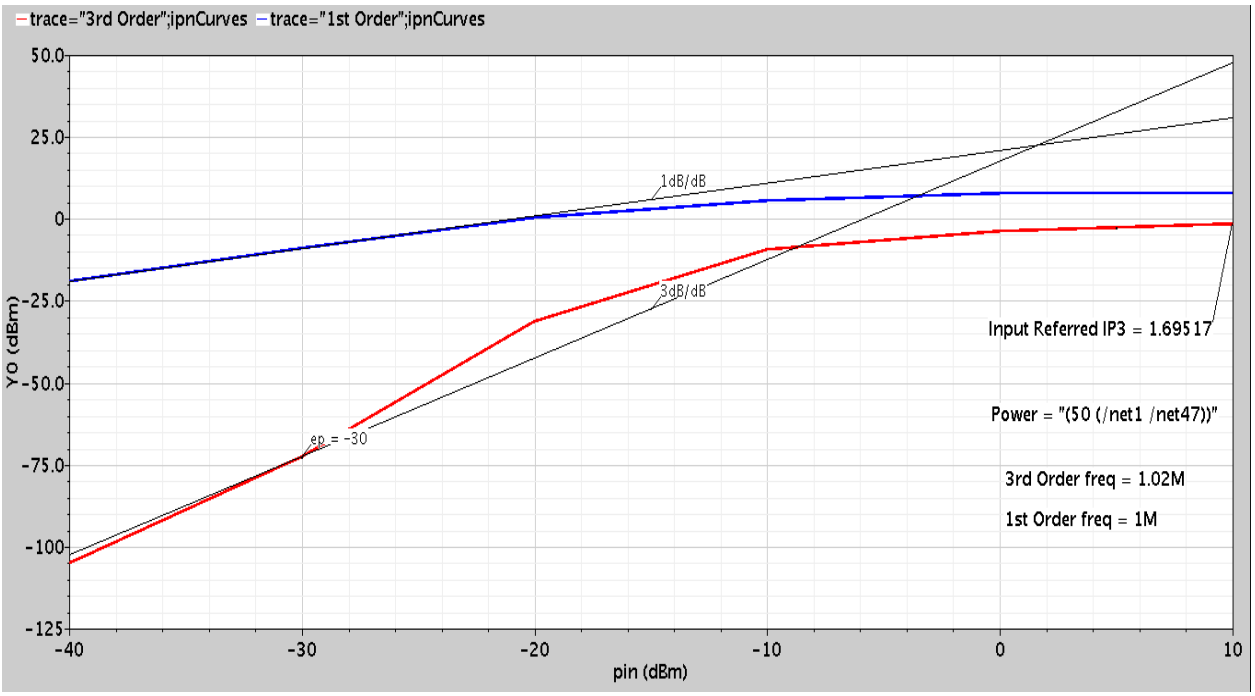


Figure 7.30 Third Order Intercept Point for High Gain

7.3.3.7 Conclusion

A CMOS dB-linear variable gain amplifier is presented. The VGA achieves over $60\ dB$ gain control range with error less than $1\ dB$ using a novel CMOS exponential function generation circuit.

Compensation techniques have been proposed to improve the overall linearity of the VGA caused by nonzero $V_S + V_{th}$ and the dependence of G_S on the input control voltage. The effectiveness of the compensation techniques has been demonstrated in the simulation.

The gain of the VGA is varied by changing the current and hence the transconductance of the input transistors of each of the two variable gain stages, while maintaining the transconductance of the load constant in order to achieve a constant bandwidth throughout the gain range.

Common mode feedback circuit has been included in the design to achieve high CMRR. The use of cascoding to increase the gain range of the VGA as well as its bandwidth has trade-offs with the linearity and output voltage swing.

References:

[1] Banerjee, Dean, "Frequency Synthesis and Planning for *PLL* Architectures", *National Semiconductor,* 2009.

[2] Barale, Francesco, Frequency Dividers Design for Mulit-*GHz PLL* systems, *Georgia Institute of Technology*, 2008.

[3] Deng, Zhiming, Building Blocks for A Wide-Band Phase-Locked Loop, *University of California at Berkeley*.

[4] Duong, Quoc-Hoang, "An all CMOS 84dB-linear low-power Variable Gain Amplifier", *Symposium on VLSI Circuits Digest of Technical Papers*, 2005, pp. 114-7.

[5] Fei, Louis Fan, "Frequency Divider Design strategies", *Broadband Technology*, 2005, pp. 18-24

[6] Hu, W. Chunglen, L. and Wang, X., "Fast Frequency acquisition phase-frequency detector with zero blind zone in *PLL*", *Electronics Letters*, VOL. 43, No. 19, September 2007.

[7] Jalali, Mohammad Sadegh Bakhtiar, Alireza Sharif and Mirabbasi, Shahriar, "A Charge-Pump with a High Output Swing for PLL and CDR Applications", *IEEE Press*, 2010, pp. 297-300.

[8] Krishnanji, Sivasankari, Design of a Variable Gain Amplifier for an Ultra Wideband Receiver, *Texas A&M University*, 2005.

[9] Razavi, Behzad, Design of Analog CMOS Integrated Circuits. *McGraw-Hill*, New York, 2001.

[10] Razavi, Behzad, RF Microelectronics (Second Edition). *Prentice Hall*, New York, 2002.

[11] Yang, Yu-Che and Lu, Shey-Shi, "A Single-*VCO* Fractional-*N* Frequency Synthesizer for Digital *TV* Tuners", *IEEE Transactions on Industrial Electronic*, VOL. 57, No. 9, September 2010, pp. 3207-15.

[12] Zheng, Yuanjin Yan, Jiangnan and Xu, Yong Ping, "A CMOS dB-linear VGA with pre-distortion compensation for wireless communication applications.", *IEEE Press*, 2004, pp. 813-6